

S E C O N D E D I T I O N

Handbook of
**Discrete and
Computational
Geometry**

DISCRETE MATHEMATICS AND ITS APPLICATIONS

Series Editor
Kenneth H. Rosen, Ph.D.

AT&T Laboratories
Middletown, New Jersey

- Miklos Bona*, Combinatorics of Permutations
- Kun-Mao Chao and Bang Ye Wu*, Spanning Trees and Optimization Problems
- Charalambos A. Charalambides*, Enumerative Combinatorics
- Charles J. Colbourn and Jeffrey H. Dinitz*, The CRC Handbook of Combinatorial Designs
- Steven Furino, Ying Miao, and Jianxing Yin*, Frames and Resolvable Designs: Uses, Constructions, and Existence
- Randy Goldberg and Lance Riek*, A Practical Handbook of Speech Coders
- Jacob E. Goodman and Joseph O'Rourke*, Handbook of Discrete and Computational Geometry, Second Edition
- Jonathan Gross and Jay Yellen*, Graph Theory and Its Applications
- Jonathan Gross and Jay Yellen*, Handbook of Graph Theory
- Darrel R. Hankerson, Greg A. Harris, and Peter D. Johnson*, Introduction to Information Theory and Data Compression, Second Edition
- Daryl D. Harms, Miroslav Kraetzl, Charles J. Colbourn, and John S. Devitt*, Network Reliability: Experiments with a Symbolic Algebra Environment
- David M. Jackson and Terry I. Visentin*, An Atlas of Smaller Maps in Orientable and Nonorientable Surfaces
- Richard E. Klima, Ernest Stitzinger, and Neil P. Sigmon*, Abstract Algebra Applications with Maple
- Patrick Knupp and Kambiz Salari*, Verification of Computer Codes in Computational Science and Engineering
- Donald L. Kreher and Douglas R. Stinson*, Combinatorial Algorithms: Generation Enumeration and Search
- Charles C. Lindner and Christopher A. Rodgers*, Design Theory
- Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone*, Handbook of Applied Cryptography
- Richard A. Mollin*, Algebraic Number Theory
- Richard A. Mollin*, Fundamental Number Theory with Applications

Richard A. Mollin, An Introduction to Cryptography

Richard A. Mollin, Quadratics

Richard A. Mollin, RSA and Public-Key Cryptography

Kenneth H. Rosen, Handbook of Discrete and Combinatorial Mathematics

Douglas R. Shier and K.T. Wallenius, Applied Mathematical Modeling: A Multidisciplinary Approach

Douglas R. Stinson, Cryptography: Theory and Practice, Second Edition

Roberto Togneri and Christopher J. deSilva, Fundamentals of Information Theory and Coding Design

Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography

ADVISORY EDITORIAL BOARD

Bernard Chazelle
Princeton University

David P. Dobkin
Princeton University

Herbert Edelsbrunner
Duke University

Ronald L. Graham
University of California, San Diego

Victor Klee
University of Washington

Donald E. Knuth
Stanford University

János Pach
City College, City University of New York

Richard Pollack
Courant Institute, New York University

Günter M. Ziegler
Technische Universität Berlin

SECOND EDITION

Handbook of
Discrete and
Computational
Geometry

edited by
Jacob E. Goodman
Joseph O'Rourke



CHAPMAN & HALL/CRC

A CRC Press Company
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Handbook of discrete and computational geometry / edited by Jacob E. Goodman and Joseph O'Rourke.

p. cm. — (The CRC Press series on discrete mathematics and its applications)

Includes bibliographical references and index.

ISBN 1-58488-301-4 (alk. paper)

1. Combinatorial geometry—Handbooks, manuals, etc. 2. Geometry—Data processing—Handbooks, manuals, etc., I. Goodman, Jacob E. II. O'Rourke, Joseph. III. Title IV. Series.

QA167.H36 2004

516'.13—dc22

2004040662

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

All rights reserved. Authorization to photocopy items for internal or personal use, or the personal or internal use of specific clients, may be granted by CRC Press LLC, provided that \$1.50 per page photocopied is paid directly to Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923 USA. The fee code for users of the Transactional Reporting Service is ISBN 1-58488-301-4/04/\$0.00+\$1.50. The fee is subject to change without notice. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the CRC Press Web site at www.crcpress.com

© 2004 by Chapman & Hall/CRC

No claim to original U.S. Government works

International Standard Book Number 1-58488-301-4

Library of Congress Card Number 2004040662

Printed in the United States of America 1 2 3 4 5 6 7 8 9 0

Printed on acid-free paper

PREFACE

While books and journals of high quality have proliferated in discrete and computational geometry during recent years, there has been to date no single reference work fully accessible to the nonspecialist as well as to the specialist, covering all the major aspects of both fields. The *Handbook of Discrete and Computational Geometry* is intended to do exactly that: to make the most important results and methods in these areas of geometry readily accessible to those who use them in their everyday work, both in the academic world—as researchers in mathematics and computer science—and in the professional world—as practitioners in fields as diverse as operations research, molecular biology, and robotics.

A significant part of the growth that discrete mathematics as a whole has experienced in recent years has consisted of a substantial development in discrete geometry. This has been fueled partly by the advent of powerful computers and by the recent explosion of activity in the relatively young field of computational geometry. This synthesis between discrete and computational geometry, in which the methods and insights of each field have stimulated new understanding of the other, lies at the heart of this Handbook.

The phrase “discrete geometry,” which at one time stood mainly for the areas of packing, covering, and tiling, has gradually grown to include in addition such areas as combinatorial geometry, convex polytopes, and arrangements of points, lines, planes, circles, and other geometric objects in the plane and in higher dimensions. Similarly, “computational geometry,” which referred not long ago to simply the design and analysis of geometric algorithms, has in recent years broadened its scope, and now means the study of geometric problems from a computational point of view, including also computational convexity, computational topology, and questions involving the combinatorial complexity of arrangements and polyhedra. It is clear from this that there is now a significant overlap between these two fields, and in fact this overlap has become one of practice as well, as mathematicians and computer scientists have found themselves working on the same geometric problems and have forged successful collaborations as a result.

At the same time, a growing list of areas in which the results of this work are applicable has been developing. It includes areas as widely divergent as engineering, crystallography, computer-aided design, manufacturing, operations research, geographic information systems, robotics, error-correcting codes, tomography, geometric modeling, computer graphics, combinatorial optimization, computer vision, pattern recognition, and solid modeling.

With this in mind, it has become clear that a handbook encompassing the most important results of discrete and computational geometry would benefit not only the workers in these two fields, or in related areas such as combinatorics, graph theory, geometric probability, and real algebraic geometry, but also the *users* of this body of results, both industrial and academic. This Handbook is designed to fill that role. We believe it will prove an indispensable working tool both for researchers in geometry and geometric computing and for professionals who use geometric tools in their work.

The Handbook covers a broad range of topics in both discrete and computational geometry, as well as in a number of applied areas. These include geometric data structures, polytopes and polyhedra, convex hull and triangulation algorithms, packing and covering, Voronoi diagrams, combinatorial geometric questions, com-

putational convexity, shortest paths and networks, computational real algebraic geometry, geometric arrangements and their complexity, geometric reconstruction problems, randomization and de-randomization techniques, ray shooting, parallel computation in geometry, oriented matroids, computational topology, mathematical programming, motion planning, sphere packing, computer graphics, robotics, crystallography, and many others. A final chapter is devoted to a list of available software. Results are presented in the form of theorems, algorithms, and tables, with every technical term carefully defined in a glossary that precedes the section in which the term is first used. There are numerous examples and figures to illustrate the ideas discussed, as well as a large number of unsolved problems.

The main body of the volume is divided into six parts. The first two, on combinatorial and discrete geometry and on polytopes and polyhedra, deal with fundamental geometric objects such as planar arrangements, lattices, and convex polytopes. The next section, on algorithms and geometric complexity, discusses these basic geometric objects from a computational point of view. The fourth and fifth sections, on data structures and computational techniques, discuss various computational methods that cut across the spectrum of geometric objects, such as randomization and de-randomization, and parallel algorithms in geometry, as well as efficient data structures for searching and for point location. The sixth section, which is the longest in the volume, contains chapters on fourteen applications areas of both discrete and computational geometry, including low-dimensional linear programming, combinatorial optimization, motion planning, robotics, computer graphics, pattern recognition, graph drawing, splines, manufacturing, solid modeling, rigidity of frameworks, scene analysis, error-correcting codes, and crystallography. It concludes with a fifteenth chapter, an up-to-the-minute compilation of available software relating to the various areas covered in the volume. A comprehensive index follows, which includes proper names as well as all of the terms defined in the main body of the Handbook.

A word about references. Because it would have been prohibitive to provide complete references to all of the many thousands of results included in the Handbook, we have to a large extent restricted ourselves to references for either the most important results, or for those too recent to have been included in earlier survey books or articles; for the rest we have provided annotated references to easily accessible surveys of the individual subjects covered in the Handbook, which themselves contain extensive bibliographies. In this way, the reader who wishes to pursue an older result to its source will be able to do so.

On behalf of the sixty-one contributors and ourselves, we would like to express our appreciation to all those whose comments were of great value to the authors of the various chapters: Pankaj K. Agarwal, Noga Alon, Boris Aronov, Saugata Basu, Margaret Bayer, Louis Billera, Martin Blümlinger, Jürgen Bokowski, B.F. Caviness, Bernard Chazelle, Danny Chen, Xiangping Chen, Yi-Jen Chiang, Edmund M. Clarke, Kenneth Clarkson, Robert Connelly, Henry Crapo, Isabel Cruz, Mark de Berg, Jesús De Loera, Giuseppe Di Battista, Michael Drmota, Peter Eades, Jürgen Eckhoff, Noam D. Elkies, Eva Maria Feichtner, Ioannis Fudos, Branko Grünbaum, Dan Halperin, Eszter Hargittai, Ulli Hund, Jürg Hüslér, Peter Johansson, Norman Johnson, Amy Josephczyk, Gil Kalai, Gyula Károlyi, Kevin Klenk, Włodzimierz Kuiperberg, Endre Makai, Jr., Jiří Matoušek, Peter McMullen, Hans Melissen, Bengt Nilsson, Michel Pocchiola, Richard Pollack, Jörg Rambau, Jürgen Richter-Gebert, Allen D. Rogers, Marie-Françoise Roy, Egon Schulte, Dana Scott, Jürgen Sellen, Micha Sharir, Peter Shor, Maxim Michailovich Skriganov, Neil J.A. Sloane, Richard

P. Stanley, Géza Tóth, Ioannis Tollis, Laureen Treacy, Alexander Vardy, Gert Vegter, Pamela Vermeer, Siniša Vrećica, Kevin Weiler, Asia Ivić Weiss, Neil White, Chee-Keng Yap, and Günter M. Ziegler.

In addition, we would like to convey our thanks to the editors of CRC Press for having the vision to commission this Handbook as part of their *Discrete Mathematics and Its Applications* series; to the CRC staff, for their help with the various stages of the project; and in particular to Nora Konopka, with whom we found it a pleasure to work from the inception of the volume.

Finally, we want to express our sincere gratitude to our families: Josy, Rachel, and Naomi Goodman, and Marylynn Salmon and Nell and Russell O'Rourke, for their patience and forbearance while we were in the throes of this project.

Jacob E. Goodman
Joseph O'Rourke

PREFACE TO THE SECOND EDITION

This second edition of the *Handbook of Discrete and Computational Geometry* represents a substantial revision of the first edition, published seven years earlier. The new edition has added over 500 pages, a growth by more than 50%. Each chapter has been thoroughly revised and updated, and we have added thirteen new chapters. The additional room permitted the expansion of the curtailed bibliographies of the first edition, which often required citing other surveys to locate original sources. The new bibliographies make the chapters, insofar as is possible, self-contained. Most chapters have been revised by their original authors, but in a few cases new authors have joined the effort. All together, taking into account the chapters new to this edition, the number of authors has grown from sixty-three to eighty-two.

In the first edition there was one index; now there are two: in addition to the Index of Defined Terms there is also an Index of Cited Authors, which includes everyone referred to by name in either the text or the bibliography of each chapter. The first edition chapter on computational geometry software has been split into two chapters: one on the libraries LEDA and CGAL, the other on additional software. There are five new chapters in the applications section: on algorithms for modeling motion, on surface simplification and 3D-geometry compression, on statistical applications, on Geographic Information Systems and computational cartography, and on biological applications of computational topology. There are new chapters on collision detection and on nearest neighbors in high-dimensional spaces. We have added material on mesh generation, as well as a new chapter on curve and surface reconstruction, and new chapters on embeddings of finite metric spaces, on polygonal linkages, and on geometric graph theory.

All of these new chapters, together with the many new results contained within the Handbook as a whole, attest to the rapid growth in the field since preparation for the first edition began a decade ago. And as before, we have engaged the world's leading experts in each area as our authors.

In addition to the many people who helped with the preparation of the various chapters comprising the first edition, many of whom once again gave invaluable assistance with the present edition, we would also like to thank the following on behalf

of both the authors and ourselves: Nina Amenta, David Avis, Michael Baake, David Bremner, Hervé Brönnimann, Christian Buchta, Sergio Cabello, Yi-Jen Chiang, Mirela Damian, Douglas Dunham, Stefan Felsner, Lukas Finschi, Bernd Gärtner, Ewgenij Gawrilow, Daniel Hug, Ekkehard Köhler, Jeffrey C. Lagarias, Vladimir I. Levenshtein, Casey Mann, Matthias Müller-Hannemann, Rom Pinchasi, Marc E. Pfetsch, Charles Radin, Jorge L. Ramírez Alfonsín, Matthias Reitzner, Thilo Schröder, Jack Snoeyink, Hellmuth Stacheler, Pavel Valtr, and Nikolaus Witte.

We would also like to express our appreciation to Bob Stern, CRC's Executive Editor, who gave us essentially a free hand in choosing how best to fill the additional 500 pages that were allotted to us for this new edition, as well as to Christine Andreasen for her sharp eye and unfailing good humor.

Jacob E. Goodman
Joseph O'Rourke

TABLE OF CONTENTS

Prefaces

Contributors

COMBINATORIAL AND DISCRETE GEOMETRY

- 1 Finite point configurations (*J. Pach*)
- 2 Packing and covering (*G. Fejes Tóth*)
- 3 Tilings (*D. Schattschneider and M. Senechal*)
- 4 Helly-type theorems and geometric transversals (*R. Wenger*)
- 5 Pseudoline arrangements (*J.E. Goodman*)
- 6 Oriented matroids (*J. Richter-Gebert and G.M. Ziegler*)
- 7 Lattice points and lattice polytopes (*A. Barvinok*)
- 8 Low-distortion embeddings of finite metric spaces
(*P. Indyk and J. Matoušek*)
- 9 Geometry and topology of polygonal linkages
(*R. Connelly and E.D. Demaine*)
- 10 Geometric graph theory (*J. Pach*)
- 11 Euclidean Ramsey theory (*R.L. Graham*)
- 12 Discrete aspects of stochastic geometry (*R. Schneider*)
- 13 Geometric discrepancy theory and uniform distribution
(*J.R. Alexander, J. Beck, and W.W.L. Chen*)
- 14 Topological methods (*R.T. Živaljević*)
- 15 Polyominoes (*S.W. Golomb and D.A. Klarner*)

POLYTOPES AND POLYHEDRA

- 16 Basic properties of convex polytopes
(*M. Henk, J. Richter-Gebert, and G.M. Ziegler*)
- 17 Subdivisions and triangulations of polytopes (*C.W. Lee*)
- 18 Face numbers of polytopes and complexes (*L.J. Billera and A. Björner*)
- 19 Symmetry of polytopes and polyhedra (*E. Schulte*)
- 20 Polytope skeletons and paths (*G. Kalai*)
- 21 Polyhedral maps (*U. Brehm and E. Schulte*)

ALGORITHMS AND COMPLEXITY OF FUNDAMENTAL GEOMETRIC OBJECTS

- 22 Convex hull computations (*R. Seidel*)
- 23 Voronoi diagrams and Delaunay triangulations (*S. Fortune*)
- 24 Arrangements (*D. Halperin*)
- 25 Triangulations and mesh generation (*M. Bern*)
- 26 Polygons (*J. O'Rourke and S. Suri*)
- 27 Shortest paths and networks (*J.S.B. Mitchell*)
- 28 Visibility (*J. O'Rourke*)
- 29 Geometric reconstruction problems (*S.S. Skiena*)
- 30 Curve and surface reconstruction (*T.K. Dey*)
- 31 Computational convexity (*P. Gritzmann and V. Klee*)
- 32 Computational topology (*G. Vegter*)
- 33 Computational real algebraic geometry (*B. Mishra*)

GEOMETRIC DATA STRUCTURES AND SEARCHING

- 34 Point location (*J. Snoeyink*)
- 35 Collision and proximity queries (*M.C. Lin and D. Manocha*)
- 36 Range searching (*P.K. Agarwal*)
- 37 Ray shooting and lines in space (*M. Pellegrini*)
- 38 Geometric intersection (*D.M. Mount*)
- 39 Nearest neighbors in high-dimensional spaces (*P. Indyk*)

COMPUTATIONAL TECHNIQUES

- 40 Randomization and derandomization
(*O. Cheong, K. Mulmuley, and E. Ramos*)
- 41 Robust geometric computation (*C.K. Yap*)
- 42 Parallel algorithms in geometry (*M.T. Goodrich*)
- 43 Parametric search (*J.S. Salowe*)
- 44 The discrepancy method in computational geometry (*B. Chazelle*)

APPLICATIONS OF DISCRETE AND COMPUTATIONAL GEOMETRY

- 45 Linear programming (*M. Dyer, N. Megiddo, and E. Welzl*)
- 46 Mathematical programming (*M.J. Todd*)
- 47 Algorithmic motion planning (*M. Sharir*)
- 48 Robotics (*D. Halperin, L.E. Kavraki, and J.-C. Latombe*)
- 49 Computer graphics (*D. Dobkin and S. Teller*)
- 50 Modeling motion (*L.J. Guibas*)
- 51 Pattern recognition (*J. O'Rourke and G.T. Toussaint*)
- 52 Graph drawing (*R. Tamassia and G. Liotta*)
- 53 Splines and geometric modeling (*C.L. Bajaj*)
- 54 Surface simplification and 3D geometry compression (*J. Rossignac*)
- 55 Manufacturing processes (*R. Janardan and T.C. Woo*)
- 56 Solid modeling (*C.M. Hoffmann*)
- 57 Computation of robust statistics: Depth, median, and related measures
(*P.J. Rousseeuw and A. Struyf*)
- 58 Geographic information systems (*M. van Kreveld*)
- 59 Geometric applications of the Grassmann-Cayley algebra (*N.L. White*)
- 60 Rigidity and scene analysis (*W. Whiteley*)
- 61 Sphere packing and coding theory (*G.A. Kabatiansky and J.A. Rush*)
- 62 Crystals and quasicrystals (*M. Senechal*)
- 63 Biological applications of computational topology (*H. Edelsbrunner*)

GEOMETRIC SOFTWARE

- 64 Software (*M. Joswig*)
- 65 Two computational geometry libraries: LEDA and CGAL
(*L. Kettner and S. Näher*)

CONTRIBUTORS

Pankaj K. Agarwal
Department of Computer Science
Duke University
Durham, North Carolina 27708
e-mail: pankaj@cs.duke.edu

John Ralph Alexander, Jr.
Department of Mathematics
University of Illinois
Urbana, Illinois 61801
e-mail: jralex@math.uiuc.edu

Chanderjit L. Bajaj
Center for Computational Visualization
Computer Sciences & Institute of
Computational and Engineering Sciences
University of Texas at Austin
Austin, Texas 78712
e-mail: bajaj@cs.utexas.edu

Alexander I. Barvinok
Department of Mathematics
University of Michigan
Ann Arbor, Michigan 48109
e-mail: barvinok@umich.edu

J  zsef Beck
Department of Mathematics
Rutgers University
New Brunswick, New Jersey 08903
e-mail: jbeck@math.rutgers.edu

Marshall Bern
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, California 94304
e-mail: bern@parc.com

Louis J. Billera
Department of Mathematics
Malott Hall, Cornell University
Ithaca, New York 14853-4201
e-mail: billera@math.cornell.edu

Anders Bj  rner
Department of Mathematics
Royal Institute of Technology
S-100 44 Stockholm, Sweden
e-mail: bjorner@math.kth.se

Ulrich Brehm
Institut f  r Geometrie
Technische Universit  t Dresden
D-01062 Dresden, Germany
e-mail: brehm@math.tu-dresden.de

Bernard Chazelle
Department of Computer Science
Princeton University
Princeton, New Jersey 08544
e-mail: chazelle@cs.princeton.edu

William W.L. Chen
Department of Mathematics
Macquarie University
New South Wales 2109, Australia
e-mail: wchen@ics.mq.edu.au

Otfried Cheong
Department of Computing Sciences
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
e-mail: ocheong@win.tue.nl

Robert Connelly
Department of Mathematics
Cornell University
Ithaca, New York 14853
e-mail: connelly@math.cornell.edu

Erik D. Demaine
MIT Laboratory for Computer Science
200 Technology Square
Cambridge, Massachusetts 02139
e-mail: edemaine@mit.edu

Tamal K. Dey
Dept. of Computer & Information Science
The Ohio State University
Columbus, Ohio 43210
e-mail: tamaldey@cis.ohio-state.edu

David P. Dobkin
Department of Computer Science
Princeton University
Princeton, New Jersey 08544
e-mail: dpd@cs.princeton.edu

Martin Dyer
School of Computer Studies
University of Leeds
Leeds LS2 9JT, United Kingdom
e-mail: dyer@comp.leeds.ac.uk

Herbert Edelsbrunner
Department of Computer Science
Duke University
Durham, North Carolina 27708
e-mail: edels@cs.duke.edu

Gábor Fejes Tóth
Rényi Institute of Mathematics
Hungarian Academy of Sciences
1364 Budapest, Pf. 127, Hungary
e-mail: gfejes@renyi.hu

Steven Fortune
Bell Laboratories
600 Mountain Ave
Murray Hill, New Jersey 07974
e-mail: sjf@bell-labs.com

Solomon Golomb
Dept. of Electrical Engineering-Systems
University of Southern California
Los Angeles, California 90089
e-mail: milly@mizar.usc.edu

Jacob E. Goodman
Department of Mathematics
City College, CUNY
New York, New York 10031
e-mail: jegcc@cunyvm.cuny.edu

Michael T. Goodrich
Department of Computer Science
University of California, Irvine
Irvine, California 92697
e-mail: goodrich@acm.org

Ronald L. Graham
Computer Science and Engineering
University of California, San Diego
La Jolla, California 92093
e-mail: rgraham@cs.ucsd.edu

Peter Gritzmann
Technische Universität München
Zentrum Mathematik
D-85747 Garching, Germany
e-mail: gritzman@ma.tum.de

Leonidas J. Guibas
Department of Computer Science
Stanford University
Stanford, California 94305
e-mail: guibas@cs.stanford.edu

Dan Halperin
School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
e-mail: danha@post.tau.ac.il

Martin Henk
FB Mathematik / IMO
Universität Magdeburg
39106 Magdeburg, Germany
e-mail: henk@mail.math.uni-magdeburg.de

Christoph M. Hoffmann
Computer Science Department
Purdue University
West Lafayette, Indiana 47907
e-mail: hoffmann@cs.purdue.edu

Piotr Indyk
MIT Laboratory for Computer Science
Cambridge, Massachusetts 02139
e-mail: indyk@theory.lcs.mit.edu

Ravi Janardan
Dept. of Computer Science & Engineering
University of Minnesota
Minneapolis, Minnesota 55455
e-mail: janardan@cs.umn.edu

Michael Joswig
Technische Universität Berlin
Fakultät 2, Inst. für Mathematik, MA 6-2
D-10623 Berlin, Germany
e-mail: joswig@math.tu-berlin.de

Grigory Kabatiansky
Inst. of Information Transmission Problems
Russian Academy of Sciences
Bolshoi Karetny, 19
Moscow 101 447, Russia
e-mail: kaba@iitp.ru

Gil Kalai
Institute of Mathematics
Hebrew University
Jerusalem, Israel
e-mail: kalai@math.huji.ac.il

Lydia E. Kavraki
Department of Computer Science
Rice University
Houston, Texas 77005
e-mail: kavraki@cs.rice.edu

Lutz Kettner
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken, Germany
e-mail: kettner@mpi-sb.mpg.de

Victor Klee
Department of Mathematics
University of Washington
Seattle, Washington 98195
e-mail: jmklee@worldnet.att.net

Marc van Kreveld
Department of Computer Science
Utrecht University
P.O. Box 80.089
3508 TB Utrecht, The Netherlands
e-mail: marc@cs.uu.nl

Jean-Claude Latombe
Department of Computer Science
Stanford University
Stanford, California 94305
e-mail: latombe@cs.stanford.edu

Carl Lee
Department of Mathematics
University of Kentucky
Lexington, Kentucky 40506
e-mail: lee@ms.uky.edu

Ming C. Lin
Department of Computer Science
University of North Carolina
Chapel Hill, North Carolina 27599
e-mail: lin@cs.unc.edu

Giuseppe Liotta
Dipartimento di Ingegneria Elettronica
e dell'Informazione
Università di Perugia
Via G. Duranti 93
06125 Perugia, Italy
e-mail: liotta@diei.unipg.it

Dinesh Manocha
Department of Computer Science
University of North Carolina
Chapel Hill, North Carolina 27599
e-mail: dm@cs.unc.edu

Jiří Matoušek
Department of Computer Science
Charles University
Malostranské nám. 25
118 00 Praha 1, The Czech Republic
e-mail: matousek@mff.cuni.cz

Nimrod Megiddo
IBM Almaden Research Center
650 Harry Road
San Jose, California 95120
e-mail: megiddo@theory.stanford.edu

Bhubaneswar Mishra
Courant Institute, NYU
251 Mercer street
New York, New York 10012
e-mail: mishra@cs.nyu.edu

Joseph S. B. Mitchell
Department of Applied Mathematics
and Statistics
Stony Brook University
Stony Brook, New York 11794
e-mail: jsbm@ams.sunysb.edu

David M. Mount
Department of Computer Science
University of Maryland
College Park, Maryland 20742
e-mail: mount@cs.umd.edu

Ketan Mulmuley
Department of Computer Science
The University of Chicago
Ryerson Hall, 1100 E. 58th St.
Chicago, Illinois 60637
e-mail: mulmuley@cs.uchicago.edu

Stefan Näher
Fachbereich IV - Informatik
Universität Trier
D-54286 Trier, Germany
e-mail: naehler@informatik.uni-trier.de

Joseph O'Rourke
Department of Computer Science
Smith College
Northampton, Massachusetts 01063
e-mail: orourke@cs.smith.edu

János Pach
Department of Computer Science
City College, CUNY
New York, New York 10031
e-mail: pach@cims.nyu.edu

Marco Pellegrini
IMC-CNR
Via Santa Maria 46
Pisa 56126, Italy
e-mail: pellegrini@iit.cnr.it

Edgar A. Ramos
Max-Planck-Institut für Informatik
Algorithms and Complexity Group (AG1)
Im Stadtwald
D-66123 Saarbrücken, Germany
e-mail: ramos@mpi-sb.mpg.de

Jürgen Richter-Gebert
Technische Universität München
Zentrum Mathematik
85747 Garching, Germany
e-mail: richter@ma.tum.de

Jarek Rossignac
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332
e-mail: jarek@cc.gatech.edu

Peter J. Rousseeuw
Dept. of Mathematics & Computer Science
University of Antwerp
Middelheimlaan 1
B-2020 Antwerpen, Belgium
e-mail: Peter.Rousseeuw@ua.ac.be

Jason Rush
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052
e-mail: jarush@microsoft.com

Jeffrey Salowe
Cadence Design Systems, Inc.
555 River Oaks Parkway, MS 2B1
San Jose, California 95134
e-mail: jsalowe@cadence.com

Doris Schattschneider
Department of Mathematics
Moravian College
Bethlehem, Pennsylvania 18018
e-mail: schattdo@moravian.edu

Rolf Schneider
Mathematisches Institut
Albert-Ludwigs-Universität
D-79104 Freiburg i. Br., Germany
e-mail: Rolf.Schneider@math.uni-freiburg.de

Egon Schulte
Department of Mathematics
Northeastern University
Boston, Massachusetts 02115
e-mail: schulte@neu.edu

Raimund Seidel
Fachrichtung 6.2-Informatik
Universität des Saarlandes
D-66123 Saarbrücken, Germany
e-mail: rseidel@cs.uni-sb.de

Marjorie Senechal
Department of Mathematics
Smith College
Northampton, Massachusetts 01063
e-mail: senechal@math.smith.edu

Micha Sharir
School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
e-mail: michas@tau.ac.il

Steven S. Skiena
Department of Computer Science
SUNY at Stony Brook
Stony Brook, New York 11794
e-mail: skiena@cs.sunysb.edu

Jack Snoeyink
Department of Computer Science
UNC-Chapel Hill
Chapel Hill, North Carolina 27599
e-mail: snoeyink@cs.unc.edu

Anja Struyf
Dept. of Mathematics & Computing Science
University of Antwerp
Middelheimlaan 1
B-2020 Antwerpen, Belgium
e-mail: Anja.Struyf@ua.ac.be

Subhash Suri
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, California 93106
e-mail: suri@cs.ucsb.edu

Roberto Tamassia
Department of Computer Science
Brown University
115 Waterman Street
Providence, Rhode Island 02912
e-mail: rt@cs.brown.edu

Seth Teller
Computer Science and
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
e-mail: seth@mit.edu

Michael J. Todd
School of Operations Research
and Industrial Engineering
Cornell University
Ithaca, New York 14853
e-mail: miketodd@cs.cornell.edu

Godfried T. Toussaint
School of Computer Science
McGill University
Montréal, Québec H3A 2K6, Canada
e-mail: godfried@opus.cs.mcgill.ca

Gert Vegter
Dept. of Mathematics & Computer Science
University of Groningen
9700 AV Groningen, The Netherlands
e-mail: gert@cs.rug.nl

Emo Welzl
Theoretische Informatik
ETH-Zentrum, IFW
CH-8092 Zürich, Switzerland
e-mail: emo@inf.ethz.ch

Rephael Wenger
Department of Computer Science
Ohio State University
Columbus, Ohio 43210
e-mail: wenger@cis.ohio-state.edu

Neil White
Department of Mathematics
University of Florida
P.O. Box 118105
Gainesville, Florida 32611
e-mail: white@math.ufl.edu

Walter Whiteley
Department of Mathematics
and Statistics
York University
North York, Ontario M3J 1P3, Canada
e-mail: whiteley@mathstat.yorku.ca

Tony C. Woo
Industrial Engineering
University of Washington
Seattle, Washington 98195
e-mail: twooo@u.washington.edu

Chee K. Yap
Courant Institute, NYU
251 Mercer Street
New York, New York 10012
e-mail: yap@cs.nyu.edu

Günter M. Ziegler
Institut für Mathematik, MA 6-2
Technische Universität Berlin
D-10623 Berlin, Germany
e-mail: ziegler@math.tu-berlin.de

Rade Živaljević
Matematički Institut
Knez Mihailova 35/1
11001 Beograd, Yugoslavia
e-mail: rade@turing.mi.sanu.ac.yu

1 FINITE POINT CONFIGURATIONS

János Pach

INTRODUCTION

The study of combinatorial properties of finite point configurations is a vast area of research in geometry, whose origins go back at least to the ancient Greeks. Since it includes virtually all problems starting with “consider a set of n points in space,” space limitations impose the necessity of making choices. As a result, we will restrict our attention to Euclidean spaces and will discuss problems that we find particularly important. The chapter is partitioned into incidence problems (Section 1.1), metric problems (Section 1.2), and coloring problems (Section 1.3).

1.1 INCIDENCE PROBLEMS

In this section we will be concerned mainly with the structure of incidences between a finite point configuration P and a set of finitely many lines (or, more generally, k -dimensional flats, spheres, etc.). Sometimes this set consists of all lines connecting the elements of P . The prototype of such a question was raised by Sylvester [Syl93] more than one hundred years ago: Is it true that for any configuration of finitely many points in the plane, not all on a line, there is a line passing through exactly two points? This question was rediscovered by Erdős [Erd43], and affirmative answers to it were given by Gallai and others [St44]. Generalizations for circles and conic sections in place of lines were established by Motzkin [Mot51] and Wilson-Wiseman [WW88], respectively.

GLOSSARY

Incidence: A point of configuration P lies on an element of a given collection of lines (k -flats, spheres, etc.).

Simple crossing: A point incident with exactly two elements of a given collection of lines or circles.

Ordinary line: A line passing through exactly two elements of a given point configuration.

Ordinary circle: A circle passing through exactly three elements of a given point configuration.

Ordinary hyperplane: A $(d-1)$ -dimensional flat passing through exactly d elements of a point configuration in Euclidean d -space.

Motzkin hyperplane: A hyperplane whose intersection with a given d -dimensional point configuration lies—with the exception of exactly one point—in a $(d-2)$ -dimensional flat.

Family of pseudolines: A family of two-way unbounded Jordan curves, any two of which have exactly one point in common, which is a proper crossing.

Family of pseudocircles: A family of closed Jordan curves, any two of which have at most two points in common, at which the two curves properly cross each other.

Regular family of curves: A family Γ of curves in the xy -plane defined in terms of D real parameters satisfying the following properties. There is an integer s such that (a) the dependence of the curves on x, y , and the parameters is algebraic of degree at most s ; (b) no two distinct curves of Γ intersect in more than s points; (c) for any D points of the plane, there are at most s curves in Γ passing through all of them.

Degrees of freedom: The smallest number D of real parameters defining a regular family of curves.

Spanning tree: A tree whose vertex set is a given set of points and whose edges are line segments.

Spanning path: A spanning tree that is a polygonal path.

Convex position: P forms the vertex set of a convex polygon or polytope.

k -set: A k -element subset of P that can be obtained by intersecting P with an open halfspace.

Halving plane: A hyperplane with $\lfloor |P|/2 \rfloor$ points of P on each side.

SYLVESTER-TYPE RESULTS

1. Gallai theorem (dual version): Any set of lines in the plane, not all of which pass through the same point, determines a simple crossing. This holds even for families of pseudolines [KR72].
2. Pinchasi theorem: Any set of at least five pairwise crossing unit circles in the plane determines a simple crossing.
Any sufficiently large set of pairwise crossing pseudocircles in the plane, not all of which pass through the same pair of points, determines an intersection point incident to at most three pseudocircles [NPP⁺02]
3. Pach-Pinchasi theorem: Given n red and n blue points in the plane, not all on a line, there always exists a bichromatic line containing at most two points of each color [PP00].
Any finite set of red and blue points contains a monochromatic spanned line, but not always a monochromatic ordinary line [Cha70].
4. Motzkin-Hansen theorem: For any finite set of points in Euclidean d -space, not all of which lie on a hyperplane, there exists a Motzkin hyperplane [Mot51, Han65]. We obtain as a corollary that n points in d -space, not all of which lie on a hyperplane, determine at least n distinct hyperplanes. (A hyperplane is *determined* by a point set P if its intersection with P is not contained in a $(d-2)$ -flat.) Putting the points on two skew lines in 3-space shows that the existence of an ordinary hyperplane cannot be guaranteed for $d > 2$.

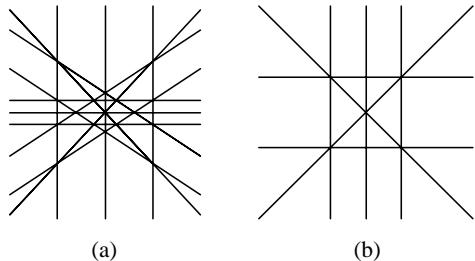
If $n > 8$ is sufficiently large, then any set of n noncocircular points in the plane determines at least $\binom{n-1}{2}$ distinct circles, and this bound is best possible [Ell67]. The number of ordinary circles determined by n noncocircular points is known to be at least $11n(n-1)/247$ [BB94].

5. Csima-Sawyer theorem: Any set of n noncollinear points in the plane determines at least $6n/13$ ordinary lines ($n > 7$). This bound is sharp for $n = 13$ and false for $n = 7$ (see Figure 1.1.1). [KM58, CS93]). In 3-space, any set of n noncoplanar points determines at least $2n/5$ Motzkin hyperplanes [Han80, GS84].

FIGURE 1.1.1

Extremal examples for the (dual) Csima-Sawyer theorem:

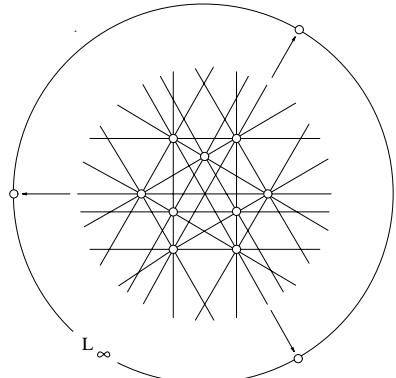
- (a) 13 lines (including the line at infinity) determining only 6 simple points;
- (b) 7 lines determining only 3 simple points.



6. Orchard problem [Syl67]: What is the maximum number of collinear triples determined by n points in the plane, no four on a line? There are several constructions showing that this number is at least $n^2/6 - O(n)$, which is asymptotically best possible, cf. [BGS74, FP84]. (See Figure 1.1.2.)

FIGURE 1.1.2

12 points and 19 lines, each passing through exactly 3 points.



7. Dirac's problem [Dir51]: Does there exist a constant c such that any set of n points in the plane, not all on a line, has an element incident to at least $n/2 - c$ connecting lines? If true, this result is best possible, as is shown by the example of n points distributed as evenly as possible on two intersecting lines. (It was believed that, apart from some small examples listed in [Grü72], this statement is true with $c = 0$, until Felsner exhibited an infinite series of configurations, showing that $c \geq 3/2$.) It is known that

there is a positive constant c such that one can find a point incident to at least cn connecting lines. A useful equivalent formulation of this assertion is that any set of n points in the plane, no more than $n - k$ of which are on the same line, determines at least $c'kn$ distinct connecting lines, for a suitable constant $c' > 0$. Note that according to the $d = 2$ special case of the Motzkin-Hansen theorem, due to Erdős (see No. 4 above), for $k = 1$ the number of distinct connecting lines is at least n . For $k = 2$, the corresponding bound is $2n - 4$, ($n \geq 10$).

8. Ungar's theorem [Ung82]: n noncollinear points in the plane always determine at least $2\lfloor n/2 \rfloor$ lines of different slopes (see Figure 1.1.3); this proves Scott's conjecture. Furthermore, any set of n points in the plane, not all on a line, permits a spanning tree, all of whose $n - 1$ edges have different slopes [Jam87]. Pach, Pinchasi, and Sharir showed that n noncoplanar points in 3-space determine at least $2n - 3$ different directions if n is even and at least $2n - 2$ if n is odd, provided that no 3 points are on a line. Even without this latter assumption, the number of different directions is at least $2n - O(1)$.

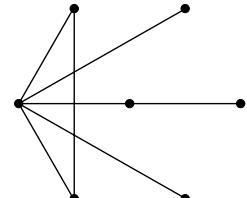


FIGURE 1.1.3
7 points determining 6 distinct slopes.

UPPER BOUNDS ON THE NUMBER OF INCIDENCES

Given a set P of n points and a family Γ of m curves or surfaces, the number of incidences between them can be obtained by summing over all $p \in P$ the number of elements of Γ passing through p . If the elements of Γ are taken from a regular family of curves with D degrees of freedom [PS90], the maximum number of incidences between P and Γ is $O(n^{D/(2D-1)}m^{(2D-2)/(2D-1)} + n + m)$. In the most important applications, Γ is a family of straight lines or unit circles in the plane ($D = 2$), or it consists of circles of arbitrary radii ($D = 3$). The best upper bounds known for the number of incidences are summarized in Table 1.1.1. It follows from the first line of the table that for any set P of n points in the plane, the number of distinct straight lines containing at least k elements of P is $O(n^2/k^3 + n/k)$, and this bound cannot be improved (Szemerédi-Trotter). In the second half of the table, $\kappa(n, m)$ and $\beta(n, m)$ denote extremely slowly growing functions, which are certainly $o(n^\epsilon m^\epsilon)$ for every $\epsilon > 0$. A family of pseudocircles is *special* if its curves admit a 3-parameter algebraic representation. A collection of spheres in 3-space is said to be in *general position* here if no three of them pass through the same circle [CEG⁺90, NPP⁺02].

MIXED PROBLEMS

Many problems about finite point configurations involve some notions that cannot be defined in terms of incidences: convex position, midpoint of a segment, etc.

TABLE 1.1.1 Maximum number of incidences between n points of P and m elements of Γ [SzT83, CEG⁺90, NPP⁺02].

PT. SET P	FAMILY Γ	BOUND	TIGHT
Planar	lines	$O(n^{2/3}m^{2/3} + n + m)$	yes
Planar	pseudolines	$O(n^{2/3}m^{2/3} + n + m)$	yes
Planar	unit circles	$O(n^{2/3}m^{2/3} + n + m)$?
Planar	pairwise crossing circles	$O(n^{1/2}m^{5/6} + n^{2/3}m^{2/3} + n + m)$?
Planar	special pseudocircles	$O(n^{6/11}m^{9/11}\kappa(n, m) + n^{2/3}m^{2/3} + n + m)$?
Planar	pairwise crossing pseudocircles	$O(n^{2/3}m^{2/3} + n + m^{4/3})$?
3-dim'l	spheres	$O(n^{4/7}m^{9/7}\beta(n, m) + n^2)$?
3-dim'l	spheres in gen. position	$O(n^{3/4}m^{3/4}\beta(n, m) + n + m)$?
d-dim'l	circles	$O(n^{6/11}m^{9/11}\kappa(n, m) + n^{2/3}m^{2/3} + n + m)$?

Below we list a few questions of this type. They are discussed in this part of the chapter, and not in Section 1.2 which deals with metric questions, because we can disregard most aspects of the Euclidean metrics in their formulation. For example, convex position can be defined by requiring that some sets should lie on one side of certain hyperplanes. This is essentially equivalent to introducing an order along each straight line.

1. Erdős-Klein-Szekeres problem: What is the maximum number of points that can be chosen in the plane so that no three are on a line and no k are in convex position ($k > 3$)? If this number is denoted by $c(k)$, it is known [TV98, ES35, ES61] that

$$2^{k-2} \leq c(k) \leq \binom{2n-5}{n-2}.$$

Let $e(k)$ denote the maximum size of a planar point set P that has no three elements on a line and no k elements that form the vertex set of an “empty” convex polygon, i.e., a convex k -gon whose interior is disjoint from P . We have $e(3) = 2$, $e(4) = 4$, $e(5) = 9$, and Horton showed that $e(k)$ is infinite for all $k \geq 7$ [Har78, Hor83]. It is an outstanding open problem to decide whether $e(6)$ is finite.

2. The number of empty k -gons: Let $H_k^d(n)$ ($n \geq k \geq d+1$) denote the minimum number of k -tuples that induce an empty convex polytope of k vertices in a set of n points in d -space, no $d+1$ of which lie on a hyperplane. Clearly, $H_2^1(n) = n-1$ and $H_k^1(n) = 0$ for $k > 2$. For $k = d+1$, we have

$$\frac{1}{d!} \leq \lim_{n \rightarrow \infty} H_k^d(n)/n^d \leq \frac{2}{(d-1)!},$$

[Val95]. For $d = 2$, the best estimates known for $H_k^2 = \lim_{n \rightarrow \infty} H_k^2(n)/n^2$ are given in [Dum00] and [BV03]:

$$1 \leq H_3^2 \leq 1.62, \quad 1/2 \leq H_4^2 \leq 1.94, \quad 0 \leq H_5^2 \leq 1.021,$$

$$0 \leq H_6^2 \leq 0.201, \quad H_7^2 = H_8^2 = \dots = 0.$$

3. The number of k -sets [ELSS73]: Let $N_k^d(n)$ denote the maximum number of k -sets in a set of n points in d -space, no $d+1$ of which lie on the same hyperplane. In other words, $N_k^d(n)$ is the maximum number of different ways in which k points of an n -element set can be separated from the others by a hyperplane. It is known that

$$ne^{\Omega(\sqrt{\log k})} \leq N_k^d(n) \leq O\left(n(k+1)^{1/3}\right)$$

[Tót01, Dey98]. The most interesting case is $k = \frac{n}{2}$ in the plane, which is the maximum number of distinct ways to cut a set of n points in the plane in half (number of halving lines). For the number of halving planes [SST01], $N_{\lfloor n/2 \rfloor}^3(n) = O(n^{5/2})$, and

$$n^{d-1}e^{\Omega(\sqrt{\log n})} \leq N_{\lfloor n/2 \rfloor}^d(n) = o(n^d)$$

[Tót01, ŽV92].

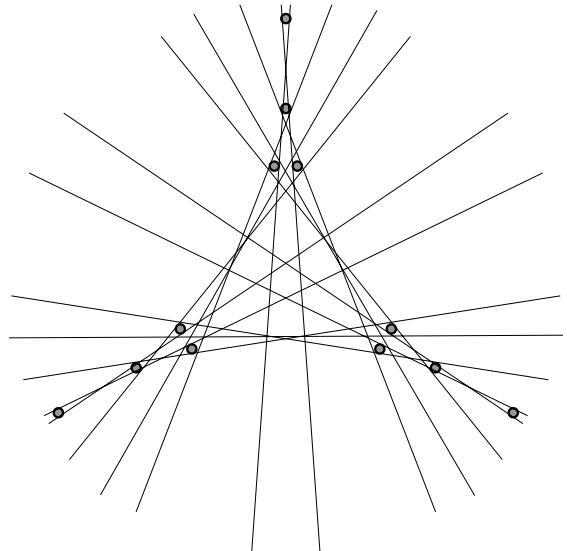


FIGURE 1.1.4

12 points determining 15 combinatorially distinct halving lines.

The maximum number of *at-most- k* -element subsets of a set of n points in d -space, no $d+1$ of which lie on a hyperplane, is $O(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil})$, and this bound is asymptotically tight [CS89]. In the plane the maximum number of *at-most- k* -element subsets of a set of n points is kn for $k < \frac{n}{2}$, which is reached for convex n -gons [AG86, Pe85].

4. The number of midpoints: Let $M(n)$ denote the minimum number of different midpoints of the $\binom{n}{2}$ line segments determined by n points in convex position in the plane. One might guess that $M(n) \geq (1 - o(1))\binom{n}{2}$, but it was shown in [EFF91] that

$$\binom{n}{2} - \left\lfloor \frac{n(n+1)(1-e^{-1/2})}{4} \right\rfloor \leq M(n) \leq \binom{n}{2} - \left\lfloor \frac{n^2 - 2n + 12}{20} \right\rfloor.$$

5. Midpoint-free subsets: As a partial answer to a question proposed in [BMP04], it was proved by V. Bálint et al. that if $m(n)$ denotes the largest number m such that every set of n points in the plane has a midpoint-free subset of size m , then

$$\left\lceil \frac{-1 + \sqrt{8n + 1}}{2} \right\rceil \leq m(n).$$

However, asymptotically, $n^{1-c}/\sqrt{\log n} \leq m(n) \leq n/\log^{c'} n$, for suitable constants $c, c' > 0$ [Pac03].

OPEN PROBLEMS

Here we give six problems from the multitude of interesting questions that remain open.

1. Motzkin-Dirac conjecture: Any set of n noncollinear points in the plane determines at least $n/2$ ordinary lines ($n > 13$).
2. Generalized orchard problem (Grünbaum): What is the maximum number $c_k(n)$ of collinear k -tuples determined by n points in the plane, no $k+1$ of which are on a line ($k \geq 3$)? In particular, show that $c_4(n) = o(n^2)$. Grünbaum [Grü76] established the lower bound $c_k(n) = \Omega(n^{1+1/(k-2)})$, which was improved by Ismailescu [Ism02] to $c_k(n) = \Omega(n^{\frac{\log k+4}{\log k}})$ for $5 \leq k \leq 18$, $c_k(n) = \Omega(n^{\frac{1}{k-3.59}})$ for $k \geq 18$. For $k = 3$, we have $c_3(n) = n^2/6 - \Theta(n)$ [BGS74, FP84].
3. Maximum independent subset problem (Erdős): Determine the largest number $\alpha(n)$ such that any set of n points in the plane, no four on a line, has an $\alpha(n)$ -element subset with no collinear triples. Füredi [Für91] has shown that $\Omega(\sqrt{n \log n}) \leq \alpha(n) \leq o(n)$.
4. Slope problem (Jamison): Does every set of n points in the plane, not all on a line, permit a spanning path, all of whose $n-1$ edges have different slopes?
5. Empty triangle problem (Bárány): Does every set of n points in the plane, no three on a line, determine at least $t(n)$ empty triangles that share a side, where $t(n)$ is a suitable function tending to infinity?
6. Balanced partition problem (Kupitz): Does there exist an integer k with the property that for every planar point set P , there is a connecting line such that the difference between the number of elements of P on its left side and right side does not exceed k ? Some examples due to Alon show that this assertion is not true with $k = 1$. Pinchasi proved that there is a connecting line, for which this difference is $O(\log \log n)$.

1.2 METRIC PROBLEMS

The systematic study of the distribution of the $\binom{n}{2}$ distances determined by n points was initiated by Erdős in 1946 [Erd46]. Given a point configuration $P =$

$\{p_1, p_2, \dots, p_n\}$, let $g(P)$ denote the number of distinct distances determined by P , and let $f(P)$ denote the number of times that the unit distance occurs between two elements of P . That is, $f(P)$ is the number of pairs p_i, p_j ($i < j$) such that $|p_i - p_j| = 1$. What is the minimum of $g(P)$ and what is the maximum of $f(P)$ over all n -element subsets of Euclidean d -space? These questions have raised deep number-theoretic and combinatorial problems, and have contributed richly to many recent developments in these fields.

GLOSSARY

Unit distance graph: A graph whose vertex set is a given point configuration P , in which two points are connected by an edge if and only if their distance is one.

Diameter: The maximum distance between two points of P .

General position in the plane: No three points of P are on a line, and no four on a circle.

Separated set: The distance between any two elements is at least one.

Nearest neighbor of $p \in P$: A point $q \in P$, whose distance from p is minimum.

Farthest neighbor of $p \in P$: A point $q \in P$, whose distance from p is maximum.

Homothetic sets: Similar sets in parallel position.

REPEATED DISTANCES

Extremal graph theory has played an important role in this area. For example, it is easy to see that the unit distance graph assigned to an n -element planar point set P cannot contain $K_{2,3}$, a complete bipartite graph with 2 and 3 vertices in its classes. Thus, by a well-known graph-theoretic result, $f(P)$, the number of edges in this graph, is at most $O(n^{3/2})$. This bound can be improved to $O(n^{4/3})$ by using more sophisticated combinatorial techniques (apply line 3 of Table 1.1.1 with $m = n$); but we are still far from knowing what the best upper bound is.

TABLE 1.2.1 Estimates for the maximum number of unit distances determined by an n -element planar point set P .

POINT SET P	LOWER BOUND	UPPER BOUND	SOURCE
Arbitrary	$n^{1+c/\log\log n}$	$O(n^{4/3})$	[Erd46, SST84]
Separated	$\lfloor 3n - \sqrt{12n - 3} \rfloor$	$\lfloor 3n - \sqrt{12n - 3} \rfloor$	[Reu72, Har74]
Of diameter 1	n	n	[HP34]
In convex position	$2n - 7$	$O(n \log n)$	[EH90, Für90]
No 3 collinear	$\Omega(n \log n)$	$O(n^{4/3})$	Kártész
Separated, no 3 coll.	$(2 + 5/16 - o(1))n$	$(2 + 3/7)n$	[Tót97]

In Table 1.2.1, we summarize the best currently known estimates on the maximum number of times the unit distance can occur among n points in the plane, under various restrictions on their position. In the first line of the table—and throughout this chapter— c denotes (unrelated) positive constants. The second and third lines show how many times the minimum distance and the maximum distance, resp., can occur among n arbitrary points in the plane. Table 1.2.2 contains some analogous results in higher dimensions. In the first line, $\beta(n)$ is an extremely slowly growing function, closely related to the functional inverse of the Ackermann function.

FIGURE 1.2.1

A separated point set with $\lfloor 3n - (12n - 3)^{1/2} \rfloor$ unit distances ($n = 69$). All such sets have been characterized by Kupitz [Kup94].

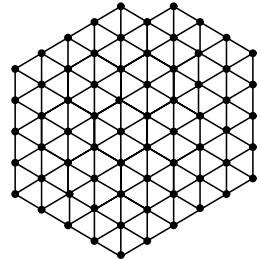


TABLE 1.2.2 Estimates for the maximum number of unit distances determined by an n -element point set P in d -space.

POINT SET P	LOWER BOUND	UPPER BOUND	SOURCE
$d = 3$, arbitrary	$\Omega(n^{4/3} \log \log n)$	$O(n^{3/2} \beta(n))$	[Erd60, CEG ⁺ 90]
$d = 3$, separated	$6n - O(n^{2/3})$	$6n - \Omega(n^{2/3})$	Newton
$d = 3$, diameter 1	$2n - 2$	$2n - 2$	[Grü56, Hep56]
$d = 3$, on sphere (rad. $1/\sqrt{2}$)	$\Omega(n^{4/3})$	$O(n^{4/3})$	[EHP89]
$d = 3$, on sphere (rad. $r \neq 1/\sqrt{2}$)	$\Omega(n \sqrt{\log n})$	$O(n^{4/3})$	[SV04b]
$d = 4$	$\lfloor \frac{n^2}{4} \rfloor + n - 1$	$\lfloor \frac{n^2}{4} \rfloor + n$	[Bra97, vW99]
$d \geq 4$ even, arb.	$\frac{n^2}{2} \left(1 - \frac{1}{\lfloor d/2 \rfloor}\right) + n - O(d)$	$\frac{n^2}{2} \left(1 - \frac{1}{\lfloor d/2 \rfloor}\right) + n - \Omega(d)$	[Erd67]
$d > 4$ odd, arb.	$\frac{n^2}{2} \left(1 - \frac{1}{\lfloor d/2 \rfloor}\right) + \Omega(n^{4/3})$	$\frac{n^2}{2} \left(1 - \frac{1}{\lfloor d/2 \rfloor}\right) + O(n^{4/3})$	[EP90]

The second line of Table 1.2.1 can be extended by showing that the smallest distance cannot occur more than $3n - 2k + 4$ times between points of an n -element set in the plane whose convex hull has k vertices [Bra92a]. The maximum number of occurrences of the second-smallest and second-largest distance is $(24/7 + o(1))n$ and $3n/2$ (if n is even), respectively [Bra92b, Ves78].

Given any point configuration P , let $\Phi(P)$ denote the sum of the numbers of farthest neighbors for every element $p \in P$. Table 1.2.3 contains tight upper bounds on $\Phi(P)$ in the plane and in 3-space, and asymptotically tight ones for higher dimensions [ES89, Csi96, EP90]. Dumitrescu and Guha raised the following related question: given a colored point set in the plane, its *heterocolored diameter* is the largest distance between two elements of different colors. Let $\phi_k(n)$ denote

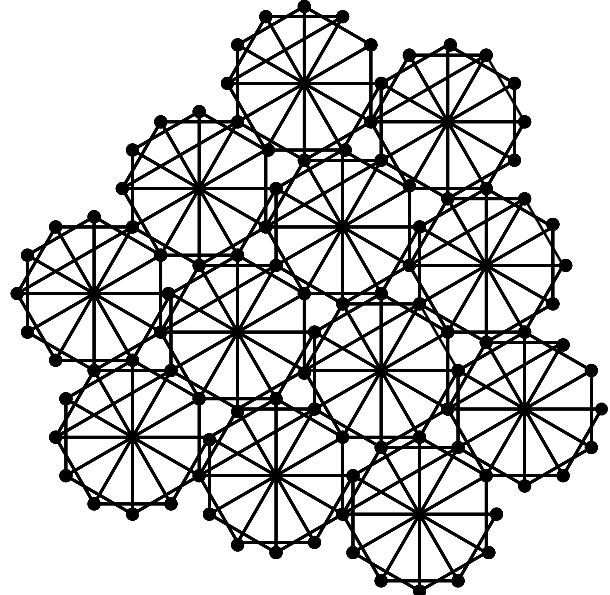


FIGURE 1.2.2

n points, among which the second-smallest distance occurs $(\frac{24}{7} + o(1))n$ times.

the maximum number of times that the heterocolored diameter can occur in a k -colored n -element point set between two points of different colors. It is known that $\phi_2(n) = n$, $\phi_3(n)$ and $\phi_4(n) = 3n/2 + O(1)$ and $\phi_k(n) \leq (2 - \frac{1}{\lceil k/2 \rceil})n$ for every k .

TABLE 1.2.3 Upper bounds on $\Phi(P)$, the total number of farthest neighbors of all points of an n -element set P .

POINT SET P	UPPER BOUND	SOURCE
Planar, n is even	$3n - 3$	[ES89, Avi84]
Planar, n is odd	$3n - 4$	[ES89, Avi84]
Planar, in convex position	$2n$	[ES89]
3-dimensional, $n \equiv 0 \pmod{2}$	$n^2/4 + 3n/2 + 3$	[Csi96, AEP88]
3-dimensional, $n \equiv 1 \pmod{4}$	$n^2/4 + 3n/2 + 9/4$	[Csi96, AEP88]
3-dimensional, $n \equiv 3 \pmod{4}$	$n^2/4 + 3n/2 + 13/4$	[Csi96, AEP88]
d -dimensional ($d > 3$)	$n^2(1 - 1/\lfloor d/2 \rfloor + o(1))$	[EP90]

DISTINCT DISTANCES

It is obvious that if all distances between pairs of points of a d -dimensional set P are the same, then $|P| \leq d + 1$. If P determines at most g distinct distances, we have that $|P| \leq \binom{d+g}{d}$; see [BBS83]. This implies that if d is fixed and n tends to infinity, then the minimum number of distinct distances determined by n points in d -space is at least $\Omega(n^{1/d})$. Denoting this minimum by $g_d(n)$, for $d \geq 3$ we have the following results [SV04a]:

$$\Omega(n^{\frac{2}{d} - \frac{2}{d(d+2)}}) \leq g_d(n) \leq O(n^{2/d}).$$

For $d = 3$, Solymosi and Vu established a better bound, $g_3(n) = \Omega(n^{0.5643})$. In Table

1.2.4, we list some lower and upper bounds on the minimum number of distinct distances determined by an n -element point set P , under various assumptions on its structure.

TABLE 1.2.4 Estimates for the minimum number of distinct distances determined by an n -element point set P in the plane.

POINT SET P	LOWER BOUND	UPPER BOUND	SOURCE
Arbitrary	$\Omega(n^{0.8641})$	$O(n/\sqrt{\log n})$	[ST01, KT04]
In convex position	$\lfloor n/2 \rfloor$	$\lfloor n/2 \rfloor$	[Alt63]
No 3 collinear	$\lceil (n-1)/3 \rceil$	$\lfloor n/2 \rfloor$	Szemerédi [Erd75]
In general position	$\Omega(n)$	$O(n^{1+c}/\sqrt{\log n})$	[EFPR93]

RELATED RESULTS

1. Integer distances: There are arbitrarily large, noncollinear finite point sets in the plane such that all distances determined by them are integers, but there exists no infinite set with this property [AE45].
2. Generic subsets: Any set of n points in the plane contains $\Omega(n^{0.287})$ points such that all distances between them are distinct [LT95]. This bound could perhaps be improved to about $n^{1/3}$.
3. Borsuk's problem: It was conjectured that every (finite) d -dimensional point set P can be partitioned into $d+1$ parts of smaller diameter. It follows from the results quoted in the third lines of Tables 1.2.1 and 1.2.2 that this is true for $d=2$ and 3. Surprisingly, Kahn and Kalai [KK93] proved that there exist sets P that cannot be partitioned into fewer than $(1.2)^{\sqrt{d}}$ parts of smaller diameter. In particular, the conjecture is false for $d=321$ (see, e.g., O. Pikhurko). On the other hand, it is known that for large d , every d -dimensional set can be partitioned into $(\sqrt{3/2} + o(1))^d$ parts of smaller diameter [Sch88].
4. Nearly equal distances: Two numbers are said to be nearly equal if their difference is at most one. If n is sufficiently large, then the maximum number of times that nearly the same distance occurs among n separated points in the plane is $\lfloor n^2/4 \rfloor$. The maximum number of pairs in a separated set of n points in the plane, whose distance is nearly equal to any one of k arbitrarily chosen numbers, is $\frac{n^2}{2}(1 - \frac{1}{k+1} + o(1))$, as n tends to infinity [EMP93].
5. Repeated angles: In an n -element planar point set, the maximum number of noncollinear triples that determine the same angle is $O(n^2 \log n)$, and this bound is asymptotically tight for a dense set of angles (Pach-Sharir). The corresponding maximum in 3-space is at most $O(n^{8/3})$ [CCEG79]. In 4-space the angle $\pi/2$ can occur $\Omega(n^3)$ times, and all other angles can occur at most $O(n^{74/25})$ times [Pu88]. For dimension $d \geq 5$ all angles can occur $\Omega(n^3)$ times.

6. Repeated areas: Let $t_d(n)$ denote the maximum number of triples in an n -element point set in d -space that induce a unit area triangle. It is known that $\Omega(n^2 \log \log n) \leq t_2(n) \leq O(n^{7/3})$, $t_3(n) = O(n^{8/3})$, $t_4(n), t_5(n) = o(n^3)$, and $t_6(n) = \Theta(n^3)$ ([EP71, PS90]). Maximum- and minimum-area triangles occur among n points in the plane at most n and at most $\Theta(n^2)$ times [BRS01].
7. Congruent triangles: Let $T_d(n)$ denote the maximum number of triples in an n -element point set in d -space that induce a triangle congruent to a given triangle T . It is known [AS01, ÁF02] that
$$\begin{aligned} \Omega(n^{1+c/\log \log n}) &\leq T_2(n) \leq O(n^{4/3}), \\ \Omega(n^{4/3}) &\leq T_3(n) \leq O(n^{5/3+\epsilon}), \\ \Omega(n^2) &\leq T_4(n) \leq O(n^{2+\epsilon}), \\ T_5(n) &= \Theta(n^{7/3}), \text{ and} \\ T_d(n) &= \Theta(n^3) \text{ for } d \geq 6. \end{aligned}$$
8. Similar triangles: There exists a positive constant c such that for any triangle T and any $n \geq 3$, there is an n -element point set in the plane with at least cn^2 triples that induce triangles similar to T . For all quadrilaterals Q , whose points, as complex numbers, have an algebraic cross ratio, the maximum number of 4-tuples of an n -element set that induce quadrilaterals similar to Q is $\Theta(n^2)$. For all other quadrilaterals Q , this function is slightly subquadratic. The maximum number of pairwise homothetic triples in a set of n points in the plane is $O(n^{3/2})$, and this bound is asymptotically tight [EE94, LR97]. The number of similar tetrahedra among n points in three-dimensional space is at most $O(n^{2.2})$ [ATT98]. Further variants were studied in [Bra02].
9. Isosceles triangles, unit circles: In the plane, the maximum number of triples that determine an isosceles triangle, is $O(n^{2.102})$ [PT02]. The maximum number of distinct unit circles passing through at least 3 elements of a planar point set of size n is at least $\Omega(n^{3/2})$ and at most $n^2/3 - O(n)$ [Ele84].

CONJECTURES OF ERDŐS

1. The number of times the unit distance can occur among n points in the plane does not exceed $n^{1+c/\log \log n}$.
2. Any set of n points in the plane determines at least $\Omega(n/\sqrt{\log n})$ distinct distances.
3. Any set of n points in convex position in the plane has a point from which there are at least $\lfloor n/2 \rfloor$ distinct distances.
4. There is an integer $k \geq 4$ such that any finite set in convex position in the plane has a point from which there are no k points at the same distance.
5. Any set of n points in the plane, not all on a line, contains at least $n - 2$ triples that determine distinct angles (Corrádi, Erdős, Hajnal).

-
6. The diameter of any set of n points in the plane with the property that the set of all distances determined by them is separated (on the line) is at least $\Omega(n)$. Perhaps it is at least $n - 1$, with equality when the points are collinear.
 7. There is no set of n points everywhere dense in the plane such that all distances determined by them are rational (Erdős, Ulam).
-

1.3 COLORING PROBLEMS

If we partition a space into a small number of parts (i.e., we color its points with a small number of colors), at least one of these parts must contain certain “unavoidable” point configurations. In the simplest case, the configuration consists of a pair of points at a given distance. The prototype of such a question is the Hadwiger-Nelson problem: What is the minimum number of colors needed for coloring the plane so that no two points at unit distance receive the same color? The answer is known to be between 4 and 7.

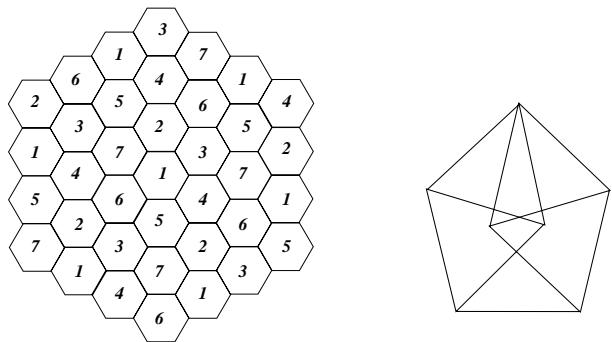


FIGURE 1.3.1

*The chromatic number of the plane is
(i) at most 7 and (ii) at least 4.*

GLOSSARY

Chromatic number of a graph: The minimum number of colors, $\chi(G)$, needed to color all the vertices of G so that no two vertices of the same color are adjacent.

List-chromatic number of a graph: The minimum number k such that for any assignment of a list of k colors to every vertex of the graph, for each vertex it is possible to choose a single color from its list so that no two vertices adjacent to each other receive the same color.

Chromatic number of a metric space: The chromatic number of the unit distance graph of the space, i.e., the minimum number of colors needed to color all points of the space so that no two points of the same color are at unit distance.

Polychromatic number of metric space: The minimum number of colors, χ , needed to color all points of the space so that for each color class C_i , there is

a distance d_i such that no two points of C_i are at distance d_i . A sequence of “forbidden” distances, (d_1, \dots, d_χ) , is called a *type* of the coloring. (The same coloring may have several types.)

Girth of a graph: The length of the shortest cycle in the graph.

A point configuration P is **k -Ramsey** in d -space if, for any coloring of the points of d -space with k colors, at least one of the color classes contains a congruent copy of P .

A point configuration P is **Ramsey** if, for every k , there exists $d(k)$ such that P is k -Ramsey in $d(k)$ -space.

Brick: The vertex set of a right parallelepiped.

FORBIDDEN DISTANCES

Table 1.3.1 contains the best bounds we know for the chromatic numbers of various spaces. All lower bounds can be established by showing that the corresponding unit distance graphs have some *finite* subgraphs of large chromatic number [dBEG51]. $S^{d-1}(r)$ denotes the sphere of radius r in d -space, where the distance between two points is the length of the chord connecting them.

TABLE 1.3.1 Estimates for the chromatic numbers of metric spaces.

SPACE	LOWER BOUND	UPPER BOUND	SOURCE
Line	2	2	
Plane	4	7	Nelson, Isbell
Rational points of plane	2	2	[Woo73]
3-space	6	15	[Nec02, Cou02, RT03]
Rational points of 3-space	2	2	Benda, Perles
$S^2(r), \frac{1}{2} \leq r \leq \frac{\sqrt{3}-\sqrt{3}}{2}$	3	4	[Sim75]
$S^2(r), \frac{\sqrt{3}-\sqrt{3}}{2} \leq r \leq \frac{1}{\sqrt{3}}$	3	5	Straus
$S^2(r), r \geq \frac{1}{\sqrt{3}}$	4	7	[Sim76]
$S^2\left(\frac{1}{\sqrt{2}}\right)$	4	4	[Sim76]
Rational points of 4-space	4	4	Benda, Perles
Rational points of 5-space	6	?	[Chi90]
d -space	$(1 + o(1))(1.2)^d$	$(3 + o(1))^d$	[FW81, LR72]
$S^{d-1}(r), r \geq \frac{1}{2}$	d	?	[Lov83]

Next we list several problems and results strongly related to the Hadwiger-Nelson problem (quoted in the introduction to this section).

1. 4-chromatic unit distance graphs of large girth: O’Donnell [O’D00] answered a question of Erdős by exhibiting a series of unit distance graphs in the plane with arbitrary large girths and chromatic number 4.
2. Polychromatic number: Stechkin and Woodall [Woo73] showed that the polychromatic number of the plane is between 4 and 6. It is known that for any $r \in [\sqrt{2}-1, 1/\sqrt{5}]$, there is a coloring of type $(1, 1, 1, 1, 1, r)$ [Soi94]. However,

the list-chromatic number of the unit distance graph of the plane, which is at least as large as its polychromatic number, is infinite [Alo93].

3. Dense sets realizing no unit distance: The *lower* (resp. *upper*) *density* of an unbounded set in the plane is the \liminf (resp. \limsup) of the ratio of the Lebesgue measure of its intersection with a disk of radius r around the origin to $r^2\pi$, as $r \rightarrow \infty$. If these two numbers coincide, their common value is called the ***density*** of the set. Let δ^d denote the maximum density of a planar set, no pair of points of which is at unit distance. Croft [Cro67] and Székely [Szé84] showed that $0.2293 \leq \delta^2 \leq 12/43$.
4. The graph of large distances: Let $G_i(P)$ denote the graph whose vertex set is a finite point set P , with two vertices connected by an edge if and only if their distance is one of the i largest distances determined by P . In the plane, $\chi(G_1(P)) \leq 3$ for every P ; see Borsuk's problem in the preceding section. It is also known that for any finite planar set, $G_i(P)$ has a vertex with fewer than $3i$ neighbors [ELV89]. Thus, $G_i(P)$ has fewer than $3in$ edges, and its chromatic number is at most $3i$. However, if $n > ci^2$ for a suitable constant $c > 0$, we have $\chi(G_i(P)) \leq 7$.

EUCLIDEAN RAMSEY THEORY

According to an old result of Gallai, for any finite d -dimensional point configuration P and for any coloring of d -space with finitely many colors, at least one of the color classes will contain a homothetic copy of P . The corresponding statement is false if, instead of a homothet, we want to find a *translate*, or even a *congruent copy*, of P . Nevertheless, for some special configurations, one can establish interesting positive results, provided that we color a sufficiently high-dimensional space with a sufficiently small number of colors. The Hadwiger-Nelson-type results discussed in the preceding subsection can also be regarded as very special cases of this problem, in which P consists of only two points. The field, known as “Euclidean Ramsey theory”, was started by a series of papers by Erdős, Graham, Montgomery, Rothschild, Spencer, and Straus [EGM⁺73, EGM⁺75a, EGM⁺75b].

For details, see [Chapter 11](#) of this Handbook.

OPEN PROBLEMS

1. (Erdős, Simmons) Is it true that the chromatic number of $S^{d-1}(r)$, the sphere of radius r in d -space, is equal to $d+1$, for every $r > 1/2$? In particular, does this hold for $d = 3$ and $r = 1/\sqrt{3}$?
2. (Sachs) What is the minimum number of colors, $\chi(d)$, sufficient to color any system of nonoverlapping unit balls in d -space so that no two balls that are tangent to each other receive the same color? Equivalently, what is the maximum chromatic number of a unit distance graph induced by a d -dimensional separated point set? It is easy to see [JR84] that $\chi(2) = 4$, and we also know that $5 \leq \chi(3) \leq 9$.
3. (Ringel) Does there exist any finite upper bound on the number of colors needed to color any system of (possibly overlapping) disks (of not necessarily

equal radii) in the plane so that no two disks that are tangent to each other receive the same color, provided that no three disks touch one another at the same point? If such a number exists, it must be at least 5.

4. (Graham) Is it true that any 3-element point set P that does not induce an equilateral triangle is 2-Ramsey in the plane? This is known to be false for equilateral triangles, and correct for right triangles (Shader). Is every 3-element point set P 3-Ramsey in 3-space? The answer is again in the affirmative for right triangles [BT96].
5. (Solymosi) Is it true that, if n is sufficiently large, then for any 2-coloring of all the $\binom{n}{2}$ segments connecting any set of n points in general position in the plane, there exists a monochromatic empty triangle? Note that, if in the Erdős-Klein-Szekeres problem (discussed in section 1.1 above), we have $e(6) < \infty$, then the answer to this question is in the affirmative, because for any 2-coloring of the edges of a complete graph with 6 vertices, there is a monochromatic triangle.

1.4 SOURCES AND RELATED MATERIAL

SURVEYS

These surveys discuss and elaborate many of the results cited above.

[PA95, Mat02]: Monographs devoted to combinatorial geometry.

[BMP04]: A representative survey of results and open problems in discrete geometry, originally started by the Moser brothers.

[Pac93]: A collection of essays covering a large area of discrete and computational geometry, mostly of some combinatorial flavor.

[HDK64]: A classical treatise of problems and exercises in combinatorial geometry, complete with solutions.

[KW91]: A collection of beautiful open questions in geometry and number theory, together with some partial answers organized into challenging exercises.

[EP95]: A survey full of original problems raised by the “founding father” of combinatorial geometry.

[JT95]: A collection of more than two hundred unsolved problems about graph colorings, with an extensive list of references to related results.

[Grü72]: A monograph containing many results and conjectures on configurations and arrangements.

RELATED CHAPTERS

[Chapter 4: Helly-type theorems and geometric transversals](#)

[Chapter 5: Pseudoline arrangements](#)

- Chapter 11: Euclidean Ramsey theory
 - Chapter 13: Geometric discrepancy theory and uniform distribution
 - Chapter 14: Topological methods
 - Chapter 24: Arrangements
-

REFERENCES

- [AE45] N.H. Anning and P. Erdős. Integral distances. *Bull. Amer. Math. Soc.*, 51:598–600, 1945.
- [AEP88] D. Avis, P. Erdős, and J. Pach. Repeated distances in space. *Graphs Combin.*, 4:207–217, 1988.
- [ÁF02] B.M. Ábrego and S. Fernández-Merchant. Convex polyhedra in \mathbb{R}^3 spanning $\Omega(n^{4/3})$ congruent triangles. *J. Combin. Theory Ser. A*, 98:406–409, 2002.
- [Alo93] N. Alon. Restricted colorings of graphs. In *Surveys in Combinatorics*, volume 187 of London Math. Soc. Lecture Note Ser., Cambridge University Press, 1993, pages 1–33.
- [AG86] N. Alon and E. Győri. The number of small semispaces of a finite set of points, *J. Combin. Theory Ser. A*, 41:154–157, 1986.
- [Alt63] E. Altman. On a problem of Erdős. *Amer. Math. Monthly*, 70:148–157, 1963.
- [AS01] P.K. Agarwal and M. Sharir. On the number of congruent simplices in a point set. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 2001, pages 1–9.
- [ATT98] T. Akutsu, H. Tamaki, and T. Tokuyama. Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. *Discrete Comput. Geom.*, 20:307–331, 1998.
- [Avi84] D. Avis. The number of furthest neighbour pairs in a finite planar set. *Amer. Math. Monthly*, 91:417–420, 1984.
- [BB94] A. Bálintová and V. Bálint. On the number of circles determined by n points in the Euclidean plane. *Acta Math. Hungar.*, 63:283–289, 1994.
- [BBS83] E. Bannai, E. Bannai, and D. Stanton. An upper bound on the cardinality of an s -distance subset in real Euclidean space II. *Combinatorica*, 3:147–152, 1983.
- [BGS74] S.A. Burr, B. Grünbaum, and N.J.A. Sloane. The orchard problem. *Geom. Dedicata*, 2:397–424, 1974.
- [BMP04] P. Brass, W.O.J. Moser, and J. Pach. *Research Problems in Discrete Geometry*. 2004.
- [Bra92a] P. Brass. *Beweis einer Vermutung von Erdős and Pach aus der kombinatorischen Geometrie*. Ph.D. dissertation, Dept. of Discrete Math., Technical University Braunschweig, 1992.
- [Bra92b] P. Brass. The maximum number of second smallest distances in finite planar sets. *Discrete Comput. Geom.*, 7:371–379, 1992.
- [Bra97] P. Brass. On the maximum number of unit distances among n points in dimension four. In I. Bárány and K. Böröczky, editors, *Intuitive Geometry (Budapest, 1995)*, Bolyai Soc. Math. Studies, 6:277–290, 1997.
- [Bra02] P. Brass. Combinatorial geometry problems in pattern recognition. *Discrete Comput. Geom.*, 28:495–510, 2002.
- [BRS01] P. Brass, G. Rote and K.J. Swanepoel. Triangles of extremal area or perimeter in a finite planar pointset. *Discrete Comput. Geom.*, 26:51–58, 2001.

- [BT96] M. Bóna and G. Tóth. A Ramsey-type problem on right-angled triangles in space. *Discrete Math.*, 150:61–67, 1996.
- [BV03] I. Bárány and P. Valtr. Planar point sets with a small number of empty convex polygons. *Studia Sci. Math. Hungar.*, to appear.
- [CCEG79] J.H. Conway, H.T. Croft, P. Erdős and M.J.T. Guy. On the distribution of values of angles determined by coplanar points. *J. London Math. Soc. II. Ser.*, 19:137–143, 1979.
- [CEG⁺90] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and surfaces. *Discrete Comput. Geom.*, 5:99–160, 1990.
- [Cha70] G.D. Chakerian. Sylvester’s problem on collinear points and a relative *Amer. Math. Monthly*, 77:164–167, 1970.
- [Chi90] K.B. Chilakamarri. On the chromatic number of rational five-space. *Aequationes Math.*, 39:146–148, 1990.
- [Cou02] D. Coulson. A 15-colouring of 3-space omitting distance one. *Discrete Math.*, 256:83–90, 2002.
- [Cro67] H.T. Croft. Incidence incidents. *Eureka*, 30:22–26, 1967.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [CS93] J. Csima and E. Sawyer. There exist $6n/13$ ordinary points. *Discrete Comput. Geom.*, 9:187–202, 1993.
- [Csi96] G. Csizmadia. Furthest neighbors in space. *Discrete Math.*, 150:81–88, 1996.
- [dBE51] N.G. de Bruijn and P. Erdős. A colour problem for infinite graphs and a problem in the theory of relations. *Nederl. Akad. Wetensch. Proc. Ser. A*, 54:371–373, 1951.
- [Dey98] T. Dey. Improved bounds for planar k -sets and related problems. *Discrete Comput. Geom.*, 19:373–382, 1998.
- [Dir51] G.A. Dirac. Collinearity properties of sets of points. *Quart. J. Math. Oxford Ser. (2)*, 2:221–227, 1951.
- [Dum00] A. Dumitrescu. Planar sets with few empty convex polygons. *Studia Sci. Math. Hungar.*, 36:93–109, 2000.
- [EE94] G. Elekes and P. Erdős. Similar configurations and pseudogrids. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 85–104. North-Holland, Amsterdam, 1994.
- [EFF91] P. Erdős, P. Fishburn, and Z. Füredi. Midpoints of diagonals of convex n -gons. *SIAM J. Discrete Math.*, 4:329–341, 1991.
- [EFPR93] P. Erdős, Z. Füredi, J. Pach, and I.Z. Ruzsa. The grid revisited. *Discrete Math.*, 111:189–196, 1993.
- [EGM⁺73] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J. Spencer, and E.G. Straus. Euclidean Ramsey theorems. I. *J. Combin. Theory*, 14:341–363, 1973.
- [EGM⁺75a] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J. Spencer, and E.G. Straus. Euclidean Ramsey theorems. II. In A. Hajnal, R. Rado, and V.T. Sós, editors, *Infinite and Finite Sets*, North-Holland, Amsterdam, 1975, pages 529–558.
- [EGM⁺75b] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J. Spencer, and E.G. Straus. Euclidean Ramsey theorems. III. In A. Hajnal, R. Rado, and V.T. Sós, editors, *Infinite and Finite Sets*, North-Holland, Amsterdam, 1975, pages 559–584.

- [EH90] H. Edelsbrunner and P. Hajnal. A lower bound on the number of unit distances between the points of a convex polygon. *J. Combin. Theory Ser. A*, 55:312–314, 1990.
- [EHP89] P. Erdős, D. Hickerson, and J. Pach. A problem of Leo Moser about repeated distances on the sphere. *Amer. Math. Monthly*, 96:569–575, 1989.
- [Ele84] G. Elekes. n points in the plane determine $n^{3/2}$ unit circles. *Combinatorica*, 4:131, 1984.
- [Ell67] P.D.T.A. Elliott. On the number of circles determined by n points. *Acta Math. Acad. Sci. Hungar.*, 18:181–188, 1967.
- [ELSS73] P. Erdős, L. Lovász, A. Simmons, and E.G. Straus. Dissection graphs of planar point sets. In G. Srivastava, editor, *A Survey of Combinatorial Theory*, North-Holland, Amsterdam, 1973, pages 139–149.
- [ELV89] P. Erdős, L. Lovász, and K. Vesztergombi. Colorings of circles. *Discrete Comput. Geom.*, 4:541–549, 1989.
- [EMP93] P. Erdős, E. Makai, and J. Pach. Nearly equal distances in the plane. *Combin. Probab. Comput.*, 2:401–408, 1993.
- [EP71] P. Erdős and G. Purdy. Some extremal problems in geometry. *J. Combin. Theory Ser. A*, 10:246–252, 1971.
- [EP90] P. Erdős and J. Pach. Variations on the theme of repeated distances. *Combinatorica*, 10:261–269, 1990.
- [EP95] P. Erdős and G. Purdy. Extremal problems in combinatorial geometry. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, North-Holland, Amsterdam, 1995, pages 809–874.
- [Erd43] P. Erdős. Problem 4065. *Amer. Math. Monthly*, 50:65, 1943.
- [Erd46] P. Erdős. On sets of distances of n points. *Amer. Math. Monthly*, 53:248–250, 1946.
- [Erd60] P. Erdős. On sets of distances of n points in Euclidean space. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:165–169, 1960.
- [Erd67] P. Erdős. On some applications of graph theory to geometry. *Canad. J. Math.*, 19:968–971, 1967.
- [Erd75] P. Erdős. On some problems of elementary and combinatorial geometry. *Ann. Mat. Pura Appl. Ser. IV*, 103:99–108, 1975.
- [ES35] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935.
- [ES89] H. Edelsbrunner and S. Skiena. On the number of furthest-neighbour pairs in a point set. *Amer. Math. Monthly*, 96:614–618, 1989.
- [ES61] P. Erdős and G. Szekeres. On some extremum problems in elementary geometry. *Ann. Univ. Sci. Budapest. Eötvös, Sect. Math.*, 3:53–62, 1960/61.
- [FP84] Z. Füredi and I. Palásti. Arrangements of lines with a large number of triangles. *Proc. Amer. Math. Soc.*, 92:561–566, 1984.
- [Für90] Z. Füredi. The maximum number of unit distances in a convex n -gon. *J. Combin. Theory Ser. A*, 55:316–320, 1990.
- [Für91] Z. Füredi. Maximal independent subsets in Steiner systems and in planar sets. *SIAM J. Discrete Math.*, 4:196–199, 1991.
- [FW81] P. Frankl and R.M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1:357–368, 1981.

- [Grü56] B. Grünbaum. A proof of Vázsonyi's conjecture. *Bull. Res. Council Israel, Sect. A*, 6:77–78, 1956.
- [Grü72] B. Grünbaum. *Arrangements and Spreads*, volume 10 of *CBMS Regional Conf. Ser. in Math.* Amer. Math. Soc., Providence, 1972.
- [Grü76] B. Grünbaum. New views on some old questions of combinatorial geometry. *Colloq. Internaz. Theorie Combin. (Roma, 1973), Tomo I*, 451–468, 1976.
- [GS84] B. Grünbaum and G.C. Shephard. Simplicial arrangements in projective 3-space. *Mitt. Math. Sem. Giessen*, 166:49–101, 1984.
- [Han65] S. Hansen. A generalization of a theorem of Sylvester on lines determined by a finite set. *Math. Scand.*, 16:175–180, 1965.
- [Han80] S. Hansen. On configurations in 3-space without elementary planes and on the number of ordinary planes. *Math. Scand.*, 47:181–194, 1980.
- [Har74] H. Harborth. Solution to problem 664A. *Elem. Math.*, 29:14–15, 1974.
- [Har78] H. Harborth. Konvexe Fünfecke in ebenen Punktmenzen. *Elem. Math.*, 34:116–118, 1978.
- [HDK64] H. Hadwiger, H. Debrunner, and V. Klee. *Combinatorial Geometry in the Plane*. Holt, Rinehart & Winston, New York, 1964.
- [Hep56] A. Heppes. Beweis einer Vermutung von A. Vázsonyi. *Acta Math. Acad. Sci. Hungar.*, 7:463–466, 1956.
- [Hor83] J.D. Horton. Sets with no empty 7-gon. *Canad. Math. Bull.*, 26:482–484, 1983.
- [HP34] H. Hopf and E. Pannwitz. Aufgabe nr. 167. *Jahresber. Deutsch. Math.-Verein*, 43:114, 1934.
- [Ism02] D. Ismailescu. Restricted point configurations with many collinear k -tuples. *Discrete Comput. Geom.*, 28:571–575, 2002.
- [Jam87] R. Jamison. Direction trees. *Discrete Comput. Geom.*, 2:249–254, 1987.
- [JR84] B. Jackson and G. Ringel. Colorings of circles. *Amer. Math. Monthly*, 91:42–49, 1984.
- [JT95] T.R. Jensen and B. Toft. *Graph Coloring Problems*. Wiley-Interscience, New York, 1995.
- [KK93] J. Kahn and G. Kalai. A counterexample to Borsuk's conjecture. *Bull. Amer. Math. Soc.*, 29:60–62, 1993.
- [KM58] L.M. Kelly and W.O.J. Moser. On the number of ordinary lines determined by n points. *Canad. J. Math.*, 10:210–219, 1958.
- [KR72] L.M. Kelly and R. Rottenberg. Simple points in pseudoline arrangements. *Pacific J. Math.*, 40:617–622, 1972.
- [KT04] N.H. Katz and G. Tardos. A new entropy inequality for the Erdős distance problem. In J. Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemp. Math.* Amer. Math. Soc., Providence, 2004.
- [Kup94] Y.S. Kupitz. On the maximal number of appearances of the minimal distance among n points in the plane. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 217–244. North-Holland, Amsterdam, 1994.
- [KW91] V. Klee and S. Wagon. *Old and New Unsolved Problems in Plane Geometry and Number Theory*. Math. Assoc. Amer., Washington, 1991.

- [Lov83] L. Lovász. Self-dual polytopes and the chromatic number of distance graphs on the sphere. *Acta Sci. Math. (Szeged)*, 45:317–323, 1983.
- [LR72] D.G. Larman and C.A. Rogers. The realization of distances within sets in Euclidean space. *Mathematika*, 19:1–24, 1972.
- [LR97] M. Laczkovich and I.Z. Ruzsa. The number of homothetic subsets. In R.L. Graham and J. Nešetřil, editors, *The Mathematics of Paul Erdős, II*, volume 14 of *Algorithms Combin.*, Springer-Verlag, Berlin, 1997, pages 294–302.
- [LT95] H. Lefmann and T. Thiele. Point sets with distinct distances. *Combinatorica*, 15:379–408, 1995.
- [Mat02] J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, New York, 2002.
- [Mot51] T. Motzkin. The lines and planes connecting the points of a finite set. *Trans. Amer. Math. Soc.*, 70:451–464, 1951.
- [Nec02] O. Nechushtan. On the space chromatic number. *Discrete Math.*, 256:499–507, 2002.
- [NPP⁺02] E. Nevo, J. Pach, R. Pinchasi, M. Sharir, and S. Smorodinsky. Lenses in arrangements of pseudo-circles and their applications. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 2002, pages 123–132.
- [O'D00] P. O'Donnell. Arbitrary girth, 4-chromatic unit distance graphs in the plane. II: Graph embedding. *Geombinatorics*, 9:180–193, 2000.
- [PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. Wiley-Intersci. Ser. Discrete Math. Optim., Wiley, New York, 1995.
- [Pac93] J. Pach, editor. *New Trends in Discrete and Computational Geometry*. Springer-Verlag, Berlin, 1993.
- [Pac03] J. Pach. Midpoints of segments induced by a point set. *Geombinatorics*, 13:98–105, 2003.
- [Pe85] G.W. Peck. On ‘ k -sets’ in the plane. *Discrete Math.*, 56:73–74, 1985.
- [PP00] J. Pach and R. Pinchasi. Bichromatic lines with few points. *J. Combin. Theory Ser. A*, 90:326–335, 2000.
- [PS90] J. Pach and M. Sharir. Repeated angles in the plane and related problems. *J. Combin. Theory Ser. A*, 59:12–22, 1990.
- [PT02] J. Pach and G. Tardos. Isosceles triangles determined by a planar point set. *Graphs Combin.*, 18:769–779, 2002.
- [Pu88] G. Purdy. Repeated angles in E^4 . *Discrete Comput. Geom.*, 3:73–75, 1988.
- [Reu72] O. Reutter. Problem 664A. *Elem. Math.*, 27:19, 1972.
- [RT03] R. Radoičić and G. Tóth. Note on the chromatic number of the space. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 695–698. Springer-Verlag, Berlin, 2003.
- [Sch88] O. Schramm. Illuminating sets of constant width. *Mathematika*, 35:180–199, 1988.
- [Sim75] G.J. Simmons. Bounds on the chromatic number of the sphere. In *Proc. 6th South-eastern Conf. on Combinatorics, Graph Theory, and Computing, Congr. Numer.* 14, 1975, pages 541–548.
- [Sim76] G.J. Simmons. The chromatic number of the sphere. *J. Austral. Math. Soc. Ser. A*, 21:473–480, 1976.
- [Soi94] A. Soifer. Six-realizable set x_6 . *Geombinatorics*, 3:140–145, 1994.

- [SST84] J. Spencer, E. Szemerédi, and W.T. Trotter. Unit distances in the Euclidean plane. In B. Bollobás, editor, *Graph Theory and Combinatorics*, Academic Press, London, 1984, pages 293–303.
- [SST01] M. Sharir, S. Smorodinsky, and G. Tardos. An improved bound for k -sets in three dimensions. *Discrete Comput. Geom.*, 26:195–204, 2001.
- [St44] R. Steinberg. Solution of problem 4065. *Amer. Math. Monthly*, 51:169–171, 1944. (Also contains a solution by T. Gallai in an editorial remark.)
- [ST01] J. Solymosi and C. Tóth. Distinct distances in the plane. *Discrete Comput. Geom.*, 25:629–634, 2001.
- [SV04a] J. Solymosi and V. Vu. Distinct distances in high-dimensional homogeneous sets. In J. Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemp. Math.* Amer. Math. Soc., Providence, 2004.
- [SV04b] K. Swanepoel and P. Valtr. The unit distance problem on spheres. In J. Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemp. Math.* Amer. Math. Soc., Providence, 2004.
- [Syl67] J.J. Sylvester. Problem 2473. *Educational Times*, 8:104–107, 1867.
- [Syl93] J.J. Sylvester. Mathematical question 11851. *Educational Times*, 46:156, 1893.
- [Szé84] L.A. Székely. Measurable chromatic number of geometric graphs and sets without some distances in Euclidean space. *Combinatorica*, 4:213–218, 1984.
- [Tót97] G. Tóth. The shortest distance among points in general position. *Comput. Geom. Theory Appl.*, 8:33–38, 1997.
- [Tót01] G. Tóth. Point sets with many k -sets. *Discrete Comput. Geom.*, 26:187–194, 2001.
- [TV98] G. Tóth and P. Valtr. Note on the Erdős-Szekeres theorem. *Discrete Comput. Geom.*, 19:457–459, 1998.
- [Ung82] P. Ungar. $2n$ noncollinear points determine at least $2n$ directions. *J. Combin. Theory Ser. A*, 33:343–347, 1982.
- [Val95] P. Valtr. On the minimum number of polygons in a planar point set. *Studia Sci. Math. Hungar.*, 30:155–163, 1995.
- [Ves78] K. Vesztergombi. On large distances in planar sets. *Discrete Math.*, 67:191–198, 1978.
- [vW99] P. van Wamelen. The maximum number of unit distances among n points in dimension four. *Beiträge Algebra Geom.*, 40:475–477, 1999.
- [Woo73] D.R. Woodall. Distances realized by sets covering the plane. *J. Combin. Theory*, 14:187–200, 1973.
- [WW88] P.R. Wilson and J.A. Wiseman. A Sylvester theorem for conic sections. *Discrete Comput. Geom.*, 3:295–305, 1988.
- [ŽV92] R.T. Živaljević and S. Vrećica. The colored Tverberg's problem and complexes of injective functions. *J. Combin. Theory Ser. A*, 61:309–318, 1992.

2 PACKING AND COVERING

Gábor Fejes Tóth

INTRODUCTION

The basic problems in the classical theory of packings and coverings, the development of which was strongly influenced by the geometry of numbers and by crystallography, are the determination of the densest packing and the thinnest covering with congruent copies of a given body K . Roughly speaking, the density of an arrangement is the ratio between the total volume of the members of the arrangement and the volume of the whole space. In Section 2.1 we define this notion rigorously and give an account of the known density bounds.

In Section 2.2 we consider packings in, and coverings of, bounded domains. Section 2.3 is devoted to multiple arrangements and their decomposability. In Section 2.4 we make a detour to spherical and hyperbolic spaces. In Section 2.5 we discuss problems concerning the number of neighbors in a packing, while in Section 2.6 we investigate some selected problems concerning lattice arrangements. We close in Section 2.7 with problems concerning packing and covering with sequences of convex sets.

2.1 DENSITY BOUNDS FOR ARRANGEMENTS IN E^d

GLOSSARY

Convex body: A compact convex set with nonempty interior. A convex body in the plane is called a **convex disk**. The collection of all convex bodies in d -dimensional Euclidean space \mathbb{E}^d is denoted by $\mathcal{K}(\mathbb{E}^d)$. The subfamily of $\mathcal{K}(\mathbb{E}^d)$ consisting of centrally symmetric bodies is denoted by $\mathcal{K}^*(\mathbb{E}^d)$.

Operations on $\mathcal{K}(\mathbb{E}^d)$: For a set A and a real number λ we set $\lambda A = \{x \mid x = \lambda a, a \in A\}$. λA is called a **homothetic copy** of A . The **Minkowski sum** $A + B$ of the sets A and B consists of all points $a + b$, $a \in A$, $b \in B$. The set $A - A = A + (-A)$ is called the **difference body** of A . B^d denotes the unit ball centered at the origin, and $A + rB^d$ is called the **parallel body** of A at distance r ($r > 0$). If $A \subset \mathbb{E}^d$ is a convex body with the origin in its interior, then the **polar body** A^* of A is $\{x \in \mathbb{E}^d \mid \langle x, a \rangle \leq 1 \text{ for all } a \in A\}$.

The **Hausdorff distance** between the sets A and B is defined by

$$d(A, B) = \inf \{\varrho \mid A \subset B + \varrho B^d, B \subset A + \varrho B^d\}.$$

Lattice: The set of all integer linear combinations of a particular basis of \mathbb{E}^d .

Lattice arrangement: The set of translates of a given set in \mathbb{E}^d by all vectors of a lattice.

Packing: A family of sets whose interiors are mutually disjoint.

Covering: A family of sets whose union is the whole space.

The **volume** (Lebesgue measure) of a measurable set A is denoted by $V(A)$. In the case of the plane we use the term **area** and the notation $a(A)$.

Density of an arrangement relative to a set: Let \mathcal{A} be an arrangement (a family of sets each having finite volume) and D a set with finite volume. The **inner density** $d_{\text{inn}}(\mathcal{A}|D)$, **outer density** $d_{\text{out}}(\mathcal{A}|D)$, and **density** $d(\mathcal{A}|D)$ of \mathcal{A} relative to D are defined by

$$d_{\text{inn}}(\mathcal{A}|D) = \frac{1}{V(D)} \sum_{A \in \mathcal{A}, A \subset D} V(A),$$

$$d_{\text{out}}(\mathcal{A}|D) = \frac{1}{V(D)} \sum_{A \in \mathcal{A}, A \cap D \neq \emptyset} V(A),$$

and

$$d(\mathcal{A}|D) = \frac{1}{V(D)} \sum_{A \in \mathcal{A}} V(A \cap D).$$

(If one of the sums on the right side is divergent, then the corresponding density is infinite.)

The **lower density** and **upper density** of an arrangement \mathcal{A} are given by the limits $d_-(\mathcal{A}) = \liminf_{\lambda \rightarrow \infty} d_{\text{inn}}(\mathcal{A}|\lambda B^d)$, $d_+(\mathcal{A}) = \limsup_{\lambda \rightarrow \infty} d_{\text{out}}(\mathcal{A}|\lambda B^d)$. If $d_-(\mathcal{A}) = d_+(\mathcal{A})$, then we call the common value the **density** of \mathcal{A} and denote it by $d(\mathcal{A})$. It is easily seen that these quantities are independent of the choice of the origin.

The **packing density** $\delta(K)$ and **covering density** $\vartheta(K)$ of a convex body (or more generally of a measurable set) K are defined by

$$\delta(K) = \sup \{d_+(\mathcal{P}) \mid \mathcal{P} \text{ is a packing of } \mathbb{E}^d \text{ with congruent copies of } K\}$$

and

$$\vartheta(K) = \inf \{d_-(\mathcal{C}) \mid \mathcal{C} \text{ is a covering of } \mathbb{E}^d \text{ with congruent copies of } K\}.$$

The **translational packing density** $\delta_T(K)$, **lattice packing density** $\delta_L(K)$, **translational covering density** $\vartheta_T(K)$, and **lattice covering density** $\vartheta_L(K)$ are defined analogously, by taking the supremum and infimum over arrangements consisting of translates of K and over lattice arrangements of K , respectively. It is obvious that in the definitions of $\delta_L(K)$ and $\vartheta_L(K)$ we can take maximum and minimum instead of supremum and infimum. By a theorem of Groemer, the same holds for the translational and for the general packing and covering densities.

Dirichlet cell: Given a set S of points in \mathbb{E}^d such that the distances between the points of S have a positive lower bound, the Dirichlet cell, also known as the **Voronoi cell**, associated to an element s of S consists of those points of \mathbb{E}^d that are closer to s than to any other element of S .

KNOWN VALUES OF PACKING AND COVERING DENSITIES

Apart from the obvious examples of space fillers, there are only a few specific bodies for which the packing or covering densities have been determined. The bodies for which the packing density is known are given in Table 2.1.1.

TABLE 2.1.1 Bodies K for which $\delta(K)$ is known.

BODY	AUTHOR	SEE
Circle	Thue	[Fej72, p. 58]
Parallel body of a rectangle	L. Fejes Tóth	[EGH89]
Intersection of two congruent circles	L. Fejes Tóth	[EGH89]
Centrally symmetric n -gon (algorithm in $O(n)$ time)	Mount and Silverman	[FK93c]
Ball in \mathbb{E}^3	Hales	[Hald]
Truncated rhombic dodecahedron	A. Bezdek	[Bez94]

We have $\delta(B^2) = \pi/\sqrt{12}$. The longstanding conjecture that $\delta(B^3) = \pi/\sqrt{18}$ has been confirmed recently by Hales. A packing of balls reaching this density is obtained by placing the centers at the vertices and face-centers of a cubic lattice. We discuss the sphere packing problem in the next section.

For the rest of the bodies in Table 2.1.1, the packing density can be given only by rather complicated formulas. We note that, with appropriate modification of the definition, the packing density of a set with infinite volume can also be defined. A. Bezdek and W. Kuperberg (see [FK93c]) showed that the packing density of an infinite circular cylinder is $\pi/\sqrt{12}$, that is, infinite circular cylinders cannot be packed more densely than their base. It is conjectured that the same statement holds for circular cylinders of any finite height.

A theorem of L. Fejes Tóth (see [Fej64, p. 163]) states that

$$\delta(K) \leq \frac{a(K)}{H(K)} \quad \text{for } K \in \mathcal{K}(\mathbb{E}^2), \quad (2.1.1)$$

where $H(K)$ denotes the minimum area of a hexagon containing K . This bound is best possible for centrally symmetric disks, and it implies that

$$\delta(K) = \delta_T(K) = \delta_L(K) = \frac{a(K)}{H(K)} \quad \text{for } K \in \mathcal{K}^*(\mathbb{E}^2).$$

The packing densities of the convex disks in Table 2.1.1 have been determined utilizing this relation.

It is conjectured that an inequality analogous to (2.1.1) holds for coverings, and this is supported by the following weaker result (see [Fej64, p. 167]):

Let $h(K)$ denote the maximum area of a hexagon contained in a convex disk K . Let \mathcal{C} be a covering of the plane with congruent copies of K such that no two copies of K cross. Then

$$d_-(\mathcal{C}) \geq \frac{a(K)}{h(K)}.$$

The convex disks A and B *cross* if both $A \setminus B$ and $B \setminus A$ are disconnected. As translates of a convex disk do not cross, it follows that

$$\vartheta_T(K) \geq \frac{a(K)}{h(K)} \quad \text{for } K \in \mathcal{K}(\mathbb{E}^2).$$

Again, this bound is best possible for centrally symmetric disks, and it implies that

$$\vartheta_T(K) = \vartheta_L(K) = \frac{a(K)}{h(K)} \quad \text{for } K \in \mathcal{K}^*(\mathbb{E}^2). \quad (2.1.2)$$

Based on this, Mount and Silverman gave an algorithm that determines $\vartheta_T(K)$ for a centrally symmetric n -gon in $O(n)$ time. Also the classical result $\vartheta(B^2) = 2\pi/\sqrt{27}$ of Kershner (see [Fej72, p. 58]) follows from this relation.

One could expect that the restriction to arrangements of translates of a set means a considerable simplification. However, this apparent advantage has not been exploited so far in dimensions greater than 2. On the other hand, the lattice packing density of some special convex bodies in \mathbb{E}^3 has been determined; see Table 2.1.2.

TABLE 2.1.2 Bodies $K \in \mathbb{E}^3$ for which $\delta_L(K)$ is known.

BODY	$\delta_L(K)$	AUTHOR
$\{x \mid x \leq 1, x_3 \leq \lambda\} \quad (\lambda \leq 1)$	$\pi(3 - \lambda^2)^{1/2}/6$	Chalk
$\{x \mid x_i \leq 1, x_1 + x_2 + x_3 \leq \lambda\}$	$\begin{cases} \frac{9 - \lambda^2}{9} & \text{for } 0 < \lambda \leq \frac{1}{2} \\ \frac{9\lambda(9 - \lambda^2)}{4(-\lambda^3 - 3\lambda^2 + 24\lambda - 1)} & \text{for } \frac{1}{2} \leq \lambda \leq 1 \\ \frac{9(\lambda^3 - 9\lambda^2 + 27\lambda - 3)}{8\lambda(\lambda^2 - 9\lambda + 27)} & \text{for } 1 \leq \lambda \leq 3 \end{cases}$	Whitworth
$\{x \mid \sqrt{(x_1)^2 + (x_2)^2} + x_3 \leq 1\}$	$\pi\sqrt{6}/9 = 0.8550332\dots$	Whitworth
Tetrahedron	$18/49 = 0.3673469\dots$	Hoyleman
Octahedron	$18/19 = 0.9473684\dots$	Minkowski
Dodecahedron	$(5 + \sqrt{5})/8 = 0.9045084\dots$	Betke and Henk
Icosahedron	$0.8363574\dots$	Betke and Henk
Cuboctahedron	$45/49 = 0.9183633\dots$	Hoyleman
Icosidodecahedron	$(45 + 17\sqrt{5})/96 = 0.8647203\dots$	Betke and Henk
Rhombic Cuboctahedron	$(16\sqrt{2} - 20)/3 = 0.8758056\dots$	Betke and Henk
Rhombic Icosidodecahedron	$(768\sqrt{5} - 1290)/531 = 0.8047084\dots$	Betke and Henk
Truncated Cube	$9(5 - 3\sqrt{2})/7 = 0.9737476\dots$	Betke and Henk
Truncated Dodecahedron	$(25 + 37\sqrt{5})/120 = 0.8977876\dots$	Betke and Henk
Truncated Icosahedron	$0.78498777\dots$	Betke and Henk
Truncated Cuboctahedron	$0.8493732\dots$	Betke and Henk
Truncated Icosidodecahedron	$(19 + 10\sqrt{5})/50 = 0.8272135\dots$	Betke and Henk
Truncated Tetrahedron	$207/304 = 0.6809210\dots$	Betke and Henk
Snub Cube	$0.787699\dots$	Betke and Henk
Snub Dodecahedron	$0.7886401\dots$	Betke and Henk

All results given in Table 2.1.2 are based on Minkowski's work on critical lattices of convex bodies and can be traced in [BH00]. We emphasize the following special

case: Gauss's result that $\delta_L(B^3) = \pi/\sqrt{18}$ is the special case $\lambda = 1$ of Chalk's theorem concerning the frustum of the ball. In [BH00] Betke and Henk gave an efficient algorithm for computing $\delta_L(K)$ for an arbitrary 3-polytope. As an application they calculated the lattice packing densities of all regular and Archimedean polytopes.

The list in [Table 2.1.2](#) can be augmented by additional bodies using the following observations.

It has been noticed by Chalk and Rogers that the relation $\delta_T(K) = \delta_L(K)$ ($K \in \mathcal{K}(\mathbb{E}^2)$) readily implies that for a cylinder C in \mathbb{E}^3 based on a convex disk K we have $\delta_L(C) = \delta_L(K)$. Thus, $\delta_L(C)$ is known if the lattice packing density of its base is known.

Next, we recall the observation of Minkowski (see [Rog64, p. 69]) that an arrangement \mathcal{A} of translates of a convex body K is a packing if and only if the arrangement of translates of the body $\frac{1}{2}(K - K)$ by the same vectors is a packing. This implies that, for $K \in \mathcal{K}(\mathbb{E}^d)$,

$$\delta_T(K) = 2^d \delta_T(K - K) \frac{V(K)}{V(K - K)} \quad \text{and} \quad \delta_L(K) = 2^d \delta_L(K - K) \frac{V(K)}{V(K - K)} \quad (2.1.3)$$

Generally, K is not uniquely determined by $K - K$; e.g., we have $K - K = B^d$ for every $K \subset \mathbb{E}^d$ that is a body of constant width 1, and the determination of $\delta_L(K)$ for such a body is reduced to the determination of $\delta_L(B^d)$, which is established for $d \leq 8$. We give the known values of $\delta_L(B^d)$, together with those of $\vartheta(B^d)$, in [Table 2.1.3](#). All results given there can be traced in [CS93].

TABLE 2.1.3 Known values of $\delta_L(B^d)$ and $\vartheta_L(B^d)$.

d	$\delta_L(B^d)$	AUTHOR	$\vartheta_L(B^d)$	AUTHOR
2	$\frac{\pi}{2\sqrt{3}}$	Lagrange	$\frac{2\pi}{3\sqrt{3}}$	Kershner
3	$\frac{\pi}{\sqrt{18}}$	Gauss	$\frac{5\sqrt{5}\pi}{24}$	Bambah
4	$\frac{\pi^2}{16}$	Korkin and Zolotarev	$\frac{2\pi^2}{5\sqrt{5}}$	Delone and Ryškov
5	$\frac{\pi^2}{15\sqrt{2}}$	Korkin and Zolotarev	$\frac{245\sqrt{35}\pi^2}{3888\sqrt{3}}$	Baranovskii and Ryškov
6	$\frac{\pi^3}{48\sqrt{3}}$	Blichfeldt		
7	$\frac{\pi^3}{105}$	Blichfeldt		
8	$\frac{\pi^4}{384}$	Blichfeldt		

THE KEPLER CONJECTURE

A remark of Kepler can be interpreted in modern terminology as the conjecture that $\delta(B^3) = \pi/\sqrt{18}$. Early research concerning Kepler's conjecture concentrated on two easier problems: proving the conjecture for special arrangements and giving

upper bounds for $\delta(B^3)$.

We mentioned Gauss's result that $\delta_L(B^3) = \pi/\sqrt{18}$. A stronger result establishing Kepler's conjecture for a restricted class of packings is due to A. Bezdek, W. Kuperberg and Makai [BKM91]. They proved that the conjecture holds for packings consisting of parallel strings of balls. A string of balls is a collection of equal balls whose centers are collinear and such that each of them touches two others. The best upper bound for $\delta(B^3)$ was given by Muder [Mud93], who proved that $\delta(B^3) \leq 0.773055$.

The first step toward the solution of Kepler's conjecture in its full generality was made in the early 1950's by L. Fejes Tóth (see [Fej72]). He considered weighted averages of the volumes of Dirichlet cells of a finite collection of balls in a packing. He showed that the Kepler conjecture holds if a particular weighted average of volumes involving not more than 13 cells is greater than or equal the volume of the rhombic dodecahedron circumscribed around a ball (this being the Dirichlet cell of a ball in the face-centered cubic lattice). His argument constitutes a program that, if realizable in principle, reduces Kepler's conjecture to an optimization problem in a finite number of variables. Later, in [Fej64], he suggested that with the use of computers $\delta(B^3)$ could be "approximated with great exactitude."

In 1990 W.-Y. Hsiang announced the solution of the Kepler conjecture. His approach is very similar to the program proposed by L. Fejes Tóth. Unfortunately, Hsiang's paper [Hsi93] contains significant gaps, so it cannot be accepted as a proof. Hsiang maintains his claim of having a proof. He gave more detail in [Hsi01]. The mathematical community lost interest in checking those details, however.

About the same time as Hsiang, Tom Hales also attacked the Kepler conjecture. His first attempt [Hal92] was a program based on the Delone subdivision of space, which is dual to the subdivision by Dirichlet cells. He modified his approach in several steps [Hal93, Hal97, Hal98, FH]. His final version, worked out in collaboration with his graduate student Ferguson in [FH], uses a subdivision that is a hybrid of certain Delone-type simplices and Dirichlet cells. With each ball B in a saturated packing of unit balls, an object, called a *decomposition star*, is associated, consisting of certain terahedra having the center of B as a common vertex together with parts of a modified Dirichlet cell of B . A complicated scoring rule is introduced that takes into account the volumes of the different parts of the decomposition star with appropriate weights. The score of a decomposition star in the face-centered cubic lattice is a certain number, which Hales takes to be 8 pts. The key property of the decomposition stars and the scoring rule is that the decomposition star of a ball B , as well as its score, depends only on balls lying in a certain neighborhood of B . From the mathematical point of view, the main step of the proof is the theorem that

The Kepler conjecture holds, provided the score of each decomposition star in a saturated packing of unit balls is at most 8 pts.

The task of proving this, which is an optimization problem in finitely many variables, has been carried out with the aid of computers. As Hales points out, there is hope that in the future such a problem "might eventually become an instance of a general family of optimization problems for which general optimization techniques exist". In the absence of such general techniques, manual procedures had to be used to guide the work of computers.

Computers are used in the proof in several ways. The topological structure of the decomposition stars is described by planar maps. A computer program enu-

merates around 5000 planar maps that have to be examined as potential counterexamples to the conjecture. Interval arithmetic is used to prove various inequalities. Nonlinear optimization problems are replaced by linear problems that dominate the original ones in order to apply linear programming methods. Even the organization of the few gigabytes of data is a difficult task.

It is safe to say that the 300-page proof, aided by computer calculations taking months, is one of the most complex proofs in the history of mathematics. Lagarias, who in [Lag02] extracts the common ideas of the programs of L. Fejes Tóth, W.-Y. Hsiang, and Hales and puts them into a general framework, finds that “the Hales–Ferguson proof, assumed correct, is a tour de force of nonlinear optimization.” No one has checked all details of the proof, and possibly no human being will ever check them. However, the general framework of the proof is sound and no errors have been detected so far; thus, it is largely accepted by the mathematical community.

EXISTENCE OF ECONOMICAL ARRANGEMENTS

Table 2.1.4 lists the known bounds establishing the existence of reasonably dense packings and thin coverings. When c appears in a bound without specification, it means a suitable constant characteristic of the specific bound. The proofs of most of these are nonconstructive. For constructive methods yielding slightly weaker bounds, as well as improvements for special convex bodies, see [Chapter 61](#).

TABLE 2.1.4 Bounds establishing the existence of dense packings and thin coverings.

No.	BOUND	AUTHOR	SEE
Bounds for general convex bodies in \mathbb{E}^d			
1	$\delta_L(K) \geq cd^{3/2}/4^d$ (d large)	Schmidt, Rogers, and Shephard	[Rog64]
2	$\vartheta_T(K) \leq d \ln d + d \ln \ln d + 5d$	Rogers	[Rog64, Theorem 3.2]
3	$\vartheta_L(K) \leq d \log_2 \ln d + c$	Rogers	[Rog64]
4	$\vartheta_L(B^d) \leq cd(\ln d)^{\log_2 \sqrt{2\pi e}}$	Rogers	[Rog64]
Bounds for centrally symmetric convex bodies in \mathbb{E}^d			
5	$\delta_L(K) \geq \zeta(d)/2^{d-1}$	Minkowski–Hlawka	[PA95, Theorem 7.7]
6	$\delta_L(K) \geq cd/2^d$ (d large)	Schmidt	[Rog64]
Bounds for general convex bodies in \mathbb{E}^2			
7	$\delta(K) \geq \sqrt{3}/2 = 0.8660\dots$	G. Kuperberg and W. Kuperberg	[PA95, Theorem 4.5]
8	$\vartheta(K) \leq 1.2281771\dots$	Ismailescu	[Ism98]
9	$\delta_L(K) \geq 2/3$	Fáry	[Fej72, p. 100]
10	$\vartheta_L(K) \leq 3/2$	Fáry	[Fej72, p. 100]
Bounds for centrally symmetric convex bodies in \mathbb{E}^2			
11	$\delta_L(K) \geq 0.892656\dots$	Tammela	[PA95]
12	$\vartheta_L(K) \leq 2\pi/\sqrt{27}$	L. Fejes Tóth	[Fej72, p. 103]

Bound 1 for the packing density of general convex bodies follows by combining Bound 6 with the relation (2.1.3) and the inequality $V(K - K) \leq \binom{2d}{d} V(K)$ of

Rogers and Shephard (see [Rog64, Theorem 2.4]). For $d \geq 3$ all methods establishing the existence of dense packings rely on the theory of lattices, thus providing the same lower bounds for $\delta(K)$ and $\delta_T(K)$ as for $\delta_L(K)$.

Gritzmann (see [PA95]) proved a bound similar to Bound 4 for a larger class of convex bodies:

$$\vartheta_L(K) \leq cd(\ln d)^{1+\log_2 e}$$

holds for a suitable constant c and for every convex body K in \mathbb{E}^d that has an affine image symmetric about at least $\log_2 \ln d + 4$ coordinate hyperplanes.

UPPER BOUNDS FOR $\delta(B^d)$ AND LOWER BOUNDS FOR $\vartheta(B^d)$

The packing and covering density of B^d is not known for $d \geq 3$. Asymptotically, the best upper bound known for $\delta(B^d)$ is

$$\delta(B^d) \leq 2^{-0.599d+o(d)} \quad (\text{as } d \rightarrow \infty), \quad (2.1.4)$$

given by Kabatjanskii and Levenštejn (see [CS93]). This bound is not obtained directly by the investigation of packings in \mathbb{E}^d but rather through studying the analogous problem in spherical geometry, where the powerful technique of linear programming bounds can be used (see Section 2.4). For low dimensions, Rogers's simplex bound

$$\delta(B^d) \leq \sigma_d \quad (2.1.5)$$

gives a better estimate (see [Rog64, Theorem 7.1]). Here, σ_d is the ratio between the total volume of the sectors of $d+1$ unit balls centered at the vertices of a regular simplex of edge 2 and the volume of the simplex.

Recently, Rogers's bound has been improved in low dimensions as well. On one hand, K. Bezdek [Bez02] extended the method of Rogers by investigating the surface area of the Voronoi regions, rather than their volume; on the other hand, Cohn and Elkies [CE03, Coh02] developed linear programming bounds that apply directly to sphere packings in \mathbb{E}^d . This latter method seems to be more powerful. In dimensions 8 and 24 the bounds in [CE03] differ from the conjectured values of $\delta(B^8)$ and $\delta(B^{24})$ only by factors of 1.000001 and 1.0007071, respectively. There is hope that the exact values of $\delta(B^8)$ and $\delta(B^{24})$ can be found using these methods.

Coxeter, Few, and Rogers (see [Rog64, Theorem 8.1]) proved a dual counterpart to Rogers's simplex bound:

$$\vartheta(B^d) \geq \tau_d,$$

where τ_d is the ratio between the total volume of the intersections of $d+1$ unit balls with the regular simplex of edge $\sqrt{2(d+1)/d}$ if their centers lie at the vertices of the simplex, and the volume of the simplex. Asymptotically,

$$\tau_d \sim d/e^{3/2}.$$

In contrast to packings, where there is a sizable gap between bound (2.1.4) and the bound from the other direction (Bound 6 in Table 2.1.4), this bound compares quite favorably with the corresponding Bound 2 in Table 2.1.4.

According to a result of W. Schmidt (see [FK93c]), we have $\delta(K) < 1$ and $\vartheta(K) > 1$ for every smooth convex body; but the method of proof does not allow one to derive any explicit bound. There is a general upper bound for $\delta(K)$ that is

nontrivial (smaller than 1) for a wide class of convex bodies [FK93a]. It is quite reasonable for “longish” bodies. For cylinders in \mathbb{E}^d , the bound is asymptotically equal to the Kabatjanskiĭ and Levenšteĭn bound for B^d (as $d \rightarrow \infty$). We note that no nontrivial bound is known for $\vartheta(K)$ for any K other than a ball.

REGULARITY OF OPTIMAL ARRANGEMENTS

The packings and coverings attaining the packing and covering densities of a set are, of course, not uniquely determined, but it is a natural question whether there exist among the optimal arrangements some that satisfy certain regularity properties. Of particular interest are those bodies for which the densest packing and/or thinnest covering with congruent copies can be realized by a lattice arrangement. As mentioned above, $\delta(K) = \delta_L(K)$ for $K \in \mathcal{K}^*(\mathbb{E}^2)$. A plausible interpretation of this result is that the assumption of maximum density creates from a chaotic structure a regular one. Unfortunately, certain results indicate that such bodies are rather exceptional.

Let \mathcal{L}_p and \mathcal{L}_c be the classes of those convex disks $K \in \mathcal{K}(\mathbb{E}^2)$ for which $\delta(K) = \delta_L(K)$ and $\vartheta(K) = \vartheta_L(K)$, respectively. Then, in the topology induced by the Hausdorff metric on $\mathcal{K}(\mathbb{E}^2)$, the sets \mathcal{L}_p and \mathcal{L}_c are nowhere dense [FZ94, Fej95]. It is conjectured that an analogous statement holds also in higher dimensions.

Rogers [Rog64, p. 15] conjectures that for sufficiently large d we have $\delta(B^d) > \delta_L(B^d)$. The following result of A. Bezdek and W. Kuperberg (see [FK93c]) supports this conjecture: For $d \geq 3$ there are ellipsoids E in \mathbb{E}^d for which $\delta(E) > \delta_L(E)$. An even more surprising result holds for coverings [FK95]: For $d \geq 3$ every strictly convex body K in \mathbb{E}^d has an affine image K' such that $\vartheta(K') < \vartheta_L(K')$. In particular, there is an ellipsoid E in \mathbb{E}^3 for which

$$\vartheta(E) < 1.394 < \frac{3\sqrt{3}}{2}(3 \operatorname{arcsec} 3 - \pi) = \tau_3 \leq \vartheta_T(E) \leq \vartheta_L(E).$$

We note that no example of a convex body K is known for which $\delta_L(K) < \delta_T(K)$ or $\vartheta_L(K) > \vartheta_T(K)$.

Schmitt [Sch88] constructed a star-shaped prototile for a monohedral tiling in \mathbb{E}^3 such that no tiling with its replicas is periodic. It is not known whether a convex body with this property exists; however, with a slight modification of Schmitt’s construction, Conway produced a convex prototile that admits only nonperiodic tilings if no mirror-image is allowed (see Section 3.4). Another result of Schmitt’s [Sch91] is that there are star-shaped sets in the plane whose densest packing cannot be realized in a periodic arrangement.

2.2 FINITE ARRANGEMENTS

PACKING IN AND COVERING OF A BODY WITH GIVEN SHAPE

What is the size of the smallest square tray that can hold n given glasses? Thue’s result gives a bound that is asymptotically sharp as $n \rightarrow \infty$; however, for practical reasons, small values of n are of interest.

Generally, for given sets K and C and a positive integer n one can ask for the quantities

$$M_p(K, C, n) = \inf\{\lambda \mid n \text{ congruent copies of } C \text{ can be packed in } \lambda K\}$$

and

$$M_c(K, C, n) = \sup\{\lambda \mid n \text{ congruent copies of } C \text{ can cover } \lambda K\}.$$

[Tables 2.2.1](#) and [2.2.2](#) contain the known results about the cases when C is a circle and K is a circle, square, or regular triangle. In addition, economical circle packings and circle coverings have been constructed for many special values of n . All of these results can be traced in [Fod99, Fod00, Fod, HM97, Mel93, Mel94, Mel97, Pei94]. Concerning the thinnest covering of a circle with congruent circles, we mention the conjecture that $M_c(K, B^2, n) = 1 + 2 \cos \frac{2\pi}{n-1}$ for $8 \leq n \leq 10$. The paper of Krotoszyński [Kro93] claims this as a theorem even for $n \leq 11$; however, his proof contains a gap. In fact, Melissen and Schuur have found an example showing that the result does not hold for $n = 11$.

Most of these results were obtained by ad hoc methods. Recently, however, Peikert described a heuristic algorithm for the determination of $M_p(K, B^2, n)$ and the corresponding optimal arrangements in the case where K is the unit square. His algorithm consists of the following steps:

Step 1. Find a good upper bound m for $M_p(K, B^2, n)$. This requires the construction of a reasonably good arrangement, which can be established, e.g., by the Monte Carlo method.

Step 2. Iterate an elimination process on a successively refined grid to restrict possible locations for the centers of a packing of unit circles in mK .

Step 3. Based on the result of Step 2, guess the nerve graph of the packing, then determine the optimal packing with the given graph.

Step 4. Verify that the arrangement obtained in Step 3 is indeed optimal.

Peikert does not prove that these steps always provide the optimal arrangement in finite time, but he implemented the method successfully for $n \leq 20$. The best arrangements are shown in Figure 2.2.1. Observe that quite often an optimal arrangement can contain a freely movable circle.

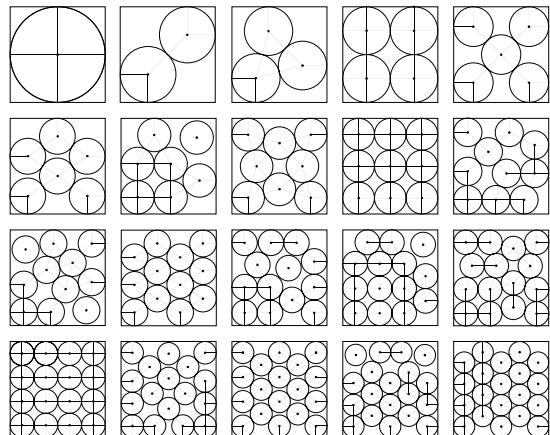


FIGURE 2.2.1
Densest packing of $n \leq 20$ equal circles in a square.

 TABLE 2.2.1 Packing of congruent circles in circles, squares, and equilateral triangles.

K	n	$M_p(K, B^2, n)$	AUTHOR
B^2	2	2	(elementary)
	3	2.154700538 ...	(elementary)
	4	2.414213562 ...	(elementary)
	5	2.701301617 ...	(elementary)
	6	3	(elementary)
	7	3	(elementary)
	8	3.304764871 ...	Pirl
	9	3.61312593 ...	Pirl
	10	3.813898249 ...	Pirl
	11	3.9238044 ...	Melissen
	12	4.02960193 ...	Fodor
	13	4.23606797 ...	Fodor
	19	4.86370330 ...	Fodor
Unit square	2	3.414213562 ...	(elementary)
	3	3.931851653 ...	(elementary)
	4	4	(elementary)
	5	4.828427125 ...	(elementary)
	6	5.328201177 ...	Graham, Melissen
	7	5.732050807 ...	Schaer
	8	5.863703305 ...	Schaer and Meir
	9	6	Schaer
	10	6.747441523 ...	Peikert
	11	7.022509506 ...	Peikert
	12	7.144957554 ...	Peikert
	13	7.463047839 ...	Peikert
	14	7.732050808 ...	Wengerodt
	15	7.863703305 ...	Peikert
	16	8	Wengerodt
	17	8.532660354 ...	Peikert
	18	8.656402355 ...	Peikert
	19	8.907460939 ...	Peikert
	20	8.978083353 ...	Peikert
	25	10	Wengerodt
	36	12	Kirchner and Wengerodt
Regular triangle of side 1	2	5.464101615 ...	(elementary)
	3	5.464101615 ...	(elementary)
	4	6.92820323 ...	Melissen
	5	7.464101615 ...	Melissen
	6	7.464101615 ...	Ohler, Groemer
	7	8.92820323 ...	Melissen
	8	9.293810046 ...	Melissen
	9	9.464101615 ...	Melissen
	10	9.464101615 ...	Ohler, Groemer
	11	10.73008794 ...	Melissen
	12	10.92820323 ...	Melissen
	$k(k+1)/2$	$2(k + \sqrt{3} - 1)$	Ohler, Groemer

The sequence $M_p(K, B^2, n)$ seems to be strictly increasing when K is a square or when K is a circle and $n \geq 7$. In contrast to this, it is conjectured that in

TABLE 2.2.2 Covering circles, squares, and equilateral triangles with congruent circles.

K	n	$M_c(K, B^2, n)$	AUTHOR
B^2	2	2	(elementary)
	3	$2/\sqrt{3}$	(elementary)
	4	$\sqrt{2}$	(elementary)
	5	1.64100446 ...	K. Bezdek
	6	1.7988 ...	K. Bezdek
	7	2	(elementary)
Unit square	2	$4\sqrt{5}/5$	(elementary)
	3	1.984555 ...	Heppes and Melissen
	4	$2\sqrt{2}$	Heppes and Melissen
	5	3.065975 ...	Heppes and Melissen
	7	3.6457524 ...	Heppes and Melissen
Regular triangle of side 1	2	2	(elementary)
	3	$2\sqrt{3}$	Melissen
	4	$2 + \sqrt{3}$	Melissen
	5	4	Melissen
	6	$\sqrt{27}$	Melissen

the case where K is a triangle, we have $M_p(K, B^2, n) = M_p(K, B^2, n - 1)$ for all triangular numbers $n = k(k + 1)/2$ ($k > 1$).

The problem of finding the densest packing of n congruent circles in a circle has been considered also in the Minkowski plane. In terms of Euclidean geometry, this is the same as asking for the smallest number $\varrho(n, K)$ such that n mutually disjoint translates of the centrally symmetric convex disk K (the unit circle in the Minkowski metric) can be contained in $\varrho(n, K)K$. Doyle, Lagarias, and Randell [DLR92] solved the problem for all $K \in \mathcal{K}^*(\mathbb{E}^2)$ and $n \leq 7$. There is an n -gon inscribed in K having equal sides in the Minkowski metric (generated by K) and having a vertex at an arbitrary boundary point of K . Let $\alpha(n, K)$ be the maximum Minkowski side-length of such an n -gon. Then we have $\varrho(n, K) = 1 + 2/\alpha(n, K)$ for $2 \leq n \leq 6$ and $\varrho(7, K) = \varrho(6, K) = 3$.

The densest packing of n congruent balls in a cube is known for $n \leq 10$ (see [Sch94]). The problem of finding the densest packing of congruent balls in other regular polytopes has been investigated by K. Bezdek (see [CFG91]).

SAUSAGE CONJECTURES

Intensive research on another type of finite packing and covering problem has been generated by the sausage conjectures of L. Fejes Tóth and Wills (see [GW93]):

What is the convex body of minimum volume in \mathbb{E}^d that can accommodate k nonoverlapping unit balls?

What is the convex body of maximum volume in \mathbb{E}^d that can be covered by k unit balls?

According to the conjectures mentioned above, for $d \geq 5$ the extreme bodies are “sausages” and in the optimal arrangements the centers of the balls are equally spaced on a line segment (Figure 2.2.2).

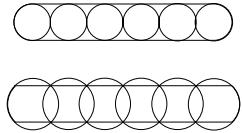


FIGURE 2.2.2
Sausage-like arrangements of circles.

After several partial results supporting these conjectures (see [GW93]) the breakthrough concerning the sausage conjecture for ball packings was achieved by Betke, Henk, and Wills [BHW94]: they proved that the conjecture holds for dimensions $d \geq 13387$. Later, Betke and Henk [BH98] improved the bound on d to $d \geq 42$.

Several generalizations of the problems mentioned above have been considered. Connections of these types of problems to the classical theory of packing and coverings, as well as to crystallography, have been observed. For details we refer to [Bör].

THE COVERING PROBLEMS OF BORSUK AND HADWIGER-LEVI

In 1933, Borsuk formulated the conjecture that any bounded set in \mathbb{E}^d can be partitioned into $d + 1$ subsets of smaller diameter. Borsuk verified the conjecture for $d = 2$, and the three-dimensional case was settled independently by Eggleston, Grünbaum, and Heppes. The conjecture is known to be true also for many special cases: for smooth convex bodies (Hadwiger), for centrally symmetric sets (Rissling), as well as for sets having the symmetry group of the regular simplex (Rogers). Quite recently, however, Kahn and Kalai [KK93] showed that Borsuk's conjecture is false in the following very strong sense: Let $b(d)$ denote the smallest integer such that every bounded set in \mathbb{E}^d can be partitioned into $b(d)$ subsets of smaller diameter. Then $b(d) \geq (1.2)^{\sqrt{d}}$ for every sufficiently large value of d .

In the 1950s, Hadwiger and Levi, independently of each other, asked for the smallest integer $h(K)$ such that the convex body K can be covered by $h(K)$ smaller positively homothetic copies of K . Hadwiger conjectured that $h(K) \leq 2^d$ for all $K \in \mathcal{K}(\mathbb{E}^d)$ and that equality holds only for parallelotopes. Levi verified the conjecture for the plane, but it is open for $d \geq 3$. Lassak proved Hadwiger's conjecture for centrally symmetric convex bodies in \mathbb{E}^3 , and K. Bezdek extended Lassak's result to convex polytopes with any affine symmetry.

Boltjanskiĭ observed that the Hadwiger-Levi covering problem for convex bodies is equivalent to an illumination problem. We say that a boundary point x of the convex body K is **illuminated from the direction u** if the ray issuing from x in the direction u intersects the interior of K . Let $i(K)$ be the minimum number of directions from which the boundary of K can be illuminated. Then $h(K) = i(K)$ for every convex body. For literature and further results concerning the Hadwiger-Levi problem, we refer to [Bez93].

2.3 MULTIPLE ARRANGEMENTS

GLOSSARY

k -fold packing: An arrangement \mathcal{A} such that each point of the space belongs to the interior of at most k members of \mathcal{A} .

***k*-fold covering:** An arrangement \mathcal{A} such that each point of the space belongs to at least k members of \mathcal{A} .

Densities: In analogy to the packing and covering densities of a body K , we define the quantities $\delta^k(K)$, $\delta_T^k(K)$, $\delta_L^k(K)$, $\vartheta^k(K)$, $\vartheta_T^k(K)$, and $\vartheta_L^k(K)$ as the suprema of the densities of all k -fold packings and the infima of the densities of all k -fold coverings with congruent copies, translates, and lattice translates of K , respectively.

TABLE 2.3.1 Bounds for k -fold packing and covering densities.

BOUND	AUTHOR
$\delta_T^k(K) \geq ck \quad K \in \mathcal{K}(\mathbb{E}^d)$	Erdős and Rogers
$\vartheta_L^k(K) \leq ((k+1)^{1/d} + 8d)^d \quad K \in \mathcal{K}(\mathbb{E}^d)$	Cohn
$\delta_L^k(K) \geq k - ck^{2/5} \quad K \in \mathcal{K}(\mathbb{E}^2)$	Bolle
$\vartheta_L^k(K) \leq k + ck^{2/5} \quad K \in \mathcal{K}(\mathbb{E}^2)$	Bolle
$\delta^k(B^d) \geq (2k/(k+1))^{d/2} \delta(B^d)$	Few
$\delta_L^k(B^d) \geq (2k/(k+1))^{d/2} \delta_L(B^d)$	Few
$\delta^k(B^d) \leq (1+d^{-1})((d+1)^k - 1)(k/(k+1))^{d/2}$	Few
$\delta^2(B^d) \leq \frac{4}{3}(d+2)(\frac{2}{3})^{d/2}$	Few
$\vartheta^k(B^d) \geq ck \quad c = c_d > 1$	G. Fejes Tóth
$\delta^k(B^2) \leq \frac{\pi}{6} \cot \frac{\pi}{6k}$	G. Fejes Tóth
$\vartheta^k(B^2) \geq \frac{\pi}{3} \csc \frac{\pi}{3k}$	G. Fejes Tóth

The information known about the asymptotic behavior of k -fold packing and covering densities is summarized in Table 2.3.1. There, in the various bounds, different constants appear, all of which we denote by c . All results given in the table can be traced in [EGH89] and [Fej83].

The known values of $\delta_L^k(B^d)$ and $\vartheta_L^k(B^d)$ (for $k \geq 2$) are given in [Table 2.3.2](#) and can be traced in [EGH89, Fej83, FK93c, Tem94a, Tem94b].

Recently, general methods for the determination of the densest k -fold lattice packings and the thinnest k -fold lattice coverings with circles have been developed by Horváth, Temesvári, and Yakovlev and by Temesvári, respectively (see [FK93c]).

These methods reduce both problems to the determination of the optima of finitely many well-defined functions of one variable. The proofs readily provide algorithms for finding the optimal arrangements; however, the authors did not try to implement them. Only the values of $\delta_L^9(B^2)$ and $\vartheta_L^8(B^2)$ have been added in this way to the list of values of $\delta_L^k(B^2)$ and $\vartheta_L^k(B^2)$ that had been determined previously by ad hoc methods.

We note that we have $\delta_L^k(B^2) = k\delta_L(B^2)$ for $k \leq 4$ and $\vartheta_L^2(B^2) = 2\vartheta_L(B^2)$. These are the only cases where the extreme multiple arrangements of circles are not better than repeated simple arrangements. These relations have been extended to arbitrary centrally symmetric convex disks by Dumir and Hans-Gill and by G. Fejes Tóth (see [FK93c]). There is a simple reason for the relations $\delta_L^3(K) = 3\delta_L(K)$ and $\delta_L^4(K) = 4\delta_L(K)$ ($K \in \mathcal{K}^*(\mathbb{E}^2)$): Every 3-fold lattice packing of the plane with a centrally symmetric disk is the union of 3 simple lattice packings and every 4-fold packing is the union of two 2-fold packings.

TABLE 2.3.2 Known values of $\delta_L^k(B^d)$ and $\vartheta_L^k(B^d)$.

RESULT	AUTHOR
$\delta_L^2(B^2) = \frac{\pi}{\sqrt{3}}$	Heppes
$\delta_L^3(B^2) = \frac{\sqrt{3}\pi}{2}$	Heppes
$\delta_L^4(B^2) = \frac{2\pi}{\sqrt{3}}$	Heppes
$\delta_L^5(B^2) = \frac{4\pi}{\sqrt{7}}$	Szirucek, Blunden
$\delta_L^6(B^2) = \frac{35\pi}{8\sqrt{6}}$	Blunden
$\delta_L^7(B^2) = \frac{8\pi}{\sqrt{15}}$	Blunden, Krejcarek, Bolle
$\delta_L^8(B^2) = \frac{3969\pi}{4\sqrt{220 - 2\sqrt{193}}\sqrt{449 + 32\sqrt{193}}}$	Bolle, Yakovlev
$\delta_L^9(B^2) = \frac{25\pi}{2\sqrt{21}}$	Temesvári
$\delta_L^2(B^3) = \frac{8\pi}{9\sqrt{3}}$	Few and Kanagasabapathy
$\vartheta_L^2(B^2) = \frac{4\pi}{3\sqrt{3}}$	Blunden
$\vartheta_L^3(B^2) = \frac{\pi\sqrt{27138 + 2910\sqrt{97}}}{216}$	Blunden
$\vartheta_L^4(B^2) = \frac{25\pi}{18}$	Blunden
$\vartheta_L^5(B^2) = \frac{32\pi}{7\sqrt{7}}$	Subak, Temesvári
$\vartheta_L^6(B^2) = \frac{98\pi}{27\sqrt{3}}$	Subak, Temesvári
$\vartheta_L^7(B^2) = 7.672\dots$	Haas, Temesvári
$\vartheta_L^8(B^2) = \frac{32\pi}{3\sqrt{15}}$	Temesvári
$\vartheta_L^2(B^3) = \frac{8\pi}{\sqrt{3}\sqrt{76\sqrt{6} - 159}}$	Few

This last observation brings us to the topic of decompositions of multiple arrangements. Our goal here is to find insight into the structure of multiple arrangements by decomposing them into possibly a few simple ones. Pach showed (see [FK93c]) that any double packing with positively homothetic copies of a convex disk can be decomposed into 4 simple packings. Further, if \mathcal{P} is a k -fold packing with convex disks such that for some integer L the inradius $r(K)$ and the area $a(K)$ of each member K of \mathcal{P} satisfy the inequality $9\pi^2kr^2(K)/a(K) \leq L$, then \mathcal{P} can be decomposed into L simple packings.

Concerning the decomposition of multiple coverings, Pach proved (see [FK93c]) that for any centrally symmetric polygon P and positive integer r there exists an integer $k = k(P, r)$ such that every k -fold covering with translates of P can be decomposed into r coverings. The attempt to extend this result by an approximation argument to all centrally symmetric disks fails, since, for fixed r , $k(P, r)$ approaches

infinity as the number of sides of P tends to infinity. For circle coverings, however, Mani and Pach (see [FK93c]) were able to establish a decomposition theorem: Every 33-fold covering with congruent circles can be decomposed into two coverings. In 3-space, results analogous to the two theorems above do not hold.

2.4 PROBLEMS IN NONEUCLIDEAN SPACES

Research on packing and covering in spherical and hyperbolic spaces has been concentrated on arrangements of balls. In contrast to spherical geometry, where the finite, combinatorial nature of the problems, as well as applications, have inspired research, investigations in hyperbolic geometry have been hampered by the lack of a reasonable notion of density relative to the whole hyperbolic space.

SPHERICAL SPACE

Let $M(d, \varphi)$ be the maximum number of caps of spherical diameter φ forming a packing on the d -dimensional spherical space \mathbb{S}^d , that is, on the boundary of B^{d+1} , and let $m(d, \varphi)$ be the minimum number of caps of spherical diameter φ covering \mathbb{S}^d . An upper bound for $M(d, \varphi)$, which is sharp for certain values of d and φ and yields the best estimate known as $d \rightarrow \infty$, is the so-called *linear programming bound* (see [CS93, pp. 257–266]). It establishes a surprising connection between $M(d, \varphi)$ and the expansion of real polynomials in terms of certain Jacobi polynomials. The Jacobi polynomials, $P_i^{(\alpha, \beta)}(x)$, $i = 0, 1, \dots, \alpha > -1, \beta > -1$, form a complete system of orthogonal polynomials on $[-1, 1]$ with respect to the weight function $(1-x)^\alpha(1+x)^\beta$. Set $\alpha = \beta = (d-1)/2$ and let

$$f(t) = \sum_{i=0}^k f_i P_i^{(\alpha, \alpha)}(t)$$

be a real polynomial such that $f_0 > 0$, $f_i \geq 0$ ($i = 1, 2, \dots, k$), and $f(t) \leq 0$ for $-1 \leq t \leq \cos \varphi$. Then

$$M(d, \varphi) \leq f(1)/f_0.$$

With the use of appropriate polynomials Kabatjanskiĭ and Levenštein (see [CS93]) obtained the asymptotic bound:

$$\frac{1}{d} \ln M(d, \varphi) \leq \frac{1 + \sin \varphi}{2 \sin \varphi} \ln \frac{1 + \sin \varphi}{2 \sin \varphi} - \frac{1 - \sin \varphi}{2 \sin \varphi} \ln \frac{1 - \sin \varphi}{2 \sin \varphi} + o(1).$$

This implies the simpler bound

$$M(d, \varphi) \leq (1 - \cos \varphi)^{-d/2} 2^{-0.099d+o(d)} \quad (\text{as } d \rightarrow \infty, \varphi \leq \varphi^* = 62.9974\dots).$$

Bound (2.1.4) for $\delta(B^d)$ follows in the limiting case when $\varphi \rightarrow 0$.

The following is a list of some special values of d and φ for which the linear programming bound turns out to be exact (see [CS93]).

$$\begin{array}{lll}
M(2, \arccos 1/\sqrt{5}) = 12 & M(4, \arccos 1/5) = 16 & M(5, \arccos 1/4) = 27 \\
M(6, \arccos 1/3) = 56 & M(7, \pi/3) = 240 & M(20, \arccos 1/9) = 112 \\
M(20, \arccos 1/7) = 162 & M(21, \arccos 1/11) = 100 & M(21, \arccos 1/6) = 275 \\
M(21, \arccos 1/4) = 891 & M(22, \arccos 1/5) = 552 & M(22, \arccos 1/3) = 4600 \\
M(23, \pi/3) = 196560
\end{array}$$

For small values of d and specific values of φ the linear programming bound is superseded by the “simplex bound” of Böröczky (see [FK93c]), which is the generalization of Rogers’s bound (2.1.5) for ball packings in \mathbb{S}^d .

The value of $M(d, \varphi)$ has been determined for all d and $\varphi \geq \pi/2$ (see [CS93]). We have

$$\begin{aligned}
M(d, \varphi) &= i + 1 \quad \text{for } \frac{1}{2}\pi + \arcsin \frac{1}{i+1} < \varphi \leq \frac{1}{2}\pi + \arcsin \frac{1}{i}, \quad i = 1, \dots, d, \\
M(d, \varphi) &= d + 2 \quad \text{for } \frac{1}{2}\pi < \varphi \leq \frac{1}{2}\pi + \arcsin \frac{1}{d+1},
\end{aligned}$$

and

$$M(d, \frac{1}{2}\pi) = 2(d + 1).$$

Except for an upper bound on $m(d, \varphi)$ establishing the existence of reasonably economic coverings of \mathbb{S}^d by equal balls due to Rogers (see [Fej83]), no results on coverings in spherical spaces of high dimensions are known.

Extensive research has been done on circle packings and circle coverings on \mathbb{S}^2 . Traditionally, here the inverse functions of $M(2, \varphi)$ and $m(2, \varphi)$ are considered. Let a_n be the maximum number such that n caps of spherical diameter a_n can form a packing and let A_n be the minimum number such that n caps of spherical diameter A_n can form a covering on \mathbb{S}^2 . The known values of a_n and A_n are given in Table 2.4.1. All the results mentioned in the table can be traced in [Fej72]. In addition, conjecturally best circle packings and circle coverings for $n \leq 130$, as well as good arrangements with icosahedral symmetry for $n \leq 55000$, have been constructed [HSS]. The ad hoc methods of the earlier constructions have recently been replaced by different computer algorithms, but none of them has been shown to give the optimum.

Observe that $a_5 = a_6$ and $a_{11} = a_{12}$. Also, $A_2 = A_3$. It is conjectured that $a_n > a_{n+1}$ and $A_n > A_{n+1}$ in all other cases.

HYPERBOLIC SPACE

The density of a general arrangement of sets in d -dimensional hyperbolic space \mathbb{H}^d cannot be defined by a limit as in \mathbb{E}^d (see [FK93c]). The main difficulty is that in hyperbolic geometry the volume and the surface area of a ball of radius r are of the same order of magnitude as $r \rightarrow \infty$. In the absence of a reasonable definition of density with respect to the whole space, two natural problems arise:

- (i) Estimate the density of an arrangement relative to a bounded domain;
- (ii) Find substitutes for the notions of densest packing and thinnest covering.

Concerning the first problem, we mention the following result of K. Bezdek (see [FK93c]). Consider a packing of finitely many, but at least two, circles of radius

 TABLE 2.4.1 Densest packing and thinnest covering with congruent circles on a sphere.

n	a_n	AUTHOR	A_n	AUTHOR
2	180°	(elementary)	180°	(elementary)
3	120°	(elementary)	180°	(elementary)
4	$109.471\dots^\circ$	L. Fejes Tóth	$141.047\dots^\circ$	L. Fejes Tóth
5	90°	Schütte and van der Waerden	$126.869\dots^\circ$	Schütte
6	90°	L. Fejes Tóth	$109.471\dots^\circ$	L. Fejes Tóth
7	$77.866\dots^\circ$	Schütte and van der Waerden	$102.053\dots^\circ$	Schütte
8	$74.869\dots^\circ$	Schütte and van der Waerden		
9	$70.528\dots^\circ$	Schütte and van der Waerden		
10	$66.316\dots^\circ$	Danzer, Hárs	$84.615\dots^\circ$	G. Fejes Tóth
11	$63.435\dots^\circ$	Böröczky, Danzer		
12	$63.435\dots^\circ$	L. Fejes Tóth	$74.754\dots^\circ$	L. Fejes Tóth
14			$69.875\dots^\circ$	G. Fejes Tóth
24	$43.667\dots^\circ$	Robinson		

r in the hyperbolic plane \mathbb{H}^2 . Then the density of the circles relative to the outer parallel domain of radius r of the convex hull of their centers is at most $\pi/\sqrt{12}$.

As a corollary it follows that if at least two congruent circles are packed in a circular domain in \mathbb{H}^2 , then the density of the packing relative to the domain is at most $\pi/\sqrt{12}$. We note that the density of such a finite packing relative to the convex hull of the circles can be arbitrarily close to 1 as $r \rightarrow \infty$. K. Böröczky, Jr. (see [Bör]) proved a dual counterpart to the above-mentioned theorem of K. Bezdek, a corollary of which is that if at least two congruent circles cover a circular domain in \mathbb{H}^2 , then the density of the covering relative to the domain is at most $2\pi/\sqrt{27}$.

Rogers's simplex bound (2.1.5) for ball packings in \mathbb{E}^d has been extended by Böröczky (see [FK93b]) to \mathbb{H}^d as follows. If balls of radius r are packed in \mathbb{H}^d then the density of each ball relative to its Dirichlet cell is less than or equal to the density of $d+1$ balls of radius r centered at the vertices of a regular simplex of side-length $2r$ relative to this simplex. Of course, we should not interpret this result as a global density bound. The impossibility of such an interpretation is shown by an ingenious example of Böröczky (see [FK93b]). He constructed a packing \mathcal{P} of congruent circles in \mathbb{H}^2 and two tilings, \mathcal{T}_1 and \mathcal{T}_2 , both consisting of congruent tiles, such that each tile of \mathcal{T}_1 , as well as each tile of \mathcal{T}_2 , contains exactly one circle from \mathcal{P} , but such that the tiles of \mathcal{T}_1 and \mathcal{T}_2 have different areas.

The first notion that has been suggested as a substitute for densest packing and thinnest covering is “solidity.” \mathcal{P} is a **solid packing** if no finite subset of \mathcal{P} can be rearranged so as to form, together with the rest of \mathcal{P} , a packing not congruent to \mathcal{P} . Analogously, \mathcal{C} is a **solid covering** if no finite subset of \mathcal{C} can be rearranged so as to form, together with the rest of \mathcal{C} , a covering not congruent to \mathcal{C} . Obviously, in \mathbb{E}^d a solid packing with congruent copies of a body K has density $\delta(K)$, and a solid covering with congruent copies of K has density $\vartheta(K)$. This justifies the use of solidity as a natural substitute for “densest packing” and “thinnest covering” in hyperbolic space.

The tiling with Schläfli symbol $\{p, 3\}$ (see Chapters 19 or 21 of this Handbook) has regular p -gonal faces such that at each vertex of the tiling three faces meet. There exists such a tiling for each $p \geq 2$: for $p \leq 5$ on the sphere, for $p \geq 7$ on the hyperbolic plane, while for $p = 6$ we have the well-known hexagonal tiling on

the Euclidean plane. The incircles of such a tiling form a solid packing and the circumcircles form a solid covering. In addition, several packings and coverings by incongruent circles, including the the incircles and the circumcircles of certain trihedral Archimedean tilings have been confirmed to be solid (see [FK93c] and [Flo00, Flo01, FH00] for recent results).

Other substitutes for the notion of densest packing and thinnest covering have been proposed in [FKK98] and [Kup00]. A packing \mathcal{P} with congruent copies of a body K is **completely saturated** if no finite subset of \mathcal{P} can be replaced by a greater number of congruent copies of K that, together with the rest of \mathcal{P} , form a packing. Analogously, a covering \mathcal{C} with congruent copies of K is **completely reduced** if no finite subset of \mathcal{C} can be replaced by a smaller number of congruent copies of K that, together with the rest of \mathcal{C} , form a covering. While there are convex bodies that do not admit a solid packing or solid covering, it has been conjectured that each body in \mathbb{E}^d or \mathbb{H}^d admits a completely saturated packing and a completely reduced covering. By a body we mean a compact connected set that is the closure of its interior. The conjecture has been established for convex bodies in \mathbb{E}^d [FKK98] and recently in full generality in [Bow03]. However, the following rather counterintuitive result of Bowen makes it doubtful whether complete saturatedness and complete reducedness are good substitutes for the notions of densest packing and thinnest covering in hyperbolic space. For any positive number ε there is a body K in \mathbb{H}^d that admits a tiling and at the same time a completely saturated packing \mathcal{P} with the following property. For every point p in \mathbb{H}^d , the limit

$$\lim_{\lambda \rightarrow \infty} \frac{1}{V(B_\lambda(p))} \sum_{P \in \mathcal{P}} V(P \cap (B_\lambda(p)))$$

exists, is independent of p , and is less than ε . Here $V(\cdot)$ denotes the volume in \mathbb{H}^d and $B_\lambda(p)$ denotes the ball of radius λ centered at p .

In [BR03] and [BR04] Bowen and Radin proposed a probabilistic approach to analyze the efficiency of packings in hyperbolic geometry. Their approach can be sketched as follows.

Instead of studying individual arrangements, one considers the space Σ_K consisting of all saturated packings of \mathbb{H}^d by congruent copies of K . A suitable metric on Σ_K is introduced that makes Σ_K compact and makes the natural action of the group \mathcal{G}^d of rigid motions of \mathbb{H}^d on Σ_K continuous. We consider Borel probability measures on Σ_K that are invariant under \mathcal{G}^d . For such an invariant measure μ the **density** $d(\mu)$ of μ is defined as $d(\mu) = \mu(A)$, where A is the set of packings $\mathcal{P} \in \Sigma_K$ for which the origin of \mathbb{H}^d is contained in some member of \mathcal{P} . It follows easily from the invariance of μ that this definition is independent of the choice of the origin. The connection of density of measures to density of packings is established by the following theorem.

If μ is an ergodic invariant Borel probability measure on Σ_K , then—with the exception of a set of μ -measure zero—for every packing $\mathcal{P} \in \Sigma_K$, and for all $p \in \mathbb{H}^d$,

$$\lim_{\lambda \rightarrow \infty} \frac{1}{V(B_\lambda(p))} \sum_{P \in \mathcal{P}} V(P \cap (B_\lambda(p))) = d(\mu). \quad (2.4.1)$$

(A measure μ is ergodic if it cannot be expressed as the positive linear combination of two invariant measures.)

The **packing density** $\delta(K)$ of K can now be defined as the supremum of $d(\mu)$ for all ergodic invariant measures on Σ_K . A packing $\mathcal{P} \in \Sigma_K$ is **optimally dense**

if there is an ergodic invariant measure μ such that the orbit of \mathcal{P} under \mathcal{G}^d is dense in the support of μ and, for all $p \in \mathbb{H}^d$, (2.4.1) holds.

It is shown in [BR03] and [BR04] that there exists an ergodic invariant measure μ with $d(\mu) = \delta(K)$ and a subset of the support of μ of full μ -measure of optimally dense packings. Bowen and Radin prove several results justifying that this is a workable notion of optimal density and optimally dense packings. In particular, the definitions carry over without any change to \mathbb{E}^d , and there they coincide with the usual notions. The advantage of this probabilistic approach is that it neglects pathological packings such as the example by Böröczky. As for packings of balls, it is shown in [BR03] that there are only countably many radii for which there exists an optimally dense packing of balls of the given radius that is periodic.

2.5 NEIGHBORS

GLOSSARY

Neighbors: Two members of a packing whose closures intersect.

Newton number $N(K)$ of a convex body K : The maximum number of neighbors of K in all packings with congruent copies of K .

Hadwiger number $H(K)$ of a convex body K : The maximum number of neighbors of K in all packings with translates of K .

n -neighbor packing: A packing in which each member has exactly n neighbors.

n^+ -neighbor packing: A packing in which each member has at least n neighbors.

Table 2.5.1 contains the results known about Newton numbers and Hadwiger numbers (see [CS93, FK93c, Tal98a, Tal99a, Tal99b, Tal00]).

It seems that the maximum number of neighbors of one body in a lattice packing with congruent copies of K is considerably smaller than $H(K)$. While $H(B^d)$ is of exponential order of magnitude, the highest known number of neighbors in a lattice packing with B^d occurs in the Barnes-Wall lattice and is $c^{O(\log d)}$ [CS93]. Moreover, Gruber showed that, in the sense of Baire categories, most convex bodies in \mathbb{E}^d have no more than $2d^2$ neighbors in their densest lattice packing. Talata [Tal98b] gave examples of convex bodies in \mathbb{E}^d for which the difference between the Hadwiger number and the maximum number of neighbors in a lattice packing is 2^{d-1} . Alon [Alo97] constructed a finite ball packing in \mathbb{E}^d in which each ball has $c^{O(\sqrt{d})}$ neighbors.

A problem related to the determination of the Hadwiger number concerns the maximum number $C(K)$ of mutually nonoverlapping translates of a set K that have a common point. No more than four nonoverlapping translates of a topological disk in the plane can share a point [BKK95], while for $d \geq 3$ there are starlike bodies in \mathbb{E}^d for which $C(K)$ is arbitrarily large.

For a given convex body K , let $M(K)$ denote the maximum natural number with the property that an $M(K)$ -neighbor packing with finitely many congruent copies of K exists. For $n \leq M(K)$, let $L(n, K)$ denote the minimum cardinality, and, for $n > M(K)$, let $\lambda(n, K)$ denote the minimum density, of an n -neighbor packing with congruent copies of K . The quantities $M_T(K)$, $M^+(K)$, $M_T^+(K)$,

 TABLE 2.5.1 Newton and Hadwiger numbers.

BODY K	RESULT	AUTHOR
B^3	$N(K) = 12$	Schütte and van der Waerden
B^4	$N(K) = 24$	Musin
B^8	$N(K) = 240$	Levenštein; Odlyzko and Sloane
B^{24}	$N(K) = 196560$	Levenštein; Odlyzko and Sloane
Regular triangle	$N(K) = 12$	Böröczky
Square	$N(K) = 8$	Böröczky
Regular pentagon	$N(K) = 6$	Linhart
Regular n -gon for $n \geq 6$	$N(K) = 6$	Böröczky
Isosceles triangle with base angle $\pi/6$	$N(K) = 21$	Wegner
Convex disk of diameter d and width w	$N(K) \leq (4 + 2\pi)d/w + w/d + 2$	L. Fejes Tóth
Parallelotope in \mathbb{E}^d	$H(K) = 3^d - 1$	Hadwiger
Tetrahedron	$H(K) = 18$	Talata
Octahedron	$H(K) = 18$	Talata
Convex body in \mathbb{E}^d	$H(K) \leq 3^d - 1$	Hadwiger
Convex body in \mathbb{E}^d	$H(K) \geq 2^{cd}, c > 0$	Talata
Simplex in \mathbb{E}^d	$H(K) \geq 1.13488^{d-o(d)}$	Talata
Compact set in \mathbb{E}^d with $\text{int}(K - K) \neq \emptyset$	$H(K) \geq d^2 + d$	Smith

$L_T(n, K)$, $L^+(n, K)$, $L_T^+(n, K)$, $\lambda_T(n, K)$, $\lambda^+(n, K)$, and $\lambda_T^+(n, K)$ are defined analogously.

Österreicher and Linhart showed (see [FK93b]) that for a smooth convex disk K we have $L(2, K) \geq 3$, $L(3, K) \geq 6$, $L(4, K) \geq 8$, and $L(5, K) \geq 16$. All of these inequalities are sharp. We have $M_T^+(K) = 3$ for all convex disks, and there exists a 4-neighbor packing of density 0 with translates of any convex disk. There exists a 5-neighbor packing of density 0 with translates of a parallelogram, but Makai proved (see [FK93b]) that $\lambda_T^+(5, K) \geq 3/7$ and $\lambda_T^+(6, K) \geq 1/2$ for every $K \in \mathcal{K}(\mathbb{E}^2)$ that is not a parallelogram, and that $\lambda_T^+(5, K) \geq 9/14$ and $\lambda_T^+(6, K) \geq 3/4$ for every $K \in \mathcal{K}^*(\mathbb{E}^2)$ that is not a parallelogram. The case of equality characterizes triangles and affinely regular hexagons, respectively. According to a result of Chvátal (see [FK93c]), $\lambda_T^+(6, P) = 11/15$ for a parallelogram P .

A construction of Wegner (see [FK93c]) shows that $M(B^3) \geq 6$ and $L(6, B^3) \leq 240$, while Kertész [Ker94] proved that $M(B^3) \leq 8$. It is an open problem whether an n -neighbor or n^+ -neighbor packing of finitely many congruent balls exists for $n = 7$ and $n = 8$.

For 6^+ -neighbor packings with (not necessarily equal) circles, the following nice theorem of Bárány, Füredi, and Pach (see [FK93b]) holds:

In a 6^+ -neighbor packing with circles, either all circles are congruent or arbitrarily small circles occur.

2.6 SELECTED PROBLEMS ON LATTICE ARRANGEMENTS

In this section we discuss, from the vast literature on lattices, some special problems concerning arrangements of convex bodies in which the restriction to lattice arrangements is automatically imposed by the nature of the problem.

GLOSSARY

Point-trapping arrangement: An arrangement \mathcal{A} such that every component of the complement of the union of the members of \mathcal{A} is bounded.

Connected arrangement: An arrangement \mathcal{A} such that the union of the members of \mathcal{A} is connected.

j -impassable arrangement: An arrangement \mathcal{A} such that every j -dimensional flat intersects the interior of a member of \mathcal{A} .

Obviously, a point-trapping arrangement of congruent copies of a body can be arbitrarily thin. On the other hand, Bárány, Böröczky, Makai, and Pach showed that the density of a point-trapping lattice arrangement of any convex body in \mathbb{E}^d is greater than or equal to $1/2$. For $d \geq 3$, equality is attained only in the “checkerboard” arrangement of parallelotopes (see [FK93c]).

Bleicher (see [FK93c]) showed that the minimum density of a point-trapping lattice of unit balls in \mathbb{E}^3 is equal to

$$32\sqrt{(7142 + 1802\sqrt{17})^{-1}} = 0.265\dots$$

The extreme lattice is generated by three vectors of length $\frac{1}{2}\sqrt{7 + \sqrt{17}}$, any two of which make an angle of $\arccos\frac{\sqrt{17}-1}{8} = 67.021\dots^\circ$

For a convex body K , let $c(K)$ denote the minimum density of a connected lattice arrangement of congruent copies of K . According to a theorem of Groemer (see [FK93c]),

$$\frac{1}{d!} \leq c(K) \leq \frac{\pi^{d/2}}{2^d \Gamma(1+d/2)} \quad \text{for } K \in \mathcal{K}^d.$$

The lower bound is attained when K is a simplex or cross-polytope, and the upper bound is attained for a ball.

For a given convex body K in \mathbb{E}^d , let $\varrho_j(K)$ denote the infimum of the densities of all j -impassable lattice arrangements of copies of K . Obviously, $\varrho_0(K) = \vartheta_L(K)$. Let $\hat{K} = (K - K)^*$ denote the polar body of the difference body of K . Between $\varrho_{d-1}(K)$ and $\delta_L(\hat{K})$ Makai (see [FK93c]) found the following surprising connection:

$$\varrho_{d-1}(K)\delta_L(\hat{K}) = 2^d V(K)V(\hat{K}).$$

Little is known about $\varrho_j(K)$ for $0 < j < d - 1$. The value of $\varrho_1(B^3)$ has been determined recently [BW94]. We have

$$\varrho_1(B^3) = 9\pi/32 = 0.8835\dots$$

An extreme lattice is generated by the vectors $\frac{4}{3}(1, 1, 0)$, $\frac{4}{3}(0, 1, 1)$, and $\frac{4}{3}(1, 0, 1)$.

2.7 PACKING AND COVERING WITH SEQUENCES OF CONVEX BODIES

In this section we consider the following problem: Given a convex set K and a sequence $\{C_i\}$ of convex bodies in \mathbb{E}^d , is it possible to find rigid motions σ_i such

that $\{\sigma_i C_i\}$ covers K , or forms a packing in K ? If there are such motions σ_i , then we say that the sequence $\{C_i\}$ permits an *isometric covering* of K , or an *isometric packing* in K , respectively. If there are not only rigid motions but even translations τ_i so that $\{\tau_i C_i\}$ is a covering of K , or a packing in K , then we say that $\{C_i\}$ permits a *translative covering* of K , or a *translative packing* in K , respectively.

First we consider translative packings and coverings of cubes by sequences of boxes. By a *box* we mean an orthogonal parallelopiped whose sides are parallel to the coordinate axes. We let $I^d(s)$ denote a cube of side s in \mathbb{E}^d .

Groemer (see [Gro85]) proved that a sequence $\{C_i\}$ of boxes whose edge lengths are at most 1 permits a translative covering of $I^d(s)$ if

$$\sum V(C_i) \geq (s+1)^d - 1,$$

and that it permits a translative packing in $I^d(s)$ if

$$\sum V(C_i) \leq (s-1)^d - \frac{s-1}{s-2}((s-1)^{d-2} - 1).$$

Slightly stronger conditions (see [Las97]) guarantee even the existence of on-line algorithms for the determination of the translations τ_i . This means that the determination of τ_i is based only on C_i and the previously fixed sets $\tau_i C_i$.

We recall (see [Las97]) that to any convex body K in \mathbb{E}^d there exist two boxes, say Q_1 and Q_2 , with $V(Q_1) \geq 2d^{-d}V(K)$ and $V(Q_2) \leq d!V(K)$, such that $Q_1 \subset K \subset Q_2$. It follows immediately that if $\{C_i\}$ is a sequence of convex bodies in \mathbb{E}^d whose diameters are at most 1 and

$$\sum V(C_i) \geq \frac{1}{2}d^d((s+1)^d - 1),$$

then $\{C_i\}$ permits an isometric covering of $I^d(s)$; and that if

$$\sum V(C_i) \leq \frac{1}{d!} \left((s-1)^d - \frac{s-1}{s-2}((s-1)^{d-2} - 1) \right),$$

then it permits an isometric packing in $I^d(s)$.

The sequence $\{C_i\}$ of convex bodies is **bounded** if the set of the diameters of the bodies is bounded. As further consequences of the results above we mention the following. If $\{C_i\}$ is a bounded sequence of convex bodies such that $\sum V(C_i) = \infty$, then it permits an isometric covering of \mathbb{E}^d with density $\frac{1}{2}d^d$ and an isometric packing in \mathbb{E}^d with density $\frac{1}{d!}$. Moreover, if all the sets C_i are boxes, then $\{C_i\}$ permits a translative covering of \mathbb{E}^d and a translative packing in \mathbb{E}^d with density 1.

In \mathbb{E}^2 , any bounded sequence $\{C_i\}$ of convex disks with $\sum a(C_i) = \infty$ permits even a translative packing and covering with density $\frac{1}{2}$ and 2, respectively. It is an open problem whether for $d > 2$ any bounded sequence $\{C_i\}$ of convex bodies in \mathbb{E}^d with $\sum V(C_i) = \infty$ permits a translative covering. If the sequence $\{C_i\}$ is unbounded, then the condition $\sum V(C_i) = \infty$ no longer suffices for $\{C_i\}$ to permit even an isometric covering of the space. For example, if C_i is the rectangle of side lengths i and $\frac{1}{i^2}$, then $\sum a(C_i) = \infty$ but $\{C_i\}$ does not permit an isometric covering of \mathbb{E}^2 . There is a simple reason for this, which brings us to one of the most interesting topics of this subject, namely Tarski's plank problem.

A **plank** is a region between two parallel hyperplanes. Tarski conjectured that if a convex body of minimum width w is covered by a collection of planks in \mathbb{E}^d , then the sum of the widths of the planks is at least w . Tarski's conjecture was first proved by Bang. Bang's theorem immediately implies that the sequence of rectangles above does not permit an isometric covering of \mathbb{E}^2 , not even of $(\frac{\pi^2}{12} + \epsilon)B^2$.

There is a nice account of the history of Tarski's plank problem and its generalizations in [Gro85]. In his paper, Bang asked whether his theorem can be generalized so that the width of each plank is measured relative to the width of the convex body being covered, in the direction normal to the plank. Bang's problem has been solved for centrally symmetric bodies by Ball [Bal91]. This case has a particularly appealing formulation in terms of normed spaces:

If the unit ball in a Banach space is covered by a countable collection of planks, then the total width of the planks is at least 2.

2.8 SOURCES AND RELATED MATERIAL

SURVEYS

The monographs [Fej72, Rog64, Zon99] are devoted solely to packing and covering; also the books [CS93, CFG91, EGH89, Fej64, GL87, PA95, Zon96] contain results relevant to this chapter. Additional material and bibliography can be found in the following surveys: [Bar69, Fej83, Fej84, Fej99, FK93b, FK93c, FK01, Few67, Flo87, Flo02, GW93, Gro85, Gru79, MP93, SA75].

RELATED CHAPTERS

- [Chapter 3: Tilings](#)
- [Chapter 7: Lattice points and lattice polytopes](#)
- [Chapter 13: Geometric discrepancy theory and uniform distribution](#)
- [Chapter 19: Symmetry of polytopes and polyhedra](#)
- [Chapter 21: Polyhedral maps](#)
- [Chapter 61: Sphere packing and coding theory](#)
- [Chapter 62: Crystals and quasicrystals](#)

REFERENCES

- [Alo97] N. Alon. Packings with large minimum kissing numbers. *Discrete Math.*, 175:249–251, 1997.
- [Bal91] K. Ball. The plank problem for symmetric bodies. *Invent. Math.*, 104:535–543, 1991.
- [Bar69] E.P. Baranovskii. Packings, coverings, partitionings and certain other arrangements in spaces with constant curvature (Russian). *Itogi Nauki—Ser. Mat. (Algebra, Topologiya, Geometriya)*, 14:189–225, 1969. Translated in *Progr. Math.*, 9:209–253, 1971.

- [Bez93] K. Bezdek. Hadwiger-Levi's covering problem revisited. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 199–233. Springer-Verlag, New York, 1993.
- [Bez94] A. Bezdek. A remark on the packing density in the 3-space. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 17–22. North-Holland, Amsterdam, 1994.
- [Bez02] K. Bezdek. Improving Rogers' upper bound for the density of unit ball packings via estimating the surface area of Voronoi cells from below in Euclidean d -space for all $d \geq 8$. *Discrete Comput. Geom.*, 28:75–106, 2002.
- [BH98] U. Betke and M. Henk. Finite packings of spheres. *Discrete Comput. Geom.*, 19:197–227, 1998.
- [BH00] U. Betke and M. Henk. Densest lattice packings of 3-polytopes. *Comput. Geom. Theory Appl.*, 16:157–186, 2000.
- [BHW94] U. Betke, M. Henk, and J.M. Wills. Finite and infinite packings. *J. Reine Angew. Math.*, 453:165–191, 1994.
- [BKK95] A. Bezdek, K. Kuperberg, and W. Kuperberg. Mutually contiguous and concurrent translates of a plane disk. *Duke Math. J.*, 78:19–31, 1995.
- [BKM91] A. Bezdek, W. Kuperberg, and E. Makai, Jr. Maximum density space packings with parallel strings of balls. *Discrete Comput. Geom.*, 6:277–283, 1991.
- [Bör] K. Böröczky, Jr. *Finite Packing and Covering*. Cambridge University Press, to appear.
- [Bow03] L. Bowen. On the existence of completely saturated packings and completely reduced coverings. *Geom. Dedicata*, 98:211–226, 2003.
- [BR03] L. Bowen and C. Radin. Densest packing of equal spheres in hyperbolic space. *Discrete Comput. Geom.*, 29:23–39, 2003.
- [BR04] L. Bowen and C. Radin. Optimally dense packings of hyperbolic space. *Geom. Dedicata*, to appear.
- [BW94] R.P. Bambah and A.C. Woods. On a problem of G. Fejes Tóth. *Proc. Indian Acad. Sci. Math. Sci.*, 104:137–156, 1994.
- [CE03] H. Cohn and N. Elkies. New upper bounds on sphere packings I. *Ann. of Math.*, 157:689–714, 2003.
- [CFG91] H.T. Croft, K.J. Falconer, and R.K. Guy. *Unsolved Problems in Geometry*. Springer-Verlag, New York, 1991.
- [Coh02] H. Cohn. New upper bounds on sphere packings II. *Geom. Topol.*, 6:329–353, 2002.
- [CS93] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*, 2nd edition. Springer-Verlag, New York, 1993.
- [DLR92] P.G. Doyle, J.C. Lagarias, and D. Randall. Self-packing of centrally symmetric convex discs in R^2 . *Discrete Comput. Geom.*, 8:171–189, 1992.
- [EGH89] P. Erdős, P.M. Gruber, and J. Hammer. *Lattice Points*. Number 39 of *Pitman Monographs*. Longman Scientific/Wiley, New York, 1989.
- [Fej64] L. Fejes Tóth. *Regular Figures*. Pergamon, Oxford, 1964.
- [Fej72] L. Fejes Tóth. *Lagerungen in der Ebene auf der Kugel und im Raum*, 2nd edition. Springer-Verlag, Berlin, 1972.
- [Fej83] G. Fejes Tóth. New results in the theory of packing and covering. In P.M. Gruber and J.M. Wills, editors, *Convexity and Its Applications*, pages 318–359. Birkhäuser, Basel, 1983.

- [Fej84] L. Fejes Tóth. Density bounds for packing and covering with convex discs. *Exposition. Math.*, 2:131–153, 1984.
- [Fej95] G. Fejes Tóth. Densest packings of typical convex sets are not lattice-like. *Discrete Comput. Geom.*, 14:1–8, 1995.
- [Fej99] G. Fejes Tóth. Recent Progress on packing and covering. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemp. Math.*, pages 145–162. Amer. Math. Soc., Providence, 1999.
- [Fer] S.P. Ferguson. Sphere packings V. ArXiv math.MG/9811077.
- [Few67] L. Few. Multiple packing of spheres: a survey. In *Proc. Colloquium Convexity (Copenhagen 1965)*, pages 88–93. Københavns Univ. Mat. Inst., 1967.
- [FH00] A. Florian and A. Heppes. Solid coverings of the Euclidean plane with incongruent circles. *Discrete Comput. Geom.*, 23:225–245, 2000.
- [FH] S.P. Ferguson and T.C. Hales. A formulation of the Kepler conjecture. ArXiv math.MG/99811072.
- [FK93a] G. Fejes Tóth and W. Kuperberg. Blichfeldt’s density bound revisited. *Math. Ann.*, 295:721–727, 1993.
- [FK93b] G. Fejes Tóth and W. Kuperberg. Packing and covering with convex sets. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 799–860. North-Holland, Amsterdam, 1993.
- [FK93c] G. Fejes Tóth and W. Kuperberg. Recent results in the theory of packing and covering. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 251–279. Springer-Verlag, New York, 1993.
- [FK95] G. Fejes Tóth and W. Kuperberg. Thin non-lattice covering with an affine image of a strictly convex body. *Mathematika*, 42:239–250, 1995.
- [FK01] G. Fejes Tóth and W. Kuperberg. Sphere packing. In Robert A. Myers, editor, *Encyclopedia of Physical Sciences and Technology*, 3rd edition, Volume 15, pages 657–665. Academic Press, New York, 2001.
- [FKK98] G. Fejes Tóth, G. Kuperberg, and W. Kuperberg. Highly saturated packings and reduced coverings. *Monatsh. Math.*, 125:127–145, 1998.
- [Flo87] A. Florian. Packing and covering with convex discs. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry (Siófok, 1985)*, volume 48 of *Colloq. Math. Soc. János Bolyai*, pages 191–207. North-Holland, Amsterdam, 1987.
- [Flo00] A. Florian. An infinite set of solid packings on the sphere. *Österreich. Akad. Wiss. Math.-Natur. Kl. Sitzungsber. II*, 209:67–79, 2000.
- [Flo01] A. Florian. Packing of incongruent circles on a sphere. *Monatsh. Math.*, 133:111–129, 2001.
- [Flo02] A. Florian. Some recent results in discrete geometry. *Rend. Circ. Mat. Palermo (2) Suppl.*, 70, part 1:297–309, 2002.
- [Fod99] F. Fodor. The densest packing of 19 congruent circles in a circle. *Geom. Dedicata*, 74:139–145, 1999.
- [Fod00] F. Fodor. The densest packing of 12 congruent circles in a circle. *Beiträge Algebra Geom.*, 41:401–409, 2000.
- [Fod] F. Fodor. The densest packing of 13 congruent circles in a circle. *Beiträge Algebra Geom.*, to appear.

- [FZ94] G. Fejes Tóth and T. Zamfirescu. For most convex discs thinnest covering is not lattice-like. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 105–108. North-Holland, Amsterdam/New York, 1994.
- [GL87] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. Elsevier, North-Holland, Amsterdam, 1987.
- [Gro85] H. Groemer. Coverings and packings by sequences of convex sets. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440 of *Ann. New York Acad. Sci.*, pages 262–278. 1985.
- [Gru79] P.M. Gruber. Geometry of numbers. In J. Tölke and J.M. Wills, editors, *Contributions to Geometry*, Proc. Geom. Symp. (Siegen, 1978), pages 186–225. Birkhäuser, Basel, 1979.
- [GW93] P. Gritzmann and J.M. Wills. Finite packing and covering. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 861–897. North-Holland, Amsterdam, 1993.
- [Hal92] T.C. Hales. The sphere packing problem. *J. Comput. Appl. Math.*, 44:41–76, 1992.
- [Hal93] T.C. Hales. Remarks on the density of sphere packings in three dimensions. *Combinatorica*, 13:181–187, 1993.
- [Hal97] T.C. Hales. Sphere packings I. *Discrete Comput. Geom.*, 17:1–51, 1997.
- [Hal98] T.C. Hales. Sphere packings II. *Discrete Comput. Geom.*, 18:135–149, 1998.
- [Hal00] T.C. Hales. Cannonballs and honeycombs. *Notices Amer. Math. Soc.*, 47:440–449, 2000.
- [Hal03] T.C. Hales. Some algorithms arising in the proof of the Kepler conjecture. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 489–507. Springer-Verlag, New York, 2003.
- [Hala] T.C. Hales. An overview of the Kepler conjecture. ArXiv math.MG/9811071.
- [Halb] T.C. Hales. Sphere packings III. ArXiv math.MG/9811075.
- [Halc] T.C. Hales. Sphere packings IV. Preprint, ArXiv math.MG/9811076.
- [Hald] T.C. Hales. The Kepler conjecture. ArXiv math.MG/9811078.
- [HM97] A. Heppes and J.B.M. Melissen. Covering a rectangle with equal circles. *Period. Math. Hungar.*, 34:63–79, 1997.
- [Hsi93] W.-Y. Hsiang. On the sphere packing problem and the proof of Kepler’s conjecture. *Internat. J. Math.*, 93:739–831, 1993.
- [Hsi01] W.-Y. Hsiang. *Least Action Principle of Crystal Formation of Dense Packing Type and Kepler’s Conjecture*, Volume 3 of Nankai Tracts in Mathematics. World Scientific, Singapore, 2001.
- [HSS] R.H. Hardin, N.J.A. Sloane, and W.D. Smith. *Spherical Codes*. In preparation.
- [Ism98] D. Ismailescu. Covering the plane with copies of a convex disc. *Discrete Comput. Geom.*, 20:251–263, 1998.
- [Ker94] G. Kertész. Nine points on the hemisphere. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 189–196. North-Holland, Amsterdam, 1994.
- [KK93] J. Kahn and G. Kalai. A counterexample to Borsuk’s conjecture. *Bull. Amer. Math. Soc.*, 29:60–62, 1993.
- [Kro93] S. Krotoszyński. Covering a disc with smaller discs. *Studia Sci. Math. Hungar.*, 28:271–283, 1993.

- [Kup00] G. Kuperberg. Notions of denseness. *Geom. Topol.*, 4:277–292, 2000.
- [Lag02] J.C. Lagarias. Bounds for local density of sphere packings and the Kepler conjecture. *Discrete Comput. Geom.*, 27:165–193, 2002.
- [Las97] M. Lassak. A survey of algorithms for on-line packing and covering by sequences of convex bodies. In I. Bárány and K. Böröczky, editors, *Intuitive Geometry*, volume 6 of *Bolyai Soc. Math. Studies*, pages 129–157. János Bolyai Math. Soc., Budapest, 1997.
- [Mel93] J.B.M. Melissen. Densest packings of congruent circles in an equilateral triangle. *Amer. Math. Monthly*, 100:816–825, 1993.
- [Mel94] J.B.M. Melissen. Densest packings of eleven congruent circles in a circle. *Geom. Dedicata*, 50:15–25, 1994.
- [Mel97] J.B.M. Melissen. Loosest circle coverings of an equilateral triangle. *Math. Mag.*, 70:119–125, 1997.
- [MP93] W. Moser and J. Pach. Research problems in discrete geometry. Report 93-32, DIMACS, Rutgers, New Brunswick, 1993.
- [Mud93] D.J. Muder. A new bound on the local density of sphere packings. *Discrete Comput. Geom.*, 10:351–375, 1993.
- [PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. Wiley, New York, 1995.
- [Pei94] R. Peikert. Dichteste Packung von gleichen Kreisen in einem Quadrat. *Elem. Math.*, 49:16–26, 1994.
- [Rog64] C.A. Rogers. *Packing and Covering*. Cambridge University Press, Cambridge, 1964.
- [SA75] T.L. Saaty and J.M. Alexander. Optimization and the geometry of numbers: packing and covering. *SIAM Rev.*, 17:475–519, 1975.
- [Sch88] P. Schmitt. An aperiodic prototile in space. 1988. Preprint.
- [Sch91] P. Schmitt. Disks with special properties of densest packings. *Discrete Comput. Geom.*, 6:181–190, 1991.
- [Sch94] J. Schaer. The densest packing of ten congruent spheres in a cube. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 403–424. North-Holland, Amsterdam, 1994.
- [Tal98a] I. Talata. Exponential lower bound for the translative kissing numbers of d -dimensional convex bodies. *Discrete Comput. Geom.*, 19:447–455, 1998.
- [Tal98b] I. Talata. On a lemma of Minkowski. *Period. Math. Hungar.*, 32:199–207, 1998.
- [Tal99a] I. Talata. The translative kissing number of tetrahedra is 18. *Discrete Comput. Geom.*, 22:231–248, 1999.
- [Tal99b] I. Talata. On extensive subsets of convex bodies. *Period. Math. Hungar.*, 38:231–246, 1999.
- [Tal00] I. Talata. A lower bound for the translative kissing numbers of simplices. *Combinatorica*, 20:281–293, 2000.
- [Tem94a] Á. Temesvári. Die dichteste gitterförmige 9-fache Kreispackung. *Rad. Hrvatske Akad. Znan. Umj. Mat.*, 11:95–110, 1994.
- [Tem94b] Á. Temesvári. Die dünnste 8-fache gitterförmige Kreisüberdeckung der Ebene. *Studia Sci. Math. Hungar.*, 29:323–340, 1994.
- [Zon96] C. Zong. *Strange Phenomena in Convex and Discrete Geometry*. Springer-Verlag, New York, 1996.
- [Zon99] C. Zong. *Sphere Packings*. Springer-Verlag, New York, 1999.

3 TILINGS

Doris Schattschneider and Marjorie Senechal

INTRODUCTION

Tilings of surfaces and packings of space have been of interest to artisans and manufacturers throughout history; they are a means of artistic expression and lend economy and strength to modular constructions. Today scientists and mathematicians study tilings because they pose interesting mathematical questions and provide mathematical models for such diverse structures as the molecular anatomy of crystals, cell packings of viruses, n -dimensional algebraic codes, and “nearest neighbor” regions for a set of discrete points. The basic questions are: What bodies can tile space? In what ways do they tile? However, in this generality such questions are intractable. To study tiles and tilings, we must impose constraints.

Even with constraints the subject is unmanageably large. In this chapter we restrict ourselves, for the most part, to tilings of unbounded spaces. In the next section we present some general results that are fundamental to the subject as a whole. Section 3.2 addresses tilings with congruent tiles. In Section 3.3 we discuss the classical subject of periodic tilings, which continues to be enriched with new results. Next, we briefly describe the newer theory of nonperiodic and aperiodic tilings, both of which are discussed in more detail in [Chapter 62](#). We conclude with a very brief description of some kinds of tilings not considered here.

3.1 GENERAL CONSIDERATIONS

In this section we define terms that will be used throughout the chapter and state some basic results. Taken together, these results state that although there is no algorithm for deciding which bodies are tiles, there are criteria for deciding the question in certain cases. We can obtain some quantitative information about the tiling in particularly well-behaved cases.

Unless otherwise stated, we assume that S is an n -dimensional space, either Euclidean (\mathbb{E}^n) or hyperbolic. We also assume that the tiles are bounded and the tilings are locally finite (see the Glossary below). Throughout this chapter, n is the dimension of the space in which we are working.

GLOSSARY

Body: A bounded region (of S) that is the closure of its (nonempty) interior.

Tiling (of S): A decomposition of S into a countable number of n -dimensional bodies whose interiors are pairwise disjoint. In this context, the bodies are also called n -cells and are the tiles of the tiling (see below). Synonyms: *tessellation*, *parquetry* (when $n = 2$), *honeycomb* (for $n \geq 2$).

Tile: A body that is an n -cell of one or more tilings of S . To say that a body *tiles* a region $R \subseteq S$ means that R can be covered exactly by copies of the body without gaps or overlaps.

Locally finite tiling: Every n -ball of finite radius in S meets only finitely many tiles of the tiling.

Prototile set (for a tiling \mathcal{T} of S): A minimal subset of tiles in \mathcal{T} such that each tile in the tiling \mathcal{T} is the congruent image of one of those in the prototile set. The tiles in the set are called prototiles and the prototile set is said to admit \mathcal{T} .

k -face (of a tiling): An intersection of at least $n - k + 1$ tiles of the tiling that is not contained in a j -face for $j < k$. (The 0-faces are the *vertices* and 1-faces the *edges*; the $(n-1)$ -faces are simply called the *faces* of the tiling.)

Patch (in a tiling): A set of tiles whose union is homeomorphic to an n -ball. See Figure 3.1.1. A spherical patch $P(r, s)$ is the set of tiles whose intersection with the ball of radius r centered at s is nonempty, together with any additional tiles needed to complete the patch (that is, to make it homeomorphic to an n -ball).

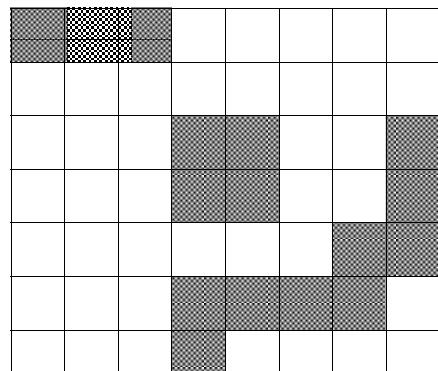


FIGURE 3.1.1
Three patches in a tiling of the plane by squares.

Normal tiling: A tiling in which (i) each prototile is homeomorphic to an n -ball, and (ii) the prototiles are uniformly bounded (there exist $r > 0$ and $R > 0$ such that each prototile contains a ball of radius r and is contained in a ball of radius R). It is technically convenient to include a third condition: (iii) the intersection of every pair of tiles is a connected set. (A normal tiling is necessarily locally finite.)

Face-to-face tiling (by polytopes): A tiling in which the faces of the tiling are also the $(n-1)$ -dimensional faces of the polytopes. (A face-to-face tiling by convex polytopes is also k -face-to- k -face for $0 \leq k \leq n-1$.) In dimension 2, this is an *edge-to-edge* tiling by polygons, and in dimension 3, a face-to-face tiling by polyhedra.

Dual tiling: Two tilings \mathcal{T} and \mathcal{T}^* are dual if there is an incidence-reversing bijection between the k -faces of \mathcal{T} and the $(n-k)$ -faces of \mathcal{T}^* (see Figure 3.1.2).

Voronoi (Dirichlet) tiling: A tiling whose tiles are the Voronoi cells of a discrete set Λ of points in S . The Voronoi cell of a point $p \in \Lambda$ is the set of all points in S that are at least as close to p as to any other point in Λ (see Chapter 23).

Delaunay (or Delone) tiling: A face-to-face tiling by convex circumscribable polytopes (i.e., the vertices of each polytope lie on a sphere).

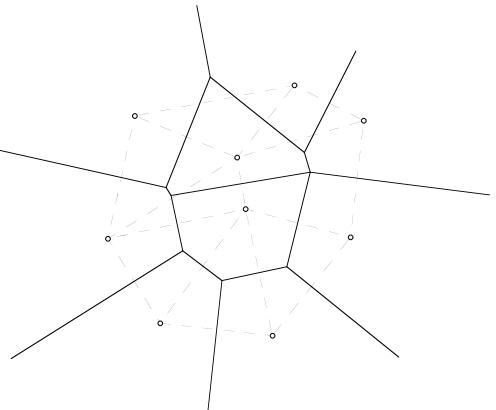


FIGURE 3.1.2

A Voronoi tiling (solid lines) and its Delaunay dual (dashed lines).

Isometry: A distance-preserving self-map of S .

Symmetry group (of a tiling): The set of isometries of S that map the tiling to itself.

MAIN RESULTS

1. **The Undecidability Theorem.** There is no algorithm for deciding whether or not an arbitrary body or set of bodies admits a tiling of S [Ber66].
2. **The Extension Theorem (for \mathbb{E}^n).** Let A be any finite set of bodies, each homeomorphic to a closed n -ball. If A tiles regions that contain arbitrarily large n -balls, then A admits a tiling of \mathbb{E}^n . (These regions need not be nested, nor need any of the tilings of the regions be extendable!) The proof for $n = 2$ in [GS87] extends to E^n with minor changes.
3. **The Normality Lemma (for \mathbb{E}^n).** In a normal tiling, the ratio of the number of tiles that meet the boundary of a spherical patch to the number of tiles in the patch tends to zero as the radius of the patch tends to infinity. In fact, a stronger statement can be made: For $s \in S$ let $t(r, s)$ be the number of tiles in the spherical patch $P(r, s)$. Then, in a normal tiling, for every $x > 0$,

$$\lim_{r \rightarrow \infty} \frac{t(r + x, s) - t(r, s)}{t(r, s)} = 0.$$

The proof for $n = 2$ in [GS87] extends to \mathbb{E}^n with minor changes.

4. **Euler's Theorem for tilings of \mathbb{E}^2 .** Let \mathcal{T} be a normal tiling of \mathbb{E}^2 , and let $t(r, s)$, $e(r, s)$, and $v(r, s)$ be the numbers of tiles, edges, and vertices, respectively, in the circular patch $P(r, s)$. Then if one of the limits $e(\mathcal{T}) = \lim_{r \rightarrow \infty} e(r, s)/t(r, s)$ or $v(\mathcal{T}) = \lim_{r \rightarrow \infty} v(r, s)/t(r, s)$ exists, so does the other, and $v(\mathcal{T}) - e(\mathcal{T}) + 1 = 0$. Like Euler's Theorem for Planar Maps, on which the proof of this theorem is based, this result can be extended in various ways [GS87].
5. **Voronoi Dual.** Every Voronoi tiling has a Delaunay dual and conversely (see Figure 3.1.2) [Vor09].

3.2 TILINGS BY ONE TILE

To say that a body tiles \mathbb{E}^n usually means that there is a tiling all of whose tiles are copies of this body. The artist M.C. Escher has demonstrated how intricate such tiles can be even when $n = 2$. But in higher dimensions the simplest tiles—for example, cubes—can produce surprises, as the recent counterexample to Keller’s conjecture attests (see below).

GLOSSARY

Monohedral tiling: A tiling with a single prototile.

r -morphic tile: A prototile that admits exactly r distinct monohedral tilings.

Figure 3.2.1 shows a 5-morphic tile and all its tilings, and Figure 3.2.3 shows a 1-morphic tile and its tiling.

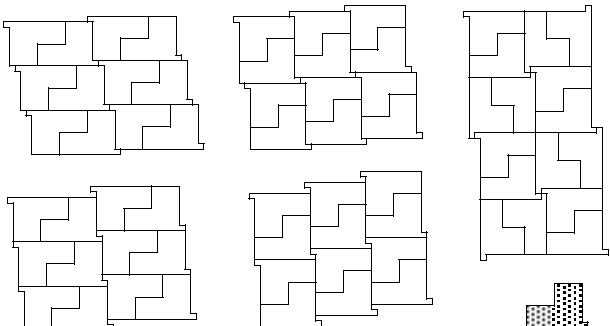


FIGURE 3.2.1
A pentamorphic tile.

k -rep tile: A body for which k copies can be assembled into a larger, similar body. (Or, equivalently, a body that can be partitioned into k congruent bodies, each similar to the original.) More formally, a k -rep tile is a closed set A_1 in S with nonempty interior such that there are sets A_2, \dots, A_k congruent to A_1 that satisfy

$$\text{Int } A_i \cap \text{Int } A_j = \emptyset$$

for all $i \neq j$ and $A_1 \cup \dots \cup A_k = g(A_1)$, where g is a similarity mapping. (Figure 3.2.2 shows a 3-dimensional chair rep tile and the second-level chair. An n -dimensional chair rep tile can be formed in a similar manner.)

Transitive action: A group G is said to act transitively on a set $\{A_1, A_2, \dots\}$ if the set is an orbit for G . (That is, for every pair A_i, A_j of elements of the set, there is a $g_{ij} \in G$ such that $g_{ij}A_i = A_j$.)

Regular system of points: A discrete set of points on which an infinite group of isometries acts transitively.

Isohedral (tiling): A tiling whose symmetry group acts transitively on its tiles.

Anisohedral tile: A prototile that admits monohedral tilings but no isohedral tilings. In Figure 3.2.3, the prototile admits a unique nonisohedral tiling; the shaded tiles are each surrounded differently, from which it follows that no isom-

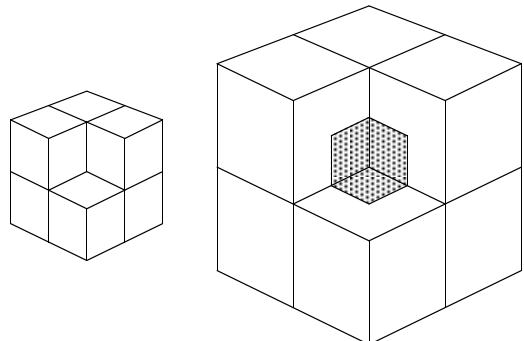


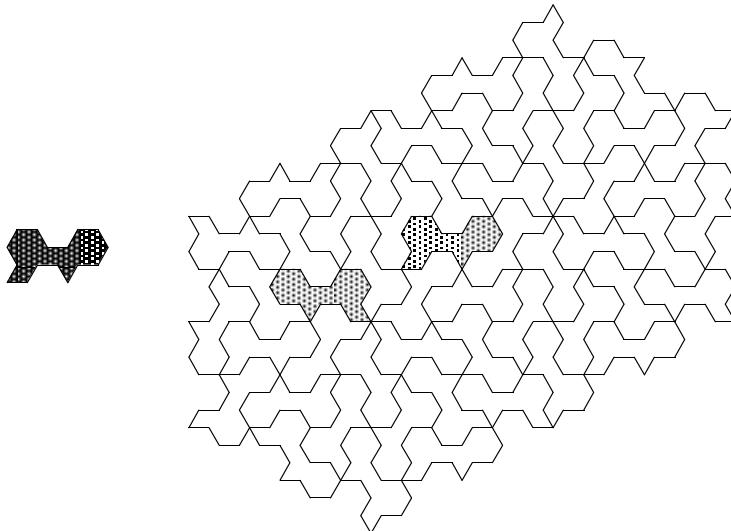
FIGURE 3.2.2

A 3-dimensional chair rep tile and a second-level chair in which seven copies surround the first.

etry can map one to the other (and the tiling to itself). This tiling is periodic, however (see [Section 3.3](#)).

FIGURE 3.2.3

An anisohedral tile (due to R. Penrose) and its unique tiling in which tiles are surrounded in two different ways.



Corona (of a tile P in a tiling \mathcal{T}): Define $C^0(P) = P$. Then $C^k(P)$, the k th corona of P , is the set of all tiles $Q \in T$ for which there exists a path of tiles $P = P_0, P_1, \dots, P_m = Q$ with $m \leq k$ in which $P_i \cap P_{i+1} \neq \emptyset$, $i = 0, 1, \dots, m - 1$.

Lattice: The group of integral linear combinations of n linearly independent vectors in S . A point orbit of a lattice, often called a **point lattice**, is a particular case of a regular system of points.

Translation tiling: A monohedral tiling of S in which every tile is a translate of a fixed prototile. See [Figure 3.2.4](#).

Lattice tiling: A monohedral tiling on whose tiles a lattice of translation vectors acts transitively. Figure 3.2.4 is not a lattice tiling since it is invariant by multiples of just one vector.

n -parallelotope: A convex n -polytope that tiles \mathbb{E}^n by translation.

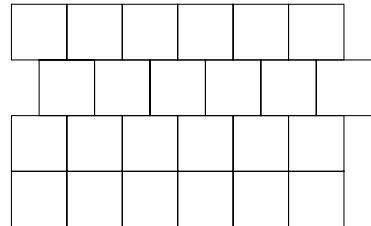


FIGURE 3.2.4
A translation non-lattice tiling.

Belt (of an n -parallelotope): A maximal subset of parallel $(n-2)$ -faces of a parallelotope in \mathbb{E}^n . The number of $(n-2)$ -faces in a belt is its length.

Center of symmetry (for a set A in \mathbb{E}^n): A point $a \in A$ such that A is invariant under the mapping $x \rightarrow 2a - x$; the mapping is called **central inversion** and an object that has a center of symmetry is said to be **centrosymmetric**.

Stereohedron: A convex polytope that is the prototile of an isohedral tiling. A Voronoi cell of a regular system of points is a stereohedron.

Linear expansive map: A linear transformation all of whose eigenvalues have modulus greater than one.

MAIN RESULTS

1. **The Local Theorem.** Let \mathcal{T} be a monohedral tiling of S , and for $P \in \mathcal{T}$, let $S_i(P)$ be the subgroup of the symmetry group of P that leaves invariant $C^i(P)$, the i th corona of P . \mathcal{T} is isohedral if and only if there exists an integer $k > 0$ for which the following two conditions hold: (a) for all $P \in \mathcal{T}$, $S_{k-1}(P) = S_k(P)$ and (b) For every pair of tiles P, P' in \mathcal{T} , there exists an isometry γ such that $\gamma(P) = P'$ and $\gamma(C^k(P)) = C^k(P')$. In particular, if P is asymmetric, then \mathcal{T} is isohedral if and only if condition (b) holds for $k = 1$ [DS98].
2. A convex polytope is a parallelotope if and only if it is centrosymmetric, its faces are centrosymmetric, and its belts have lengths four or six. First proved by Venkov, this theorem was rediscovered independently by McMullen [Ven54, McM80].
3. The number $|F|$ of faces of a convex parallelotope in \mathbb{E}^n satisfies Minkowski's inequality, $2n \leq |F| \leq 2(2^n - 1)$. Both upper and lower bounds are realized in every dimension [Min97].
4. The number of faces of an n -dimensional stereohedron in \mathbb{E}^n is bounded. In fact, if a is the number of translation classes of the stereohedron in an isohedral tiling, then the number of faces is at most the Delaunay bound $2^n(1 + a) - 2$ [Del61].
5. Using a classification system that takes into account the symmetry groups of the tilings and their tiles, the combinatorial structure of the tiling, and the ways in which the tiles are related to adjacent tiles, Grünbaum and Shephard proved that there are 81 classes of isohedral tilings of \mathbb{E}^2 , 93 classes if the tiles are **marked** (that is, they have decorative markings to express symmetry in

addition to the tile shape) [GS77]. There is an infinite number of classes of isohedral tilings of \mathbb{E}^n , $n > 2$.

6. Anisohedral tiles exist in \mathbb{E}^n for every $n \geq 2$ [GS80]. (The first example, given for $n = 3$ by Reinhardt [Rei28], was the solution to part of Hilbert's 18th problem.) H. Heesch gave the first example for $n = 2$ [Hee35] and R. Kershner the first convex examples [Ker68].
7. Every n -parallelopiped admits a lattice tiling. However, for $n \geq 3$, there are nonconvex tiles that tile by translation but do not admit lattice tilings [SS94].
8. A lattice tiling of \mathbb{E}^n by unit cubes must have a pair of cubes sharing a whole face [Min07, Haj42]. However, a famous conjecture of Keller, which stated that for every n , any tiling of \mathbb{E}^n by congruent cubes must contain at least one pair of cubes sharing a whole face, is false: for $n \geq 10$, there are translation tilings by unit cubes in which no two cubes share a whole face [LS92].
9. Every linear expansive map that transforms the lattice \mathbb{Z}^n of integer vectors into itself defines a family of k -rep tiles; these tiles, which usually have fractal boundaries, admit lattice tilings [Ban91].

OPEN PROBLEMS

1. Which convex n -polytopes in \mathbb{E}^n are prototiles for monohedral tilings of \mathbb{E}^n ? This is unsolved for all $n \geq 2$ (see [GS87] for the case $n = 2$; the list of convex pentagons that tile has not been proved complete). For higher dimensions, little is known; it is not even known which tetrahedra tile \mathbb{E}^3 [GS80, Sen81].
2. **Heesch's Problem.** Is there an integer k_n , depending only on the dimension n of the space S , such that if a body A can be completely surrounded k_n times by tiles congruent to A , then A is a prototile for a monohedral tiling of S ? (A is completely surrounded once if A , together with congruent copies that have nonempty intersection with A , tile a patch containing A in its interior.) When $S = \mathbb{E}^2$, $k_2 > 5$. The body shown in [Figure 3.2.5](#) can be completely surrounded three times but not four; William Rex Marshall and, independently, Casey Mann, found 4-corona tiles, and Mann 5-corona tiles [Man01]. This problem is unsolved for all n .
3. **Keller's conjecture** is true for $n \leq 6$ and false for $n \geq 10$ (see Result 8 above). The cases $n = 7, 8$, and 9 are still open.
4. Do r -morphic tiles exist for every positive integer r ? Fontaine and Martin have shown the answer is yes in \mathbb{E}^2 for $r \leq 10$ [FM84].
5. Find a good upper bound for the number of faces of an n -dimensional stereohedron. Delaunay's bound, stated above, is evidently much too high; for example, it gives 390 as the bound in \mathbb{E}^3 , while the maximal known number of faces of a three-dimensional stereohedron (found by P. Engel [Eng81]) is 38.
6. For monohedral (face-to-face) tilings by convex polytopes there is an integer k_n , depending only on the dimension n of S , that is an upper bound for the constant k in the Local Theorem [DS98]. Find the value of this k_n . For the

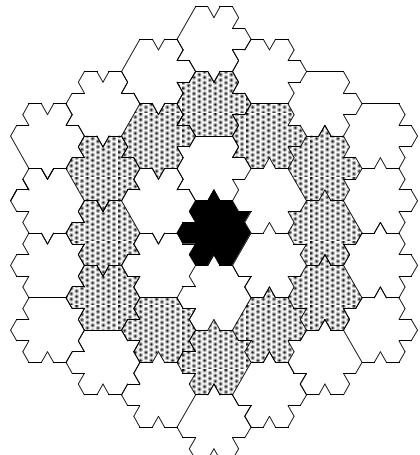


FIGURE 3.2.5
Ammann's 3-corona tile cannot be surrounded by a fourth corona. 4-corona and 5-corona tiles also exist.

Euclidean plane \mathbb{E}^2 it is known that $k_2 = 1$ (convexity of the tiles is not necessary) [SD98], but for the hyperbolic plane, $k_2 \geq 2$ [Mak92]. For \mathbb{E}^3 , it is known that $2 \leq k_3 \leq 5$.

3.3 PERIODIC TILINGS

Periodic tilings have been studied intensely, in part because their applications range from ornamental design to crystallography, and in part because many techniques (algebraic, geometric, and combinatorial) are available for studying them.

GLOSSARY

Periodic tiling of \mathbb{E}^n : A tiling, not necessarily monohedral, whose symmetry group contains an n -dimensional lattice. This definition can be adapted to include “subperiodic” tilings (those whose symmetry groups contain $1 \leq k < n$ linearly independent vectors) and tilings of other spaces (for example, cylinders). Tilings in Figures 3.2.1, 3.2.3, 3.3.1, and 3.3.3 are periodic.

Fundamental domain (generating region) for a periodic tiling: A minimal subset of S whose orbit under the symmetry group of the tiling is the whole tiling. A fundamental domain may be a tile (Figure 3.2.1), a subset of a single tile (Figure 3.3.1), or a subset of tiles (two shaded tiles in Figure 3.2.3).

Orbifold (of a tiling of S): The manifold obtained by identifying points of S that are in the same orbit under the action of the symmetry group of the tiling.

Free tiling: A tiling whose symmetry group acts freely and transitively on the tiles.

k -isohedral (tiling): A tiling whose tiles belong to k transitivity classes under the action of its symmetry group. Isohedral means 1-isohedral (Figures 3.2.1, 3.3.1, and 3.3.3). The tiling in Figure 3.2.3 is 2-isohedral.

Equitransitive (tiling by polytopes): A tiling in which each combinatorial class of tiles forms a single transitivity class under the action of the symmetry group of the tiling.

k-isogonal (tiling): A tiling whose vertices belong to k transitivity classes under the action of its symmetry group. Isogonal means 1-isogonal.

k-uniform (tiling of a 2-dimensional surface): A k -isogonal tiling by regular polygons.

Uniform (tiling for $n > 2$): An isogonal tiling with congruent edges and uniform faces.

Flag of a tiling (of S): An ordered $(n+1)$ -tuple (X_0, X_1, \dots, X_n) , with X_n a tile and X_k a k -face for $0 \leq k \leq n-1$, in which $X_{i-1} \subset X_i$ for $i = 1, \dots, n$.

Regular tiling (of S): A tiling \mathcal{T} whose symmetry group is transitive on the flags of \mathcal{T} . (For $n > 2$, these are also called regular honeycombs.) See [Figure 3.3.3](#).

k-colored tiling: A tiling in which each tile has a single color, and k different colors are used. Unlike the case of map colorings, in a colored tiling adjacent tiles may have the same color.

Perfectly k -colored tiling: A k -colored tiling for which each element of the symmetry group G of the uncolored tiling effects a permutation of the colors. The ordered pair (G, Π) , where Π is the corresponding permutation group, is called a k -color symmetry group.

CLASSIFICATION OF PERIODIC TILINGS

The mathematical study of tilings (like most mathematical investigations) has been accompanied by the development and use of a variety of notations for classification of different “types” of tilings and tiles. Far from being merely names by which to distinguish types, these notations tell us the investigators’ point of view and the questions they ask. Notation may tell us the global symmetries of the tiling, or how each tile is surrounded, or the topology of its orbifold. Notation makes possible the computer implementation of investigations of combinatorial questions about tilings.

Periodic tilings are classified by symmetry groups and, sometimes, by their skeletons (of vertices, edges, ..., $(n-1)$ -faces). The groups are known as ***crystallographic groups***; up to isomorphism, there are 17 in \mathbb{E}^2 and 219 in \mathbb{E}^3 . For \mathbb{E}^2 and \mathbb{E}^3 , the most common notation for the groups has been that of the International Union of Crystallography (IUCr) [Hah83]. This is cross-referenced to earlier notations in [Sch78]. Recently developed notations include Delaney-Dress symbols [Dre87] and orbifold notation for $n = 2$ [Con92, CH02] and for $n = 3$ [CDHT01].

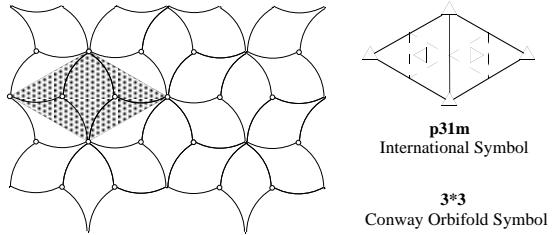
GLOSSARY

International symbol (for periodic tilings of \mathbb{E}^2 and \mathbb{E}^3): Encodes lattice type and particular symmetries of the tiling. In [Figure 3.3.1](#), the lattice unit diagram at the right encodes the symmetries of the tiling and the IUCr symbol p31m indicates that the highest-order rotation symmetry in the tiling is 3-fold, that there is no mirror normal to the edge of the lattice unit, and that there is a mirror at 60° to the edge of the lattice unit. These symbols are augmented to denote symmetry groups of perfectly 2-colored tilings.

Delaney-Dress symbol (for tilings of Euclidean, hyperbolic, or spherical space of any dimension): Associates an edge-colored and vertex-labeled

FIGURE 3.3.1

An isohedral tiling with standard IUCr lattice unit shaded; a half-leaf is a fundamental domain. The classification symbols are for the symmetry group of the tiling.



graph derived from a **chamber system** (a formal barycentric subdivision) of the tiling. In Figure 3.3.2, the nodes of the graph represent distinct triangles A, B, C, D in the chamber system, and colored edges (dashed, thick, or thin) indicate their adjacency relations. Numbers on the nodes of the graph show the degree of the tile that contains that triangle and the degree of the vertex of the tiling that is also a vertex of that triangle.

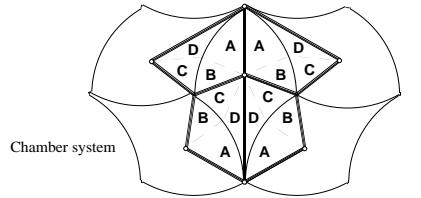


FIGURE 3.3.2

A chamber system of the tiling in Figure 3.3.1 determines the graph that is its Delaney-Dress symbol.

Orbifold notation (for symmetry groups of tilings of 2-dimensional surfaces of constant curvature): Encodes properties of the orbifold induced by the symmetry group of a periodic tiling of the Euclidean plane or hyperbolic plane, or a finite tiling of the surface of a sphere; introduced by Conway. In Figure 3.3.1, the first 3 in the orbifold symbol $3*3$ for the symmetry group of the tiling indicates there is a 3-fold rotation center (gyration point) that becomes a cone point in the orbifold, while $*3$ indicates that the boundary of the orbifold is a mirror with a corner where three mirrors intersect.

See Table 3.3.1 for the IUCr and orbifold notations for \mathbb{E}^2 .

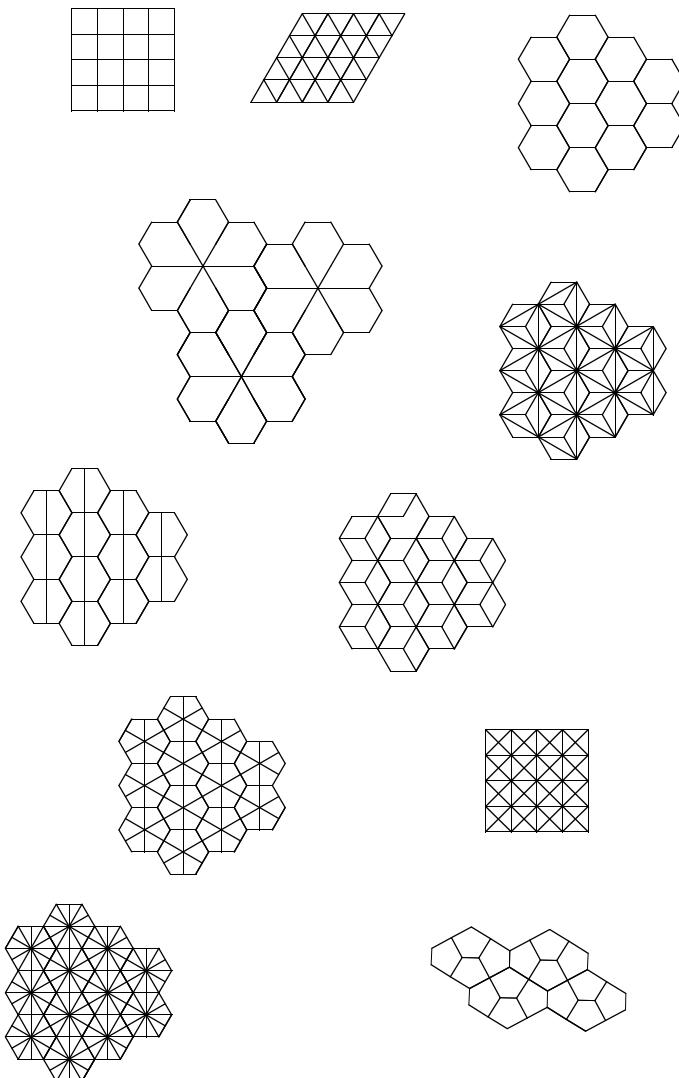
TABLE 3.3.1 IUCr and orbifold notations for the 17 symmetry groups of periodic tilings of \mathbb{E}^2 .

IUCr	ORBIFOLD	IUCr	ORBIFOLD
p1	o or o1	p3	333
pg	$\times \times$ or $1 \times \times$	p31m	$3*3$
cm	$* \times$ or $1^* \times$	p3m1	$*333$
pm	$**$ or 1^{**}	p4	442
p2	2222	p4g	4^*2
pgg	22 \times	p4m	$*442$
pmg	22*	p6	632
cmm	2*22	p6m	$*632$
pmm	*2222		

Isohedral tilings of \mathbb{E}^2 fall into 11 combinatorial classes, typified by the Laves nets (Figure 3.3.3). The Laves net for the tiling in Figure 3.3.1 is [3.6.3.6]; this gives the vertex degree sequence for each tile. In an isohedral tiling, every tile is surrounded in the same way. Grünbaum and Shephard provide an *incidence symbol* for each isohedral type by labeling and orienting the edges of each tile [GS79]. Figure 3.3.4 gives the incidence symbol for the tiling in Figure 3.3.1. The tile symbol $a^+a^-b^+b^-$ records the cycle of edges of a tile and their orientations with respect to the (arrowed) first edge (+ indicates the same, – indicates opposite orientation). The adjacency symbol b^-a^- records for each different letter edge of a single tile, beginning with the first, the edge it abuts in the adjacent tile and their relative orientations (now – indicates same, + opposite). These symbols can be augmented

FIGURE 3.3.3

The 11 Laves nets. The three regular tilings of \mathbb{E}^2 are at the top of the illustration.



to adjacency symbols to denote k -color symmetry groups. Earlier, Heesch devised signatures for the 28 types of tiles that could be fundamental domains of isohedral tilings without reflection symmetry [HK63]; this signature system was extended in [BW94].

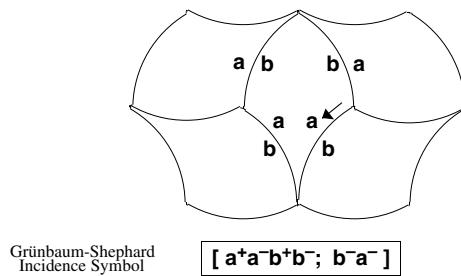


FIGURE 3.3.4

Labeling and orienting the edges of the isohedral tiling in Figure 3.3.1 determines its Grünbaum-Shephard incidence symbol.

MAIN RESULTS

1. If a finite prototile set of polygons admits an edge-to-edge tiling of the plane that has translational symmetry, then the prototile set also admits a periodic tiling [GS87].
2. The number of symmetry groups of periodic tilings in \mathbb{E}^n is finite (this is a famous theorem of Bieberbach [Bie10] that partially solved Hilbert's 18th problem: see also Chapter 62); the number of symmetry groups of corresponding tilings in hyperbolic n -space, for $n = 2$ and $n = 3$, is infinite.
3. Every k -isohedral tiling of the Euclidean plane, hyperbolic plane, or sphere can be obtained from a $(k-1)$ -isohedral tiling by a process of *splitting* (splitting an asymmetric prototile) and *gluing* (amalgamating two or more equivalent asymmetric tiles adjacent in the tiling into one new tile) [Hus93]; there are 1270 classes of normal 2-isohedral tilings and 48,231 classes of normal 3-isohedral tilings of \mathbb{E}^2 .
4. Classifying isogonal tilings in a manner analogous to isohedral ones, Grünbaum and Shephard have shown [GS78a] that there are 91 classes of normal isogonal tilings of \mathbb{E}^2 (93 classes if the tiles are marked). Similarly [GS78b], there are 26 classes of normal tilings of \mathbb{E}^2 for which the symmetry group acts transitively on the edges (30 if the tiles are marked); these tilings are called *isotoxal*. See also [GS87].
5. There are 88 combinatorial classes of periodic tilings of \mathbb{E}^3 for which the symmetry group acts transitively on the faces of the tiling [DHM93].
6. For every k , the number of k -uniform tilings of \mathbb{E}^2 is finite. There are 11 uniform tilings of \mathbb{E}^2 (also called *Archimedean*, or *semiregular*), of which 3 are regular. The Laves nets in Figure 3.3.3 are duals of these 11 uniform tilings [GS87, Sections 2.1, 2.2]. There are 28 uniform tilings of \mathbb{E}^3 [Grü94] and 20

2-uniform tilings of \mathbb{E}^2 [Krö69]; see also [GS87, Section 2.2]. In the hyperbolic plane, uniform tilings with vertex valence 3 and 4 have been classified [GS79].

7. In any equitansitive tiling of \mathbb{E}^2 by convex polygons, the maximum number of edges of any tile is 66 [DGS87].
8. There are finitely many regular tilings of \mathbb{E}^n (three for $n = 2$, one for $n = 3$, three for $n = 4$, and one for each $n > 4$) [Cox63]. There are infinitely many normal regular tilings of the hyperbolic plane, four of hyperbolic 3-space, five of hyperbolic 4-space, and none of hyperbolic n -space if $n > 4$ [Sch83, Cox54].
9. If two orbifold symbols for a tiling of the Euclidean or hyperbolic plane look exactly the same except for the numerical values of their digits, which may differ by a permutation of the natural numbers (such as *632 and *532), then the number of k -isohedral tilings for each of these orbifold types is the same [BH96].
10. There is a one-to-one correspondence between perfect k -colorings of a free tiling and the subgroups of index k of its symmetry group. See [Sen79].

OPEN PROBLEMS

1. Does every convex pentagon that tiles \mathbb{E}^2 admit a k -isohedral tiling for some $k \geq 1$, and if so, is there an upper bound on k ? (All pentagons known to tile the plane admit k -isohedral tilings, with $k \leq 3$.)
2. Classify uniform tilings of the hyperbolic plane for the cases of vertex valences greater than 4.
3. Enumerate the uniform tilings of \mathbb{E}^n for $n > 3$. (Some uniform tilings for \mathbb{E}^n , $n > 3$, are discussed in [Joh04].)
4. Delaney-Dress symbols and orbifold notations have made progress possible on the classification of k -isohedral tilings in all three 2-dimensional spaces of constant curvature; extend this work to higher-dimensional spaces.

3.4 NONPERIODIC AND APERIODIC TILINGS

Nonperiodic tilings are found everywhere in nature, from cracked glazes to biological tissues to real crystals. In a remarkable number of cases, such tilings exhibit strong regularities. For example, many such tilings have simplicial duals. Others repeat on increasingly larger scales. An even larger class of tilings are those now called repetitive, in which every bounded configuration appearing anywhere in the tiling is repeated infinitely many times throughout it (see below). Aperiodic tilings—those whose prototile sets admit only nonperiodic tilings—are particularly interesting. They were first introduced to prove the Undecidability Theorem (Section 3.1). Later, after Penrose found pairs of aperiodic prototiles (see [Figure 3.4.1](#)), they became popular in recreational mathematical circles. Their deep mathematical

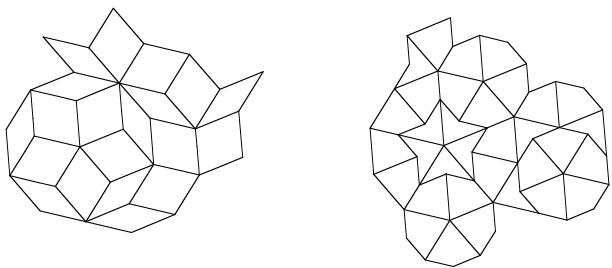


FIGURE 3.4.1

*Portions of Penrose tilings of the plane
(a) by rhombs; (b) by kites and darts.
The matching rules that force nonperiodicity are not shown (see Chapter 62).*

properties were first studied by Penrose, Conway, de Bruijn, and others. After the discovery of “quasicrystals” in 1984, aperiodic tilings became the focus of intense research. The basic ideas of this rapidly developing subject are only introduced here; they are discussed in more detail in Chapter 62.

GLOSSARY

Nonperiodic tiling: A tiling with no translation symmetry.

Hierarchical tiling: A tiling whose tiles can be composed into larger tiles, called *level-one* tiles, whose level-one tiles can be composed into level-two tiles, and so on *ad infinitum*. In some cases it is necessary to partition the original tiles before composition.

Self-similar tiling: A hierarchical tiling for which the larger tiles are copies of the prototiles (all enlarged by a constant expansion factor λ). k -rep tiles are the special case when there is just one prototile (Figure 3.2.2).

Uniquely hierarchical tiling: A tiling whose j -level tiles can be composed into $(j+1)$ -level tiles in only one way ($j = 0, 1, \dots$).

Composition rule (for a hierarchical tiling): The equations $T'_i = m_{i1}T_1 \cup \dots \cup m_{ik}T_k$, $i = 1, \dots, k$, that describe the numbers m_{ij} of each prototile T_j in the next higher level prototile T'_i . These equations define a linear map whose matrix has i, j entry m_{ij} .

Relatively dense configuration: A configuration C of tiles in a tiling for which there exists a radius r_C such that every ball of radius r_C in the tiling contains a copy of C .

Repetitive: A tiling in which every bounded configuration of tiles is relatively dense in the tiling.

Local isomorphism class: A family of tilings such that every bounded configuration of tiles that appears in any of them appears in all of the others. (For example, the uncountably many Penrose tilings with the same prototile set form a single local isomorphism class.)

Projected tiling: A tiling obtained by the canonical projection method (see Chapter 62).

Aperiodic prototile set: A prototile set that admits only nonperiodic tilings; see Figure 3.4.1.

Aperiodic tiling: A tiling with an aperiodic prototile set.

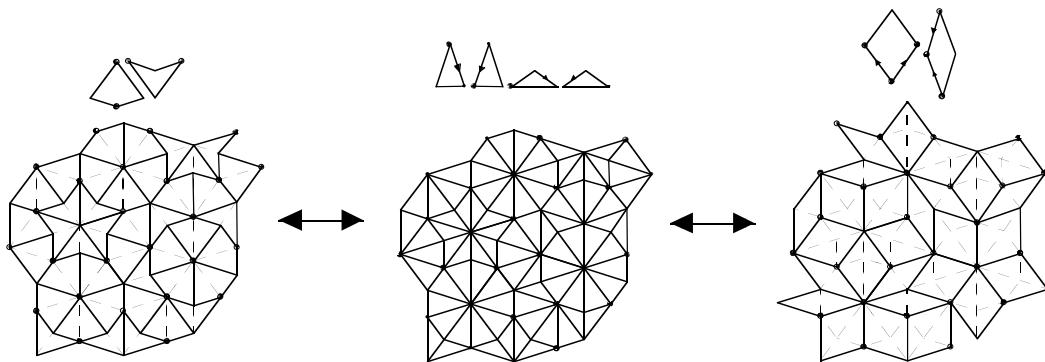
Matching rules: A list of rules for fitting together the prototiles of a given prototile set.

Mutually locally derivable tilings: Two tilings are mutually locally derivable if the tiles in either tiling can, through a process of decomposition into smaller tiles, or regrouping with adjacent tiles, or a combination of both processes, form the tiles of the other (see Figure 3.4.2).

Complex Perron number: An algebraic integer that is strictly larger in modulus than its Galois conjugates (except for its complex conjugate).

FIGURE 3.4.2

The Penrose tilings by kites and darts and by rhombs are mutually locally derivable.



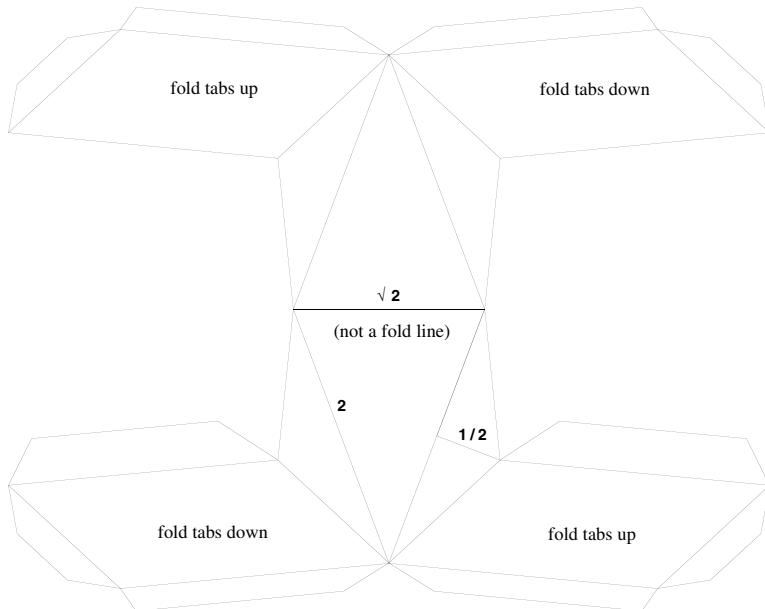
MAIN RESULTS

1. Self-similar and projected tilings are repetitive (see [Sen95]).
2. Uniquely hierarchical tilings are nonperiodic (the proof given in [GS87] for $n = 2$ extends immediately to all n). Conversely, nonperiodic self-similar tilings have the unique composition property [Sol98].
3. For each complex Perron number λ there is a self-similar tiling with expansion λ [Ken95].
4. “Irrational” projected tilings are nonperiodic (see Chapter 62).
5. The prototile sets of certain irrational projected tilings can be equipped with matching rules so that all tilings admitted by the prototile set belong to a single local isomorphism class (see Chapter 62).
6. Mutual local derivability is an equivalence relation on the set of all tilings. The existence or nonexistence of hierarchical structure and matching rules is a class property [KSB93].
7. Certain convex **biprisms** admit only nonperiodic monohedral tilings of \mathbb{E}^3 if no mirror-image copies of the tiles are allowed [Sch88]; see Figure 3.4.3. These tiles can be altered to produce nonconvex aperiodic prototiles for \mathbb{E}^3 [Dan95].

8. The prototile set of every uniquely hierarchical tiling can be equipped with matching rules that force the hierarchical structure [Goo98].

FIGURE 3.4.3

Conway's biprism consists of two prisms fused at a common rhombus face. Small angle of rhombus is $\arccos(3/4) \approx 41.4^\circ$; diagonal of prism ≈ 2.87 . When assembled, the vertices of the rhombus that is a common face of the two prisms are the poles of two 2-fold rotation axes.



OPEN PROBLEMS

Does there exist a prototile in \mathbb{E}^2 that is aperiodic? Does there exist a convex prototile for \mathbb{E}^3 that is aperiodic without restriction?

3.5 OTHER TILINGS

There is a vast literature on tilings (or dissections) of bounded regions (such as rectangles and boxes, polygons, and polytopes) by tiles to satisfy particular conditions. This and much of the recreational literature focuses on tilings by tiles of a particular type, such as tilings by rectangles, tilings by clusters of n -cubes (polyominoes—see [Chapter 15](#)—and polycubes) or n -simplices (polyiamonds in \mathbb{E}^2), or tilings by recognizable animate figures. In the search for new ways to produce tiles and tilings, both mathematicians (such as P.A. MacMahon [[Mac21](#)]) and amateurs (such as M.C. Escher [[Sch90](#)]) have contributed to the subject. Recently the search for new shapes that tile a given bounded region S has produced knotted tiles, toroidal tiles, and twisted tiles. Kuperberg and Adams have shown that for any given knot K ,

there is a monohedral tiling of \mathbb{E}^3 (or of hyperbolic 3-space, or of spherical 3-space) whose prototile is a solid torus that is knotted as K . Also, Adams has shown that, given any polyhedral submanifold M with one boundary component in \mathbb{E}^n , a monohedral tiling of \mathbb{E}^n can be constructed whose prototile has the same topological type as M [Ada95].

Other directions of research seek to broaden the definition of prototile set: in new contexts, the tiles in a tiling may be homothetic (rather than congruent) images of tiles in a prototile set, or be topological images of tiles in a prototile set. For example, a tiling of \mathbb{E}^n by polytopes in which every tile is combinatorially isomorphic to a fixed convex n -polytope (the combinatorial prototile) is said to be **monotypic**. It has been shown that in \mathbb{E}^2 , there exist monotypic face-to-face tilings by convex n -gons for all $n \geq 3$; in \mathbb{E}^3 , every convex 3-polytope is the combinatorial prototile of a monotypic tiling [Sch84a]. Many (but not all) classes of convex 3-polytopes admit monotypic face-to-face tilings [DGS83, Sch84b].

3.6 SOURCES AND RELATED MATERIALS

SURVEYS

The following surveys are useful, in addition to the references below.

[GS87]: The definitive, comprehensive treatise on tilings of \mathbb{E}^2 , state of the art as of the mid-1980s. All subsequent work (in any dimension) has taken this as its starting point for terminology, notation, and basic results. The Main Results of our Section 3.1 can be found here.

[Joh04]: A comprehensive and detailed account of uniform polytopes and honeycombs in Euclidean and non-Euclidean spaces of n dimensions.

[Moo97]: The proceedings of the NATO Advanced Study Institute on the Mathematics of Aperiodic Order, held in Waterloo, Canada in August 1995.

[Sch93]: A contemporary survey of tiling theory, especially useful for its accounts of monotypic and other kinds of tilings more general than those discussed in this chapter.

[Sch02]: A recent brief survey of tiling.

[Sen95]: [Chapters 5 – 8](#) form an introduction to the emerging theory of aperiodic tilings.

[SS94]: This book is especially useful for its account of tilings in \mathbb{E}^n by clusters of cubes.

RELATED CHAPTERS

[Chapter 15: Polyominoes](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 62: Crystals and quasicrystals](#)

REFERENCES

- [Ada95] C. Adams. Tilings of space by knotted tiles. *Math. Intelligencer*, 17:41–51, 1995.
- [BH96] L. Balke and D.H. Huson. Two-dimensional groups, orbifolds and tilings. *Geom. Dedicata*, 60:89–106, 1996.
- [Ban91] C. Bandt. Self-similar sets 5. Integer matrices and fractal tilings of R^n . *Proc. Amer. Math. Soc.*, 112:549–562, 1991.
- [Ber66] R. Berger. The undecidability of the domino problem. *Mem. Amer. Math. Soc.*, 66:1–72, 1966.
- [Bie10] L. Bieberbach. Über die Bewegungsgruppen der euklidischen Räume. (Erste Abh.). *Math. Ann.*, 70:297–336, 1910.
- [BW94] H.-G. Bigalke and H. Wippermann. *Reguläre Parkettierungen*. B.I. Wissenschaftsverlag, Mannheim, 1994.
- [Con92] J.H. Conway. The orbifold notation for surface groups. In M. Liebeck and J. Saxl, editors, *Groups, Combinatorics and Geometry*, Cambridge University Press, 1992, pages 438–447.
- [CDHT01] J.H. Conway, O. Delgado Friedrichs, D.H. Huson, and W.P. Thurston. Three-dimensional orbifolds and space groups. *Beiträge Algebra Geom.*, 42:475–507, 2001.
- [CH02] J.H. Conway and D.H. Huson. The orbifold notation for two-dimensional groups. *Structural Chemistry*, 13:247–257, 2002.
- [Cox54] H.S.M. Coxeter. Regular honeycombs in hyperbolic space. In *Proc. Internat. Congress Math.*, volume III, Nordhoff, Groningen and North-Holland, Amsterdam, 1954, pages 155–169. Reprinted in *Twelve Geometric Essays*, S. Illinois Univ. Press, Carbondale, 1968, and *The Beauty of Geometry: Twelve Essays*, Dover, Mineola, 1999.
- [Cox63] H.S.M. Coxeter. *Regular Polytopes*, second edition. Macmillan, New York, 1963. Reprinted by Dover, New York, 1973.
- [Dan95] L. Danzer. A family of 3D-spacefillers not permitting any periodic or quasiperiodic tilings. In G. Chapuis, editor, *Proc. Aperiodic '94*. World Scientific, Singapore, 1995, pages 11–17.
- [DGS83] L. Danzer, B. Grünbaum, and G.C. Shephard. Does every type of polyhedron tile three-space? *Structural Topology*, 8:3–14, 1983.
- [DGS87] L. Danzer, B. Grünbaum, and G.C. Shephard. Equitansitive tilings, or how to discover new mathematics. *Math. Mag.*, 60:67–89, 1987.
- [Del61] B.N. Delone. Proof of the fundamental theorem in the theory of stereohedra. *Dokl. Akad. Nauk SSSR*, 138:1270–1272, 1961. English translation in *Soviet Math.*, 2:812–815, 1961.
- [DS98] N. Dolbilin and D. Schattschneider. The local theorem for tilings. In J. Patera, editor, *Quasicrystals and Discrete Geometry*, Fields Inst. Monogr. 10, Amer. Math. Soc., Providence, 1998, pages 193–199.
- [Dre87] A.W.M. Dress. Presentations of discrete groups, acting on simply connected manifolds. *Adv. Math.*, 63:196–212, 1987.
- [DHM93] A.W.M. Dress, D.H. Huson, and E. Molnár. The classification of face-transitive 3-D tilings. *Acta Cryst. Sect. A*, 49:806–817, 1993.
- [Eng81] P. Engel. Über Wirkungsbereichsteilungen von kubischer Symmetrie, *Z. Kristallogr.*, 154:199–215, 1981.

- [FM84] A. Fontaine and G. Martin. Polymorphic polyominoes. *Math. Mag.*, 57:275–283, 1984.
- [Goo98] C. Goodman-Strauss. Matching rules and substitution tilings. *Ann. of Math.*, 147:181–223, 1998.
- [Grü94] B. Grünbaum. Uniform tilings of 3-space. *Geombinatorics*, 4:49–56, 1994.
- [GS77] B. Grünbaum and G.C. Shephard. The eighty-one types of isohedral tilings in the plane. *Math. Proc. Cambridge Phil. Soc.*, 82:177–196, 1977.
- [GS78a] B. Grünbaum and G.C. Shephard. The ninety-one types of isogonal tilings in the plane. *Trans. Amer. Math. Soc.*, 242:335–353, 1978 and 249:446, 1979.
- [GS78b] B. Grünbaum and G.C. Shephard. Isotoxal tilings. *Pacific J. Math.*, 76:407–430, 1978.
- [GS79] B. Grünbaum and G.C. Shephard. Incidence symbols and their applications. In D.K. Ray-Chaudhuri, editor, *Relations between Combinatorics and Other Parts of Mathematics*, volume 34 of *Proc. Sympos. Pure Math.*, Amer. Math. Soc., Providence, 1979, pages 199–244.
- [GS80] B. Grünbaum and G.C. Shephard. Tilings with congruent tiles. *Bull. Amer. Math. Soc.*, 3:951–973, 1980.
- [GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns*. Freeman, New York, 1987.
- [Hah83] T. Hahn, editor. *International Tables for Crystallography*, volume A. *Space Group Symmetry*. Reidel, Dordrecht, 1983.
- [Haj42] G. Hajos. Über einfache und mehrfache Bedeckung des n -dimensionalen Raumes mit einem Würfelgitter. *Math Z.*, 47:427–467, 1942.
- [Hee35] H. Heesch. Aufbau der Ebene aus kongruenten Bereichen. *Nachr. Ges. Wiss. Göttingen, New Ser.*, 1:115–117, 1935.
- [HK63] H. Heesch and O. Kienzle. *Flächenschluss. System der Formen lückenlos aneinanderschliessender Flachteile*. Springer-Verlag, Berlin, 1963.
- [Hus93] D.H. Huson. The generation and classification of tile- k -transitive tilings of the Euclidean plane, the sphere, and the hyperbolic plane. *Geom. Dedicata*, 47:269–296, 1993.
- [Joh04] N. Johnson. *Uniform Polytopes*. Cambridge University Press, 2004.
- [Ken95] R. Kenyon. The construction of self-similar tilings. *Geom. Funct. Anal.*, 6:471–488, 1996.
- [Ker68] R.B. Kershner. On paving the plane. *Amer. Math. Monthly*, 75:839–844, 1968.
- [KSB93] R. Klitzing, M. Schlottmann, and M. Baake. Perfect matching rules for undecorated triangular tilings with 10-, 12-, and 8-fold symmetry. *Internat. J. Modern Phys.*, 7:1453–1473, 1993.
- [Krö69] O. Krötenheerdt. Die homogenen Mosaiken n -ter Ordnung in der euklidischen Ebene, I. *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 18:273–290, 1969.
- [LS92] J.C. Lagarias and P.W. Shor. Keller's cube-tiling conjecture is false in high dimensions. *Bull. Amer. Math. Soc.*, 27:279–283, 1992.
- [Mac21] P.A. MacMahon. *New Mathematical Pastimes*. Cambridge University Press, 1921.
- [Mak92] V.S. Makarov. On a nonregular partition of n -dimensional Lobachevsky space by congruent polytopes. *Discrete Geometry and Topology, Proc. Steklov. Inst. Math.*, 4:103–106, 1992.
- [Man01] C. Mann. On Heesch's Problem and Other Tiling Problems. Dissertation, University of Arkansas, Fayetteville, 2001.
- [McM80] P. McMullen. Convex bodies which tile space by translation. *Mathematika*, 27:113–121, 1980; 28:191, 1981.

- [Min97] H. Minkowski. Allgemeine Lehrsätze über die konvexen Polyeder. *Nachr. Ges. Wiss. Göttingen. Math.-Phys. Kl.*, 198–219, 1897. In *Gesammelte Abhandlungen von Hermann Minkowski*, reprint, Chelsea, New York, 1967.
- [Min07] H. Minkowski. *Diophantische Approximationen*. Teubner, Leipzig, 1907; reprinted by Chelsea, New York, 1957.
- [Moo97] R.V. Moody. *Mathematics of Long Range Aperiodic Order*. NATO Advanced Science Institute Ser. C: Mathematical and Physical Sciences, 489, Kluwer, Dordrecht, 1997.
- [Rei28] K. Reinhardt. Zur Zerlegung der euklidischen Räume durch kongruente Würfel. *Sitzungsber. Preuss. Akad. Wiss. Berlin*, 150–155, 1928.
- [Sch78] D. Schattschneider. The plane symmetry groups: their recognition and notation. *Amer. Math. Monthly*, 85:439–450, 1978.
- [Sch90] D. Schattschneider. *Visions of Symmetry. Notebooks, Periodic Drawings, and Related Work of M.C. Escher*. Freeman, New York, 1990.
- [SD98] D. Schattschneider and N. Dolbilin. One corona is enough for the Euclidean plane. In J. Patera, editor, *Quasicrystals and Geometry*, Fields Inst. Monogr. 10, Amer. Math. Soc., Providence, 1998, pages 207–246.
- [Sch83] V. Schlegel. Theorie der homogen zusammengesetzten Raumgebilde. *Verh. (= Nova Acte) Kaiserl. Leop.-Carol. Deutsch. Akad. Naturforscher*, 44:343–459, 1883.
- [Sch88] P. Schmitt. An aperiodic prototile in space. Manuscript, 1988.
- [Sch84a] E. Schulte. Tiling three-space by combinatorially equivalent convex polytopes. *Proc. London Math. Soc.*, 49:128–140, 1984.
- [Sch84b] E. Schulte. Nontiles and nonfacets for Euclidean space, spherical complexes and convex polytopes. *J. Reine Angew. Math.*, 352:161–183, 1984.
- [Sch93] E. Schulte. Tilings. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, volume B, North Holland, Amsterdam, 1993, pages 899–932.
- [Sch02] E. Schulte. Tilings. In R.A. Myers, editor, *Encyclopedia of Physical Science and Technology*, 3rd edition, Academic Press, New York, 2002, volume 16, pages 763–782.
- [Sen79] M. Senechal. Color groups. *Discrete Applied Math.*, 1:51–73, 1979.
- [Sen81] M. Senechal. Which tetrahedra fill space? *Math. Mag.*, 54:227–243, 1981.
- [Sen88] M. Senechal. Color symmetry. *Comput. Math. Appl.*, 16:545–553, 1988.
- [Sen90] M. Senechal. *Crystalline Symmetries. An Informal Mathematical Introduction*. Adam Hilger, Bristol, 1990.
- [Sen95] M. Senechal. *Quasicrystals and Geometry*. Cambridge University Press, 1995.
- [Sol98] B. Solomyak. Nonperiodicity implies unique composition for self-similar translationally finite tilings. *Discrete Comput. Geom.*, 20:265–279, 1998.
- [SS94] S. Stein and S. Szabó. *Algebra and Tiling: Homomorphisms in the Service of Geometry*. Volume 25 of *Carus Math. Monographs*. Math. Assoc. Amer., Washington, 1994.
- [Ven54] B.A. Venkov. On a class of Euclidean polyhedra. *Vestnik Leningrad. Univ. Ser. Mat. Fiz. Khim.*, 9:11–31, 1954.
- [Vor09] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques II. *J. Reine Angew. Math.*, 136:67–181, 1909.
- [Wie82] T.W. Wieting. *The Mathematical Theory of Chromatic Plane Ornaments*. Marcel Dekker, New York, 1982.

4 HELLY-TYPE THEOREMS AND GEOMETRIC TRANSVERSALS

Rephael Wenger

INTRODUCTION

A geometric transversal is an affine subspace of \mathbb{R}^d , such as a point, line, plane, or hyperplane, that intersects every member of a family of convex sets. Eduard Helly's celebrated theorem gives conditions for the members of a family of convex sets to have a point in common, i.e., a point transversal. In Section 4.1 we highlight some of the more notable theorems related to Helly's theorem and point transversals. Section 4.2 is devoted to geometric transversal theory.

4.1 HELLY-TYPE THEOREMS

In 1913, Eduard Helly proved the following theorem:

THEOREM 4.1.1 *Helly's Theorem* [Hel23]

Let \mathcal{A} be a finite family of at least $d + 1$ convex sets in \mathbb{R}^d . If every $d + 1$ members of \mathcal{A} have a point in common, then there is a point common to all members of \mathcal{A} .

The theorem also holds for infinite families of compact convex sets.

Helly's theorem spawned numerous generalizations and variants. These theorems usually have the form: If every m members of a family of objects have property \mathcal{P} then the entire family has property \mathcal{Q} . When \mathcal{P} equals \mathcal{Q} , theorems of this form are sometimes referred to as **Helly-type** theorems. In Helly's theorem the objects are convex sets in \mathbb{R}^d , properties \mathcal{P} and \mathcal{Q} are the properties of having a point in common, and m equals $d + 1$. Most generalizations of Helly's theorem take four forms: replacing convex sets by other objects in \mathbb{R}^d , strengthening properties \mathcal{P} and \mathcal{Q} , replacing $m = d + 1$ by some other number or condition, and replacing \mathbb{R}^d by the d -dimensional sphere, \mathbb{S}^d .

The first five parts of this section discuss various generalizations of Helly's theorem. The sixth and seventh part discuss some theorems and algorithms related to Helly's theorem. The last part contains some open problems. The theorems will all be stated for finite families of convex sets. As with Helly's theorem, many of them extend to infinite families of compact convex sets by standard topological arguments.

GLOSSARY

Convex: A set $a \subseteq \mathbb{R}^d$ is convex if $x, y \in a$ implies that line segment $xy \subseteq a$.

Convex hull: The convex hull of a set of points $X \subseteq \mathbb{R}^d$ is the smallest (inclusionwise) convex set containing X .

Homology cell: Metric space a is a homology cell if it is nonempty and homologically trivial (acyclic) in all dimensions.

Translate: Set $a \subseteq \mathbb{R}^d$ is a translate of set $b \subseteq \mathbb{R}^d$ if $a = \{v + x \mid x \in b\}$ for some vector $v \in \mathbb{R}^d$.

Homothet: Set $a \subseteq \mathbb{R}^d$ is a (positive) homothet of set $b \subseteq \mathbb{R}^d$ if $a = \{v + tx \mid x \in b\}$ for some vector $v \in \mathbb{R}^d$ and scalar $t > 0$.

Flat: An affine subspace of dimension k .

Support: Hyperplane h supports convex set a if a intersects h and is contained in one of the closed halfspaces bounded by h ; k -flat f supports convex set a if a intersects f and f is contained in some supporting hyperplane of a .

Diameter: The diameter of a point set a is the supremum of the distances between pairs of points in a .

Width: The width of a closed convex set a is the smallest distance between parallel supporting hyperplanes of a .

Piercing number: The piercing number of a family \mathcal{A} of convex sets in \mathbb{R}^d is the minimum number of points needed to intersect every member of \mathcal{A} .

NOTATION

$\text{conv}(X)$: The convex hull of point set X .

$f_i(\mathcal{A})$: The number of subfamilies \mathcal{A}' of size $i + 1$ of a family \mathcal{A} of point sets such that $\bigcap_{a \in \mathcal{A}'} a \neq \emptyset$.

\mathcal{C}_j^d : The family of all sets of \mathbb{R}^d that are the unions of j or fewer convex sets.

\mathcal{K}_j^d : The family of all sets of \mathbb{R}^d that are the unions of j or fewer pairwise disjoint closed convex sets.

4.1.1 GENERALIZATIONS TO NONCONVEX SETS

In 1930, Helly himself gave the following topological generalization of his theorem:

THEOREM 4.1.2 [Hel30]

Let \mathcal{A} be a finite family of closed homology cells in \mathbb{R}^d . If the intersection of every $d + 1$ or fewer members of \mathcal{A} is a homology cell, then the intersection of all the members of \mathcal{A} is a homology cell.

Since the intersection of convex sets is a convex set and nonempty convex sets are homology cells, Theorem 4.1.2 implies Helly's theorem. Other proofs are available in [AH35, Deb70].

Helly's theorem can also be generalized to objects that are the unions of convex sets. Let \mathcal{C}_j^d be the family of all sets of \mathbb{R}^d that are the unions of j or fewer convex sets. The intersection of members of \mathcal{C}_j^d is not necessarily in \mathcal{C}_j^d .

THEOREM 4.1.3 [AK95, Mat97]

For every $j, d \geq 1$ there exists an integer $c(j, d) < \infty$ such that: If \mathcal{A} is a finite subfamily of \mathcal{C}_j^d of size at least $c(j, d)$, such that the intersection of every subfamily of \mathcal{A} is also in \mathcal{C}_j^d and such that every $c(j, d)$ members of \mathcal{A} have a point in common, then there is a point common to all the members of \mathcal{A} .

A tight version of Theorem 4.1.3 is known for objects that are the unions of pairwise disjoint closed convex sets. Let \mathcal{K}_j^d be the family of all sets of \mathbb{R}^d that are the unions of j or fewer pairwise disjoint closed convex sets.

THEOREM 4.1.4 [Mor73]

Let \mathcal{A} be a finite subfamily of \mathcal{K}_j^d of size at least $j(d+1)$ such that the intersection of every j members of \mathcal{A} is also in \mathcal{K}_j^d . If every $j(d+1)$ members of \mathcal{A} have a point in common, then there is a point common to all the members of \mathcal{A} .

The value $j(d+1)$ cannot be reduced. An elegant proof of this theorem appears in [Ame96].

4.1.2 INTERSECTIONS IN MORE THAN A POINT

The following generalizations of Helly's theorem apply to families of convex sets but strengthen both the hypothesis and the conclusion of the theorem, usually by assuming that the sets intersect in more than a single point.

THEOREM 4.1.5 [San57]

Let \mathcal{A} be a finite family of convex sets in \mathbb{R}^d . If every $d-k+1$ or fewer members of \mathcal{A} contain a k -flat in common, then there is a k -flat contained in all the members of \mathcal{A} .

THEOREM 4.1.6 [Kat71]

Let \mathcal{A} be a finite family of convex sets in \mathbb{R}^d . Let $\psi(0, d) = d+1$ and $\psi(k, d) = \max(d+1, 2(d-k+1))$ for $1 \leq k \leq d$. If the intersection of every $\psi(k, d)$ or fewer members of \mathcal{A} has dimension at least k , then the intersection of all the members of \mathcal{A} is a set of dimension at least k .

The values of $\psi(k, d)$ are tight and cannot be reduced.

THEOREM 4.1.7 [Vin39, Kle53]

Let \mathcal{A} be a finite family of at least $d+1$ convex sets in \mathbb{R}^d and let b be some convex set in \mathbb{R}^d . If every $d+1$ members of \mathcal{A} contain [intersect;are contained in] some translate of b , then some translate of b is contained in [intersects;contains] all the members of \mathcal{A} .

THEOREM 4.1.8 [BV82]

Let \mathcal{A} be a finite family of at least $d+1$ closed convex sets in \mathbb{R}^d . If the intersection of every $d+1$ members of \mathcal{A} has width at least w , then the intersection of all the members of \mathcal{A} has width at least w .

THEOREM 4.1.9 [BKP84]

Let \mathcal{A} be a finite family of at least $2d$ convex sets in \mathbb{R}^d . If the intersection of every $2d$ members of \mathcal{A} has diameter at least 1, then the intersection of all the members of \mathcal{A} has diameter at least $d^{-2d}/2$.

THEOREM 4.1.10 [BKP84]

Let \mathcal{A} be a finite family of at least $2d$ convex sets in \mathbb{R}^d . If the intersection of every $2d$ members of \mathcal{A} has volume at least 1, then the intersection of all the members of \mathcal{A} has volume at least d^{-2d^2} .

The value $2d$ in Theorems 4.1.9 and 4.1.10 is tight and cannot be reduced. The values $d^{-2d}/2$ and d^{-2d^2} are not tight and can be increased. Bárány, Katchalski, and Pach [BKP84] conjecture that the correct values are approximately $c_1 d^{-1/2}$ and $d^{-c_2 d}$ for some c_1 and c_2 .

4.1.3 REDUCING $d+1$

Reducing the number of intersecting convex sets in the hypothesis of Helly's theorem gives:

THEOREM 4.1.11 [Kle51]

Let \mathcal{A} be a finite family of convex sets in \mathbb{R}^d . For any $m \leq d+1$, if every m or fewer members of \mathcal{A} have a point in common, then every $(d-m+1)$ -flat in \mathbb{R}^d has some translate that intersects every member of \mathcal{A} and every $(d-m)$ -flat in \mathbb{R}^d is contained in a $(d-m+1)$ -flat that intersects every member of \mathcal{A} .

It is also true that if every $(d-m+1)$ -flat in \mathbb{R}^d has some translate that intersects every member of \mathcal{A} or every $(d-m)$ -flat in \mathbb{R}^d is contained in a $(d-m+1)$ -flat that intersects every member of \mathcal{A} , then every m members of \mathcal{A} have a point in common.

Theorem 4.1.11 also has a variant giving the topological structure of the set of $(d-m+1)$ -flats intersecting \mathcal{A} [BM02].

For a family \mathcal{A} of n convex sets, let $f_i(\mathcal{A})$ be the number of subfamilies \mathcal{A}' of \mathcal{A} of size $i+1$ such that the $i+1$ members of \mathcal{A}' have a point in common. ($f_i(\mathcal{A})$ is the number of faces of dimension i in the *nerve* of \mathcal{A} .) Helly's theorem states that if $f_d(\mathcal{A})$ equals $\binom{n}{d+1}$, then there is a point common to all the members of \mathcal{A} . What if $f_d(\mathcal{A})$ is some value less than $\binom{n}{d+1}$?

THEOREM 4.1.12 [Kal84, Eck85]

Let \mathcal{A} be a finite family of $n \geq d+1$ convex sets in \mathbb{R}^d . For any r where $0 \leq r \leq n-d-1$, if $f_d(\mathcal{A}) > \binom{n}{d+1} - \binom{n-r}{d+1}$, then some $d+r+1$ members of \mathcal{A} have a point in common.

THEOREM 4.1.13 [Kal84]

Let \mathcal{A} be a finite family of $n \geq d+1$ convex sets in \mathbb{R}^d . For any ρ where $0 \leq \rho \leq 1$, if $f_d(\mathcal{A}) > (1 - (1-\rho)^{d+1}) \binom{n}{d+1}$, then some $\lfloor \rho n \rfloor + 1$ members of \mathcal{A} have a point in common.

The values given in Theorems 4.1.12 and 4.1.13 are tight and cannot be reduced. Tight versions of these theorems are also known when $f_d(\mathcal{A})$ is replaced by $f_i(\mathcal{A})$ for any $i > d$. Theorem 4.1.13 is sometimes called a *fractional Helly theorem*.

The hypothesis that every $d + 1$ members of \mathcal{A} have a point in common can also be replaced by the hypothesis that out of every p members of \mathcal{A} some q have a point in common, where $p \geq q \geq d + 1$. For certain values of p and q , Hadwiger and Debrunner proved the following result on their so-called (p, q) -problem:

THEOREM 4.1.14 [HD57]

Let \mathcal{A} be a finite family of at least p convex sets in \mathbb{R}^d . If out of every p members of \mathcal{A} some q have a point in common, where $p \geq q \geq d + 1$ and $p(d - 1) < (q - 1)d$, then some set of $p - q + 1$ points intersects every member of \mathcal{A} .

The value of $p - q + 1$ is tight and cannot be reduced.

A similar theorem holds for general values of p and q , but tight bounds are not known:

THEOREM 4.1.15 [AK92]

For every $p \geq q \geq d + 1$, there exists a positive integer $c(p, q, d) < \infty$ such that: If \mathcal{A} is a finite family of at least p convex sets in \mathbb{R}^d and out of every p members of \mathcal{A} some q have a point in common, then some set of $c(p, q, d)$ points intersects every member of \mathcal{A} .

For the special case of homothets, the intersection of every two members of \mathcal{A} suffices.

THEOREM 4.1.16 [Grü59]

For every d there exists a positive integer $c(d) < \infty$ such that: If \mathcal{A} is a finite family of homothets of a convex set in \mathbb{R}^d and every two members of \mathcal{A} intersect, then some set of $c(d)$ points intersects every member of \mathcal{A} .

Tight bounds are known for circular disks in \mathbb{R}^2 .

THEOREM 4.1.17 [Dan86]

Let \mathcal{A} be a finite family of circular disks in \mathbb{R}^2 . If every two members of \mathcal{A} intersect, then some set of four points intersects every member of \mathcal{A} .

THEOREM 4.1.18 [HDK64]

Let \mathcal{A} be a finite family of circular unit disks in \mathbb{R}^2 . If every two members of \mathcal{A} intersect, then some set of three points intersects every member of \mathcal{A} .

Danzer proved Theorem 4.1.17, settling a question by Gallai on the minimum number of points needed to intersect all the members of any family of pairwise intersecting circular disks in \mathbb{R}^2 . Such problems are often called Gallai-type problems.

Theorem 4.1.13 generalizes to objects that are unions of convex sets. Let \mathcal{C}_j^d be as above.

THEOREM 4.1.19 [AK95]

For every α , $0 \leq \alpha \leq 1$, and every $j, d > 0$, there exists a constant $c(j, \alpha, d) > 0$ such that: If \mathcal{A} is a finite subfamily of \mathcal{C}_j^d of size $n \geq d + 1$ and $f_d(\mathcal{A}) > \alpha \binom{n}{d+1}$, then some $c(j, \alpha, d)n$ members of \mathcal{A} have a point in common.

Similarly, Theorem 4.1.15 generalizes to subfamilies of \mathcal{C}_j^d :

THEOREM 4.1.20 [AK95]

For every $p \geq q \geq d+1$ and every $j > 0$, there exists a positive integer $c(j, p, q, d) < \infty$ such that: If \mathcal{A} is a finite subfamily of \mathcal{C}_j^d of size at least p and out of every p members of \mathcal{A} some q have a point in common, then some set of $c(j, p, q, d)$ points intersects every member of \mathcal{A} .

4.1.4 SPHERICAL HELLY-TYPE THEOREMS

Various generalizations of convexity to a convexity structure on the d -sphere, \mathbb{S}^d , give rise to various Helly-type theorems.

GLOSSARY

Robinson-convex: A set $a \subseteq \mathbb{S}^d$ is Robinson-convex if for every $x, y \in a$ where x and y are not antipodal points, the small arc of the great circle joining x and y is contained in a .

Strongly convex: A set $a \subseteq \mathbb{S}^d$ is strongly convex if a is Robinson-convex and does not contain any antipodal points.

Convex cone: A set $a \subseteq \mathbb{R}^d$ is a convex cone centered at the origin if $x, y \in a$ implies $t_x x + t_y y \in a$ for any scalars $t_x, t_y \geq 0$.

NOTATION

$-a$: The set of points antipodal to the points in $a \subseteq \mathbb{S}^d$.

$\dim(a)$: The dimension of a manifold a with boundary. (By convention, the dimension of the empty set is -1 .)

RESULTS

THEOREM 4.1.21

Let \mathcal{A} be a finite family of at least $d+2$ strongly convex sets in \mathbb{S}^d . If every $d+2$ members of \mathcal{A} have a point in common, then there is a point common to all the members of \mathcal{A} .

THEOREM 4.1.22 [Rob42]

Let \mathcal{A} be a finite family of Robinson-convex sets in \mathbb{S}^d . If every $2d+2$ or fewer members of \mathcal{A} have a point in common, then there is a point common to all the members of \mathcal{A} .

Theorems 4.1.21 and 4.1.22 generalize to:

THEOREM 4.1.23 [SS75]

Let \mathcal{A} be a finite family of Robinson-convex sets in \mathbb{S}^d . Let m equal $\min_{a \in \mathcal{A}} [\dim(a) + \dim(a \cap -a)]$. If every $m+3$ or fewer members of \mathcal{A} have a point in common, then there is a point common to all the members of \mathcal{A} .

The values $d+2$, $2d+2$, and $m+3$ in Theorems 4.1.21, 4.1.22, and 4.1.23 can be reduced by one under certain suitable circumstances. A subset of \mathbb{S}^d is Robinson-convex if and only if it is the intersection of \mathbb{S}^d with some convex cone centered at the origin. Thus Theorems 4.1.22 and 4.1.23 can be formulated in terms of convex cones.

Weakening the hypothesis of Theorem 4.1.22 by replacing $2d+2$ by $d+1$ gives the following theorem:

THEOREM 4.1.24 [Kat77]

Let \mathcal{A} be a finite family of at least $d+n+1$ Robinson-convex sets in \mathbb{S}^d , $n > 0$. If every $d+1$ members of \mathcal{A} have a point in common, then some $d+[n/2]+1$ members of \mathcal{A} have a point in common.

A spherical variant of the topological Helly theorem (Theorem 4.1.2) generalizes Theorem 4.1.21.

THEOREM 4.1.25 [Deb70]

Let \mathcal{A} be a finite family of closed homology cells in \mathbb{S}^d . If the intersection of every $d+2$ or fewer members of \mathcal{A} is a homology cell, then the intersection of all the members of \mathcal{A} is a homology cell.

4.1.5 OTHER GENERALIZATIONS

Helly's theorem generalizes to multiple families of convex sets:

THEOREM 4.1.26 [Bár82]

Let $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{d+1}$ be nonempty finite families of convex sets in \mathbb{R}^d . If $\bigcap_{i=1}^{d+1} a_i \neq \emptyset$ for each choice of $a_i \in \mathcal{A}_i$, then $\bigcap_{a \in \mathcal{A}_i} a \neq \emptyset$ for some \mathcal{A}_i .

Setting $\mathcal{A}_1 = \mathcal{A}_2 = \dots = \mathcal{A}_{d+1}$ gives Helly's original theorem.

Dol'nikov gave a variation of Theorem 4.1.11 for multiple families of convex sets:

THEOREM 4.1.27 [Dol88]

Let $\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{d-m+2}$ be $d-m+2$ finite families of convex sets in \mathbb{R}^d , $2 \leq m \leq d+1$. If every m or fewer members of each family \mathcal{A}_i have a point in common, then there is some $(d-m+1)$ -flat in \mathbb{R}^d that intersects every member of $\mathcal{A} = \bigcup_{i=1}^{d-m+2} \mathcal{A}_i$.

Theorem 4.1.27 is a special case of a much more general theorem by Dol'nikov that gives conditions for an algebraic surface of dimension $d-m+1$ to intersect every member of $\mathcal{A} = \bigcup_{i=1}^{d-m+2} \mathcal{A}_i$.

4.1.6 RELATED THEOREMS

Helly's theorem implies and/or is implied by some notable theorems.

THEOREM 4.1.28 *Carathéodory's Theorem*

Each point of $\text{conv}(X)$, $X \subseteq \mathbb{R}^d$, is a convex combination of $d + 1$ or fewer points of X .

THEOREM 4.1.29 *Radon's Theorem*

Each set of $d + 2$ or more points in \mathbb{R}^d can be partitioned into two disjoint sets whose convex hulls have a point in common.

THEOREM 4.1.30 *Kirchberger's Theorem*

For point sets $X, Y \subseteq \mathbb{R}^d$, $\text{conv}(X) \cap \text{conv}(Y) \neq \emptyset$ if and only if $\text{conv}(X') \cap \text{conv}(Y') \neq \emptyset$ for some $X' \subseteq X$ and $Y' \subseteq Y$ where $|X| + |Y| \leq d + 2$.

A theorem similar to Carathéodory's theorem gives conditions for a point to lie in the interior of the convex hull of a set of points.

THEOREM 4.1.31 *Steinitz's Theorem*

Each point in the interior of $\text{conv}(X)$, $X \subseteq \mathbb{R}^d$, is in the interior of $\text{conv}(X')$ for some $X' \subseteq X$ and $|X'| \leq 2d$.

Theorem 4.1.26 is a generalization of Helly's theorem to multiple families of convex sets. Carathéodory's theorem has a similar, related generalization:

THEOREM 4.1.32

Let X_1, X_2, \dots, X_{d+1} be subsets of \mathbb{R}^d . If $x \in \text{conv}(X_i)$ for each X_i , then there exist points $x_i \in X_i$ such that $x \in \text{conv}(\{x_1, \dots, x_{d+1}\})$.

Finally, Radon's theorem has the following generalization:

THEOREM 4.1.33 *Tverberg's Theorem* [Tve66]

Each set of $(r - 1)(d + 1) + 1$ or more points in \mathbb{R}^d can be partitioned into r subsets whose convex hulls have a point in common.

The theorem is tight and the number $(r - 1)(d + 1) + 1$ cannot be reduced. For more details, see [Chapter 14](#).

4.1.7 RELATED ALGORITHMS

Helly's theorem provokes the following algorithmic problem: Given a family \mathcal{A} of n convex sets, find a point common to all the sets or, if there is no such point, find $d + 1$ members of \mathcal{A} that have no point in common. When \mathcal{A} is a family of n halfspaces, this problem is simply a specialized version of linear programming. Sharir and Welzl have generalized linear programming to a more abstract framework that they call *generalized linear programming*. The problem of finding a point common to n convex sets can be formulated and solved as a generalized linear programming problem. In addition, other Helly-type theorems have related algorithmic questions that can be formulated and solved as generalized linear programming problems [Ame94]. For more on linear programming and generalized linear programming, see [Chapters 45](#) and [46](#).

4.1.8 OPEN PROBLEMS

PROBLEM 4.1.34

Prove or disprove that there exists some constant c such that: If the intersection of every $2d$ members of a family \mathcal{A} of at least $2d$ convex sets in \mathbb{R}^d has diameter at least 1, then the intersection of all the members of \mathcal{A} has diameter at least $cd^{-1/2}$.

PROBLEM 4.1.35

Prove or disprove that there exists some constant c such that: If the intersection of every $2d$ members of a family \mathcal{A} of at least $2d$ convex sets in \mathbb{R}^d has volume at least 1, then the intersection of all the members of \mathcal{A} has volume at least d^{-cd} .

PROBLEM 4.1.36

Let \mathcal{A} be a finite family of translates of a convex set in \mathbb{R}^2 . Prove or disprove that if every two members of \mathcal{A} intersect, then some set of three points intersects every member of \mathcal{A} .

4.2 GEOMETRIC TRANSVERSALS

Much research on geometric transversals focuses on necessary and sufficient conditions for the existence of line, plane, or hyperplane transversals to a family \mathcal{A} of convex sets. This research includes conditions on the existence of transversals to special families of convex sets, such as translates or homothets. Most of the results apply either to line transversals in \mathbb{R}^2 or to hyperplane transversals in \mathbb{R}^d . The “order” in which a transversal intersects \mathcal{A} plays an important role in stating and proving such theorems. Given a family \mathcal{A} of convex sets, in how many different orders can \mathcal{A} be intersected by transversals?

The set of transversals to a family \mathcal{A} of convex sets forms a topological space with the usual topology associated with affine subspaces in \mathbb{R}^d , i.e., the topology inherited from the Grassmannian. What is the combinatorial structure and complexity of this space? What are efficient algorithms for constructing this space? Under what conditions does a set of k -flats form the space of transversals to some family of convex sets?

GLOSSARY

Transversal: An affine subspace $f \subseteq \mathbb{R}^d$ of dimension k is a k -transversal to a family \mathcal{A} of convex sets if f intersects every member of \mathcal{A} .

Line transversal: A 1-transversal to a family of convex sets in \mathbb{R}^d .

Hyperplane transversal: A $(d-1)$ -transversal to a family of convex sets in \mathbb{R}^d .

Separated: A family \mathcal{A} of convex sets is k -separated if no $k+2$ members of \mathcal{A} have a k -transversal.

Ordering: A k -ordering of a family $\mathcal{A} = \{a_1, \dots, a_n\}$ of convex sets is a family of orientations of $(k+1)$ -tuples of \mathcal{A} defined by a mapping $\chi : \mathcal{A}^{k+1} \rightarrow \{-1, 0, 1\}$

corresponding to the orientations of some family of points $X = \{x_1, \dots, x_n\}$ in \mathbb{R}^k . The orientation of $(a_{i_0}, a_{i_1}, \dots, a_{i_k})$ is the orientation of the corresponding points $(x_{i_0}, x_{i_1}, \dots, x_{i_k})$, i.e.,

$$\left(\operatorname{sgn} \det \begin{pmatrix} 1 & x_{i_0}^1 & \cdots & x_{i_0}^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{i_k}^1 & \cdots & x_{i_k}^k \end{pmatrix} \right).$$

Nontrivial ordering: A k -ordering is nontrivial if at least one of its orientations is nonzero.

Acyclic oriented matroid: A rank r acyclic oriented matroid on a set \mathcal{A} is a family of orientations of r -tuples of \mathcal{A} defined by a mapping $\chi : \mathcal{A}^r \rightarrow \{-1, 0, 1\}$ satisfying certain “chirotope” axioms and a condition of “acyclicity”; for more details, see Chapter 6.

Realizable acyclic oriented matroid: An acyclic oriented matroid of rank r is *realizable* if it can be represented as the family of orientations of a set of points in \mathbb{R}^{r-1} .

Geometric permutation: A geometric permutation of a $(k-1)$ -separated family \mathcal{A} of convex sets in \mathbb{R}^d is the pair of k -orderings induced by some k -transversal of \mathcal{A} .

Ackermann function: The extremely rapidly growing function defined recursively by $A(n) = A_n(n)$, where $A_1(n) = 2n$ and $A_k(n) = A_{k-1}^{(n)}(1)$, $k \geq 2$.

Davenport-Schinzel sequence: An (n, s) Davenport-Schinzel sequence is a sequence of integers, (u_1, \dots, u_m) , where $1 \leq u_i \leq n$ and $u_i \neq u_{i+1}$, that does not contain any alternating subsequence $(u_{i_1}, u_{i_2}, \dots, u_{i_{s+2}})$ of length $s+2$ such that $u_{i_1} = u_{i_3} = u_{i_5} = \dots$ and $u_{i_2} = u_{i_4} = u_{i_6} = \dots$ and $u_{i_1} \neq u_{i_2}$; for more details, see Section 46.4 of this Handbook.

Constant description complexity: A convex set has constant description complexity if it is defined by a constant number of algebraic equalities and inequalities of constant maximum degree.

Strictly convex: A compact convex set a is strictly convex if its boundary contains no line segments.

Fat: Convex set a is ρ -fat, $\rho \geq 1$, if the ratio between the radius of the smallest ball containing a and the largest ball containing a is at most ρ .

Stubby: Convex set a is ρ -stubby, $\rho \geq 1$, if it is contained in a ball of radius ρ and contains a ball of radius one.

NOTATION

$\mathcal{T}_k^d(\mathcal{A})$: The space of k -transversals to a family \mathcal{A} of convex sets in \mathbb{R}^d .

$g_k^d(n)$: The maximum number of geometric permutations induced by k -transversals of $(k-1)$ -separated families of n compact convex sets in \mathbb{R}^d .

$\alpha(n)$: The inverse of the Ackermann function.

$\lambda_s(n)$: The maximum length of an (n, s) Davenport-Schinzel sequence.

4.2.1 HADWIGER'S TRANSVERSAL THEOREM

In 1935, Vincensini asked if there is a Helly-type theorem for line transversals to a family \mathcal{A} of convex sets in \mathbb{R}^2 . In other words, is there a number m such that if every m members of \mathcal{A} are simultaneously intersected by a line then there exists a single line intersecting all the members of \mathcal{A} ? The answer is no, even for line transversals to families of pairwise disjoint line segments. Figure 4.2.1 illustrates a counterexample for m equal to four.

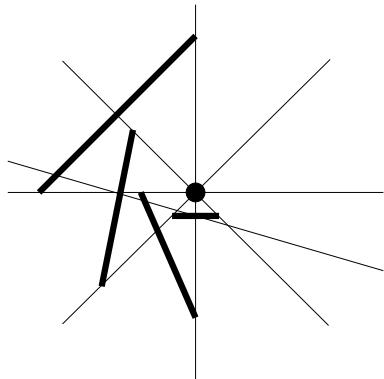


FIGURE 4.2.1

A counterexample to a Helly-type theorem for line transversals to families of convex sets in \mathbb{R}^2 : Five convex sets, four line segments and a point, where every four sets have a line transversal but all five do not.

However, in 1957 Hadwiger added a condition about the order in which every m members of \mathcal{A} are intersected by a line to give the following theorem:

THEOREM 4.2.1 *Hadwiger's Transversal Theorem* [Had57]

Let \mathcal{A} be a finite family of pairwise disjoint convex sets in \mathbb{R}^2 . If there exists a linear ordering of \mathcal{A} such that every three members of \mathcal{A} are intersected by a directed line in the given order, then \mathcal{A} has a line transversal.

As with Helly's theorem, Hadwiger's transversal theorem and most of the similar theorems in this section also apply to infinite families of compact convex sets.

Hadwiger's transversal theorem generalizes to hyperplane transversals in \mathbb{R}^d as follows:

THEOREM 4.2.2 [PW90]

Let \mathcal{A} be a finite family of connected sets in \mathbb{R}^d . If, for some k , $0 \leq k < d$, there exists a nontrivial k -ordering of \mathcal{A} such that every $k+2$ members of \mathcal{A} are intersected by an oriented k -flat consistently with that k -ordering, then \mathcal{A} has a hyperplane transversal.

An oriented k -flat f meets $\mathcal{A}' \subseteq \mathcal{A}$ consistently with a given k -ordering of \mathcal{A} if one can choose a point y_i from the intersection of each set $a_i \in \mathcal{A}'$ and f such that the orientation of every $(k+1)$ -tuple, $(y_{i_0}, y_{i_1}, \dots, y_{i_k})$, of points in f matches the orientation of the corresponding $(k+1)$ -tuple, $(a_{i_0}, a_{i_1}, \dots, a_{i_k})$, of the k -ordering. Note that Theorem 4.2.2 eliminates the assumption of pairwise disjointness in Theorem 4.2.1.

Hadwiger's transversal theorem can be generalized even further in the language of oriented matroid theory:

THEOREM 4.2.3 [AW96]

Let \mathcal{A} be a finite family of connected sets in \mathbb{R}^d . If, for some k , $0 \leq k < d$, there exists an acyclic oriented matroid of rank $k+1$ on \mathcal{A} such that every $k+2$ members of \mathcal{A} are intersected by an oriented k -flat consistently with that oriented matroid, then \mathcal{A} has a hyperplane transversal.

An oriented k -flat f meets $\mathcal{A}' \subseteq \mathcal{A}$ consistently with a given acyclic oriented matroid on \mathcal{A} if one can choose a point y_i from the intersection of each set $a_i \in \mathcal{A}'$ and f such that the orientation of every $(k+1)$ -tuple, $(y_{i_0}, y_{i_1}, \dots, y_{i_k})$, of points in f matches the orientation of the corresponding $(k+1)$ -tuple, $(a_{i_0}, a_{i_1}, \dots, a_{i_k})$, of the oriented matroid. Theorem 4.2.2 is a restriction of Theorem 4.2.3 to realizable oriented matroids.

Theorem 4.2.3 can be generalized to give the topological structure of the space of hyperplane transversals [ABM⁺02]. Essentially, if every $k+2$ members of \mathcal{A} are intersected by an oriented k -flat consistent with the given oriented matroid, then the space of hyperplane transversals has "homologically" as many hyperplanes as the set of hyperplanes containing a k -flat in \mathbb{R}^d .

Hadwiger's theorem does not generalize to line transversals in \mathbb{R}^3 even for families of pairwise disjoint convex translates [HM]. For each $m \geq 2$, there is a finite family \mathcal{A} of pairwise disjoint convex translates in \mathbb{R}^3 and a linear ordering of \mathcal{A} such that every $m-1$ members of \mathcal{A} are met by a directed line in the given order, but \mathcal{A} has no line transversal.

4.2.2 HELLY-TYPE THEOREMS

Helly-type theorems are known for infinite families and for families with some minimum separation between sets.

GLOSSARY

Limiting direction: A unit vector u is a limiting direction of an unbounded family \mathcal{A} of compact convex sets of bounded diameter if the unit vectors from the origin toward an unbounded sequence of members of \mathcal{A} approach the limit u .

Unbounded: An unbounded family of compact convex sets of bounded diameter is k -unbounded if the linear subspace spanning the set of limiting directions of \mathcal{A} has dimension at least k .

Separated: A finite family \mathcal{A} of convex sets in \mathbb{R}^d is ϵ -separated if, for every $0 < k < d$, any k of the sets can be separated from any other $d-k$ of the sets by a hyperplane more than $\epsilon D(\mathcal{A})/2$ away from all d of the sets, where $D(\mathcal{A})$ is the largest diameter of any member of \mathcal{A} .

THEOREM 4.2.4 [AGP02]

If \mathcal{A} is a k -unbounded family of compact convex sets with bounded diameter in \mathbb{R}^d , where $k < d$, and every $d+1$ members of \mathcal{A} have a k -transversal, then \mathcal{A} has a k -transversal.

THEOREM 4.2.5 [AGPW01]

For every real $\epsilon > 0$ and integer $d > 1$, there exists a constant $N_d(\epsilon)$, such that: If \mathcal{A} is an ϵ -separated family of at least $N_d(\epsilon)$ compact convex sets in \mathbb{R}^d and every $2d+2$ members of \mathcal{A} have a hyperplane transversal, then \mathcal{A} has a hyperplane transversal.

4.2.3 GALLAI-TYPE PROBLEMS

Under certain conditions a family \mathcal{A} may not have a k -transversal but there may be some small set of k -flats whose union intersects every member of \mathcal{A} .

Theorem 4.1.15 has a variant for hyperplane transversals:

THEOREM 4.2.6 [AK95]

For every $p \geq q \geq d+1$ there exists a positive integer $c(p, q, d) < \infty$ such that: If \mathcal{A} is a finite family of at least p convex sets in \mathbb{R}^d and out of every p members of \mathcal{A} some q have a hyperplane transversal, then there are $c(p, q, d)$ hyperplanes whose union intersects every member of \mathcal{A} .

In \mathbb{R}^2 almost exact minimal values of $c(p, p, 2)$ are known.

THEOREM 4.2.7 [Eck73]

Let \mathcal{A} be a finite family of convex sets in \mathbb{R}^2 . If every four members of \mathcal{A} have a line transversal, then there are two lines whose union intersects every member of \mathcal{A} .

THEOREM 4.2.8 [Eck93a]

Let \mathcal{A} be a finite family of convex sets in \mathbb{R}^2 . If every three members of \mathcal{A} have a line transversal, then there are four lines whose union intersects every member of \mathcal{A} .

It is conjectured, but not proven, that the number four in the conclusion of Theorem 4.2.8 can be reduced to three. It cannot be reduced to two.

Theorem 4.2.6 generalizes to subfamilies of \mathcal{C}_j^d , i.e., families whose members are the unions of convex sets:

THEOREM 4.2.9 [AK95]

For every $p \geq q \geq d+1$ and every j there exists a positive integer $c(j, p, q, d) < \infty$ such that: If \mathcal{A} is a finite subfamily of \mathcal{C}_j^d of size at least p and out of every p members of \mathcal{A} some q have a hyperplane transversal, then there are $c(j, p, q, d)$ hyperplanes whose union intersects every member of \mathcal{A} .

4.2.4 TRANSLATES

Many special theorems apply to transversals of families of translates. Most noteworthy is the following Helly-type theorem conjectured by Grünbaum in 1958 and proved by Tverberg in 1989:

THEOREM 4.2.10 [Tve89]

Let \mathcal{A} be a family of pairwise disjoint translates of a compact convex set in \mathbb{R}^2 . If every five or fewer members of \mathcal{A} have a line transversal, then \mathcal{A} has a line transversal.

The number five cannot be reduced, even for unit disks [AGPW00].

Under the weaker condition that every three members of \mathcal{A} have a line transversal, the following theorem holds:

THEOREM 4.2.11 [Hol03]

Let \mathcal{A} be a family of pairwise disjoint translates of a compact convex set in \mathbb{R}^2 . If every three members of \mathcal{A} have a line transversal, then some subfamily $\mathcal{A}' \subseteq \mathcal{A}$ containing all but 22 members of \mathcal{A} has a line transversal.

Katchalski and Lewis [KL80] proved this theorem with looser bounds, which were later improved by Tverberg. The current bound of 22 is a recent result by Holmsen [Hol03]. The number 22 is not known to be tight and can possibly be reduced. Katchalski and Lewis conjectured that the correct number is two, but Holmsen [Hol03] gave an example showing that the number is at least four.

Versions of Theorems 4.2.10 and 4.2.11 exist for families of pairwise disjoint ρ -stubby convex sets where the constants are replaced by functions of ρ .

The condition that the members of \mathcal{A} are pairwise disjoint can also be weakened.

THEOREM 4.2.12 [Rob97]

For every $j > 0$ there exists a number $c(j)$ such that: If \mathcal{A} is a family of translates of a compact convex set in \mathbb{R}^2 such that the intersection of any j members of \mathcal{A} is empty and such that every $c(j)$ or fewer members of \mathcal{A} have a line transversal, then \mathcal{A} has a line transversal.

Recently, Holmsen and Matoušek [HM] showed that Theorem 4.2.10 does not generalize to line transversals of pairwise disjoint convex translates in \mathbb{R}^3 . For any integer $n > 2$, there exists a family \mathcal{A} of pairwise disjoint translates of n compact convex sets such that every $n - 1$ members of \mathcal{A} have a line transversal, but \mathcal{A} does not have a line transversal. Theorem 4.2.11 also does not generalize to line transversals in \mathbb{R}^3 .

In another recent result, Holmsen, Katchalski, and Lewis proved that there is a Helly-type theorem for line transversals of disjoint unit balls in \mathbb{R}^3 :

THEOREM 4.2.13 [HKL03]

There exists an integer $m \leq 46$ such that: If \mathcal{A} is a family of pairwise disjoint unit balls in \mathbb{R}^3 and every m or fewer members of \mathcal{A} have a line transversal, then \mathcal{A} has a line transversal.

Helly-type theorems are also known for hyperplane transversals of families of translates of convex polytopes:

THEOREM 4.2.14 [Grü64]

Let \mathcal{A} be a family of translates of a convex polytope in \mathbb{R}^d with n vertices. If every $\binom{n}{2}(d + 1)$ or fewer members of \mathcal{A} have a hyperplane transversal, then \mathcal{A} has a hyperplane transversal.

THEOREM 4.2.15 [Grü64]

Let \mathcal{A} be a family of translates of a centrally symmetric convex polytope in \mathbb{R}^d with n vertices. If every $\lfloor \frac{n}{2} \rfloor(d+1)$ or fewer members of \mathcal{A} have a hyperplane transversal, then \mathcal{A} has a hyperplane transversal.

The number $\lfloor \frac{n}{2} \rfloor(d+1)$ is tight and cannot be reduced.

4.2.5 GALLAI-TYPE PROBLEMS ON TRANSLATES

Eckhoff established Gallai-type results for line transversals of translates in \mathbb{R}^2 :

THEOREM 4.2.16 [Eck73]

Let \mathcal{A} be a finite family of translates of a convex set in \mathbb{R}^2 . If every three members of \mathcal{A} have a line transversal, then there are two parallel lines whose union intersects every member of \mathcal{A} .

In higher dimensions, Eckhoff showed:

THEOREM 4.2.17 [Eck69]

For every $k \geq 0$ there exists a number $c(k)$ such that: If \mathcal{A} is a finite family of translates of a convex set in \mathbb{R}^d and every $k+2$ members of \mathcal{A} have a k -transversal, then there are $c(k)$ parallel k -flats whose union intersects every member of \mathcal{A} .

4.2.6 SPACE OF TRANSVERSALS

Given a family \mathcal{A} of convex sets in \mathbb{R}^d , let $\mathcal{T}_k^d(\mathcal{A})$ be the space of all k -transversals of \mathcal{A} . If the members of \mathcal{A} are closed, then the boundary of $\mathcal{T}_k^d(\mathcal{A})$ consists of k -flats that support one or more members of \mathcal{A} . This boundary can be partitioned into subspaces of k -flats that support the same subfamily of \mathcal{A} . Each of these subspaces can be further partitioned into connected components. The combinatorial complexity of $\mathcal{T}_k^d(\mathcal{A})$ is the number of such connected components.

Even in \mathbb{R}^2 , the boundaries of two convex sets can intersect in an arbitrarily large number of points and have an arbitrarily large number of common supporting lines. Thus the space of line transversals to two convex sets in \mathbb{R}^2 can have arbitrarily large combinatorial complexity. However, if \mathcal{A} consists of pairwise disjoint convex sets in \mathbb{R}^2 or, more generally, suitably separated convex sets in \mathbb{R}^d , then the complexity is bounded. If the convex sets have constant description complexity, then again the transversal space complexity is bounded. Finally, if the sets are convex polytopes, then the transversal space is bounded by the total number of polytope faces. [Table 4.2.1](#) gives bounds on the transversal space complexity for various families of sets.

The function $\alpha(n)$ is the very slowly growing inverse of the Ackermann function. The function $\lambda_s(n)$ is the maximum length of an (n,s) Davenport-Schinzel sequence. Function $\lambda_s(n)$ equals $n\alpha(n)^{O(\alpha(n)^s - 3)}$.

In \mathbb{R}^2 the bounds are based on the maximum number s of common supporting lines per pair of convex sets, i.e., on the number of lines tangent to both sets that do not separate the sets. For sets of constant description complexity, this maximum s is bounded. Note that $\lambda_s(n) \in O(n^{1+\epsilon})$ for any $\epsilon > 0$.

TABLE 4.2.1 Bounds on $\mathcal{T}_k^d(\mathcal{A})$.

FAMILY	k	d	COMPLEXITY OF $\mathcal{T}_k^d(\mathcal{A})$	SOURCE
($d-2$)-separated family of n compact and strictly convex sets	$d-1$	d	$O(n^{d-1})$	[CGP ⁺ 94]
n connected sets such that any two sets have at most s common supporting lines	1	2	$O(\lambda_s(n))$	[AB87]
n convex sets with const. description complexity	1	3	$O(n^{3+\epsilon})$ for any $\epsilon > 0$	[KS]
n convex sets with const. description complexity	2	3	$O(n^{2+\epsilon})$ for any $\epsilon > 0$	[ASS96]
n convex sets with const. description complexity	3	4	$O(n^{3+\epsilon})$ for any $\epsilon > 0$	[KS]
n line segments	$d-1$	d	$O(n^{d-1})$	[PS89]
Convex polytopes with a total of n_f faces	$d-1$	d	$O(n_f^{d-1} \alpha(n_f))$	[PS89]
Convex polytopes with a total of n_f faces	1	3	$O(n_f^{3+\epsilon})$ for any $\epsilon > 0$	[Aga94]
n ($d-1$)-balls	$d-1$	d	$O(n^{\lceil d/2 \rceil})$	[HII ⁺ 93]

The asymptotic bounds on the worst case complexity of hyperplane transversals ($k = d - 1$) to line segments and convex polytopes are tight. There are examples of families \mathcal{A} of convex polytopes where the complexity of $\mathcal{T}_1^3(\mathcal{A})$ is $\Omega(n_f^3)$.

4.2.7 GEOMETRIC PERMUTATIONS

A directed line intersects a family \mathcal{A} of pairwise disjoint convex sets in a well-defined order. Thus an undirected line transversal to \mathcal{A} induces a pair of linear orderings or “permutations” on \mathcal{A} consisting of the two orders in which oriented versions of the line intersect \mathcal{A} . Similarly an oriented k -transversal f intersects a $(k-1)$ -separated family $\mathcal{A} = \{a_1, \dots, a_n\}$ of convex sets in a well-defined k -ordering. The orientation of $(a_{i_0}, a_{i_1}, \dots, a_{i_k})$ is the orientation in f of any corresponding set of points $(x_{i_0}, x_{i_1}, \dots, x_{i_k})$, where $x_{i_j} \in a_{i_j} \cap f$. An unoriented k -transversal to a $(k-1)$ -separated family \mathcal{A} of convex sets induces a pair of k -orderings on \mathcal{A} , consisting of the two k -orderings in which oriented versions of the k -transversal intersect \mathcal{A} . Each such pair of k -orderings is called a *geometric permutation* of \mathcal{A} .

If \mathcal{A} is $(k-1)$ -separated, then two k -transversals that induce different geometric permutations on \mathcal{A} must lie in different connected components of $\mathcal{T}_k^d(\mathcal{A})$. The converse also holds for hyperplane transversals.

THEOREM 4.2.18 [Wen90b]

Let \mathcal{A} be a $(d-2)$ -separated family of compact convex sets in \mathbb{R}^d . Two hyperplane transversals induce the same geometric permutation on \mathcal{A} if and only if they lie in the same connected component of $\mathcal{T}_{d-1}^d(\mathcal{A})$.

Consider geometric permutations induced by k -transversals of $(k-1)$ -separated families of compact convex sets in \mathbb{R}^d . Let $g_k^d(n)$ be the maximum number of such geometric permutations over all such families \mathcal{A} of size n . The following is known about $g_k^d(n)$:

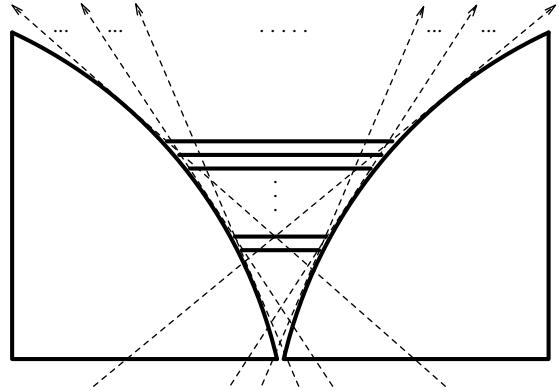


FIGURE 4.2.2

An example of n convex sets, two quarter circles and $n - 2$ line segments, that have $2n - 2$ geometric permutations. (From [GPW93], with permission.)

THEOREM 4.2.19

1. $g_1^2(n) = 2n - 2$ [ES90]. (See Figure 4.2.2.)
2. $g_1^d(n) = \Omega(n^{d-1})$ [KLL92].
3. $g_{d-1}^d(n) = O(n^{d-1})$ [Wen90a].
4. $g_k^d(n) = O(k)^{d^2} \left(\binom{2^{k+1}-2}{k} \binom{n}{k+1} \right)^{k(d-k)}$ (or $g_k^d(n) = O(n^{k(k+1)(d-k)})$ for fixed k and d) [GPW96].

For families of pairwise disjoint translates, special bounds hold. Note that such families also have a special Helly-type transversal theorem (Theorem 4.2.10).

THEOREM 4.2.20 [KLL87, KLL92]

A family of pairwise disjoint translates of a compact convex set in \mathbb{R}^2 has at most three geometric permutations.

A family of pairwise disjoint ρ -stubby compact convex sets in \mathbb{R}^2 has at most c_ρ geometric permutations, where the constant c_ρ depends upon ρ .

Starting with work by Smorodinsky, Mitchell and Sharir [SMS00], there has been substantial recent progress on geometric permutations of line transversals to balls.

THEOREM 4.2.21 [SMS00]

The maximum number of geometric permutations induced by line transversals to a family of n pairwise disjoint balls in \mathbb{R}^d is $\Theta(n^{d-1})$.

THEOREM 4.2.22 [HXC01, KSZ03]

The maximum number of geometric permutations induced by line transversals to a sufficiently large, finite family of pairwise disjoint unit balls in \mathbb{R}^d is four.

THEOREM 4.2.23 [SMS00]

The maximum number of geometric permutations induced by line transversals to a sufficiently large, finite family of pairwise disjoint unit disks in \mathbb{R}^2 is two.

Theorem 4.2.21 generalizes to families of ρ -fat convex sets. The constant of proportionality depends on ρ and d .

THEOREM 4.2.24 [KV01]

The maximum number of geometric permutations induced by line transversals to a family of n ρ -fat convex sets $\Theta(n^{d-1})$.

4.2.8 TRANSVERSAL ALGORITHMS

As may be expected, the time to construct a representation of $\mathcal{T}_k^d(\mathcal{A})$ is directly related to the complexity of $\mathcal{T}_k^d(\mathcal{A})$. Most algorithms use upper and lower envelopes to represent and construct $\mathcal{T}_k^d(\mathcal{A})$. (See Chapter 24.) Table 4.2.2 gives known bounds on the worst case time to construct a representation of the space $\mathcal{T}_k^d(\mathcal{A})$ for various families of convex sets. All sets are assumed to be compact. As noted, for $\mathcal{T}_1^3(\mathcal{A})$ and $\mathcal{T}_3^4(\mathcal{A})$, the bound is for expected running time, not worst case time.

TABLE 4.2.2 Algorithms to construct $\mathcal{T}_k^d(\mathcal{A})$.

FAMILY	k	d	TIME COMPLEXITY	SOURCE
$(d-2)$ -separated family of n strictly convex sets with constant description complexity	$d-1$	d	$O(n^{d-1} \log^2(n))$	[CGP ⁺ 94]
n convex sets with const. description complexity s.t. any two sets have at most s common supporting lines	1	2	$O(\lambda_s(n) \log n)$	[AB87]
n convex sets with const. description complexity	1	3	$O(n^{3+\epsilon}) \forall \epsilon > 0$ (exp'd.)	[KS]
n convex sets with const. description complexity	2	3	$O(n^{2+\epsilon}) \forall \epsilon > 0$	[ASS96]
n convex sets with const. description complexity	3	4	$O(n^{3+\epsilon}) \forall \epsilon > 0$ (exp'd.)	[KS]
Convex polygons with a total of n_f faces	1	2	$\Theta(n_f \log(n_f))$	[Her89]
Convex polytopes with a total of n_f faces	1	3	$O(n_f^{3+\epsilon}) \forall \epsilon > 0$	[PS92]
Convex polytopes with a total of n_f faces	2	3	$\Theta(n_f^2 \alpha(n_f))$	[EGS89]
Convex polytopes with a total of n_f faces	$d-1$	d	$O(n_f^d), d > 3$	[PS89]
n $(d-1)$ -balls	$d-1$	d	$O(n^{\lceil d/2 \rceil + 1})$	[HII ⁺ 93]
n convex homothets	1	2	$O(n \log(n))$	[Ede85]
n pairwise disjoint translates of a convex set with constant description complexity	1	2	$O(n)$	[EW89]

The model of computation used in the lower bound for the time to construct $\mathcal{T}_1^2(\mathcal{A})$ is an algebraic decision tree. In the worst case, $\mathcal{T}_2^3(\mathcal{A})$ may have $\Omega(n_f^2 \alpha(n_f))$ complexity, which gives the lower bound on constructing $\mathcal{T}_2^3(\mathcal{A})$. Similarly, $\mathcal{T}_1^3(\mathcal{A})$ may have $\Omega(n_f^3)$ complexity, giving an $\Omega(n_f^3)$ lower bound on the time to construct $\mathcal{T}_1^3(\mathcal{A})$.

4.2.9 CONVEXITY ON THE AFFINE GRASSMANNIAN

Goodman and Pollack in [GP95] extend the notion of point set convexity to convexity of a set of k -flats in \mathbb{R}^d , giving several alternate and equivalent formulations

of this convexity structure. In one such formulation, a set \mathcal{F} of k -flats is *convex* if \mathcal{F} is the transversal space of some family of convex point sets. They explore the conditions for \mathcal{F} to be such a transversal space.

GLOSSARY

Convex (set of k -flats): A set \mathcal{F} of k -flats is convex if \mathcal{F} is the space of k -transversals to some (possibly infinite) family of convex sets in \mathbb{R}^d .

Surround: A set \mathcal{F} of k -flats surrounds a k -flat f if there is some j -flat g containing f such that every $(j-1)$ -flat containing f and lying in g strictly separates two members of \mathcal{F} also lying in g .

Convex hull (of a set of k -flats): The convex hull of a set \mathcal{F} of k -flats in \mathbb{R}^d is the set of all k -flats surrounded by \mathcal{F} in \mathbb{R}^d .

THEOREM 4.2.25 [GP95]

A set \mathcal{F} of k -flats in \mathbb{R}^d is the space of k -transversals to some (possibly infinite) family of convex point sets in \mathbb{R}^d if and only if every k -flat surrounded by \mathcal{F} is in \mathcal{F} .

There is no Helly-type theorem for convex sets of k -flats in \mathbb{R}^d since such a theorem would be equivalent to a Helly-type theorem for k -transversals in \mathbb{R}^d . Such convex sets may have many connected components and may even have arbitrarily complex homology. Under suitable conditions in \mathbb{R}^3 , however, each such connected component is itself convex.

THEOREM 4.2.26 [GPW95]

Let \mathcal{F} be the space of all line transversals to a finite family of pairwise disjoint compact convex sets in \mathbb{R}^3 . Each connected component of \mathcal{F} can itself be represented as the space of line transversals to some finite family of pairwise disjoint compact convex sets in \mathbb{R}^3 .

The theorem does not hold for line transversals to infinite families of noncompact convex sets.

4.2.10 OPEN PROBLEMS

PROBLEM 4.2.27

Let \mathcal{A} be a finite family of convex sets in \mathbb{R}^2 . Prove or disprove that if every three members of \mathcal{A} have a line transversal, then there are three lines whose union intersects every member of \mathcal{A} .

PROBLEM 4.2.28

Let \mathcal{A} be a family of pairwise disjoint translates of a compact convex set in \mathbb{R}^2 . Prove or disprove that if every three members of \mathcal{A} have a line transversal, then some subfamily $\mathcal{A}' \subseteq \mathcal{A}$ containing all but four members of \mathcal{A} has a line transversal.

PROBLEM 4.2.29

Prove or disprove that there exists some m such that: If every m or fewer members of a finite 1-separated family \mathcal{A} of unit balls have a plane transversal, then \mathcal{A} has a plane transversal. (A family is 1-separated if no three members have a line transversal.) Prove or disprove the same for a 1-separated family \mathcal{A} of convex translates.

PROBLEM 4.2.30

Prove or disprove that there exist some m and r such that: If every m members of a finite family \mathcal{A} of at least m convex sets in \mathbb{R}^3 have a line transversal, then there are r lines whose union intersects every member of \mathcal{A} . Prove a similar result under the conditions that out of every p members of \mathcal{A} some q have a line transversal, for suitably large p and q . Generalize to k -transversals in \mathbb{R}^d .

PROBLEM 4.2.31

Let \mathcal{F} be the space of all k -transversals to a finite $(k-1)$ -separated family of compact convex sets in \mathbb{R}^d . Prove or disprove that each connected component of \mathcal{F} can itself be represented as the space of k -transversals to some family of convex sets in \mathbb{R}^d .

4.3 SOURCES AND RELATED MATERIAL

SURVEYS

The following surveys and books are excellent sources for many of the results in this chapter.

- [DGK63]: The classical survey of Helly's theorem and related results.
- [Eck93b]: A more recent survey of Helly's theorem and related results, updating the material in [DGK63].
- [GPW93]: A survey of geometric transversal theory.
- [SA95]: Contains applications of Davenport-Schinzel sequences and upper and lower envelopes to geometric transversals.
- [Mat02]: A recent text covering many aspects of discrete geometry including the fractional Helly theorem and the (p, q) -problem.

RELATED CHAPTERS

- Chapter 2: Packing and covering
- Chapter 3: Tilings
- Chapter 6: Oriented matroids
- Chapter 14: Topological methods
- Chapter 18: Face numbers of polytopes and complexes
- Chapter 24: Arrangements

-
- [Chapter 45](#): Linear programming
[Chapter 46](#): Mathematical programming
[Chapter 47](#): Algorithmic motion planning

REFERENCES

- [AB87] M.J. Atallah and C. Bajaj. Efficient algorithms for common transversals. *Inform. Process. Lett.*, 25:87–91, 1987.
- [ABM⁺02] J.L. Arocha, J. Bracho, L. Montejano, D. Oliveros, and R. Strausz. Separoids, their categories and a Hadwiger-type theorem for transversals. *Discrete Comput. Geom.*, 27:377–385, 2002.
- [Aga94] P.K. Agarwal. On stabbing lines for polyhedra in 3d. *Comput. Geom. Theory Appl.*, 4:177–189, 1994.
- [AGP02] B. Aronov, J.E. Goodman, and R. Pollack. A Helly-type theorem for higher-dimensional transversals. *Comput. Geom. Theory Appl.*, 21:177–183, 2002.
- [AGPW00] B. Aronov, J.E. Goodman, R. Pollack, and R. Wenger. On the Helly number for hyperplane transversals to unit balls. In G. Kalai and V. Klee, editors, The Branko Grünbaum Birthday Issue, *Discrete Comput. Geom.*, 24:171–176, 2000.
- [AGPW01] B. Aronov, J.E. Goodman, R. Pollack, and R. Wenger. A Helly-type theorem for hyperplane transversals to well-separated convex sets. In P.K. Agarwal, D. Halperin, and R. Pollack, editors, The Micha Sharir Birthday Issue, *Discrete Comput. Geom.*, 25:507–517, 2001.
- [AH35] P. Alexandroff and H. Hopf. *Topologie I*, volume 45 of *Grundlehren der Math.* Julius Springer, Berlin, Germany, 1935.
- [AK92] N. Alon and D. Kleitman. Piercing convex sets and the Hadwiger–Debrunner (p, q) -problem. *Adv. Math.*, 96:103–112, 1992.
- [AK95] N. Alon and G. Kalai. Bounding the piercing number. *Discrete Comput. Geom.*, 13:245–256, 1995.
- [Ame94] N. Amenta. Helly-type theorems and generalized linear programming. *Discrete Comput. Geom.*, 12:241–261, 1994.
- [Ame96] N. Amenta. A short proof of an interesting Helly-type theorem. *Discrete Comput. Geom.*, 15:423–427, 1996.
- [ASS96] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.
- [AW96] L. Anderson and R. Wenger. Oriented matroids and hyperplane transversals. *Adv. Math.*, 119:117–125, 1996.
- [Bár82] I. Bárány. A generalization of Carathéodory’s theorem. *Discrete Math.*, 40:141–152, 1982.
- [BKP84] I. Bárány, M. Katchalski, and J. Pach. Helly’s theorem with volumes. *Amer. Math. Monthly*, 91:362–365, 1984.
- [BM02] J. Bracho and L. Montejano. Helly-type theorems on the homology of the space of transversals. *Discrete Comput. Geom.*, 27:387–393, 2002.
- [BV82] E.O. Buchman and F.A. Valentine. Any new Helly numbers? *Amer. Math. Monthly*, 89:370–375, 1982.

- [CGP⁺94] S.E. Cappell, J.E. Goodman, J. Pach, R. Pollack, M. Sharir, and R. Wenger. Common tangents and common transversals. *Adv. Math.*, 106:198–215, 1994.
- [Dan86] L. Danzer. Zur Lösung des Gallaischen Problems über Kreisscheiben in der euklidischen Ebene. *Studia Sci. Math. Hungar.*, 21:111–134, 1986.
- [Deb70] H. Debrunner. Helly type theorems derived from basic singular homology. *Amer. Math. Monthly*, 77:375–380, 1970.
- [DGK63] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Convexity*, volume 7 of *Proc. Symp. Pure Math.*, pages 101–180. Amer. Math. Soc., Providence, 1963.
- [Dol88] V.L. Dol'nikov. Generalized transversals of families of sets in \mathbb{R}^n and connections between the Helly and Borsuk theorems. *Soviet Math. Dokl.*, 36:519–522, 1988.
- [Eck69] J. Eckhoff. *Transversalenprobleme vom Gallai'schen Typ*. Ph.D. dissertation, Georg-August-Universität, Göttingen, 1969.
- [Eck73] J. Eckhoff. Transversalenprobleme in der Ebene. *Arch. Math.*, 24:195–202, 1973.
- [Eck85] J. Eckhoff. An upper bound theorem for families of convex sets. *Geom. Dedicata*, 19:217–227, 1985.
- [Eck93a] J. Eckhoff. A Gallai-type transversal problem in the plane. *Discrete Comput. Geom.*, 9:203–214, 1993.
- [Eck93b] J. Eckhoff. Helly, Radon and Carathéodory type theorems. In *Handbook of Convex Geometry*, pages 389–448. North-Holland, Amsterdam, 1993.
- [Ede85] H. Edelsbrunner. Finding transversals for sets of simple geometric figures. *Theoret. Comput. Sci.*, 35:55–69, 1985.
- [EGS89] H. Edelsbrunner, L.J. Guibas, and M. Sharir. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete Comput. Geom.*, 4:311–336, 1989.
- [ES90] H. Edelsbrunner and M. Sharir. The maximum number of ways to stab n convex non-intersecting sets in the plane is $2n - 2$. *Discrete Comput. Geom.*, 5:35–42, 1990.
- [EW89] P. Egyed and R. Wenger. Stabbing pairwise-disjoint translates in linear time. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 364–369, 1989.
- [GP95] J.E. Goodman and R. Pollack. Foundations of a theory of convexity on affine Grassmann manifolds. *Mathematika*, 42:305–328, 1995.
- [GPW93] J.E. Goodman, R. Pollack, and R. Wenger. Geometric transversal theory. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 163–198. Springer-Verlag, Heidelberg, 1993.
- [GPW95] J.E. Goodman, R. Pollack, and R. Wenger. On the connected components of the space of line transversals to a family of convex sets. *Discrete Comput. Geom.*, 13:469–476, 1995.
- [GPW96] J.E. Goodman, R. Pollack, and R. Wenger. Bounding the number of geometric permutations induced by k -transversals. *J. Combin. Theory Ser. A*, 75:187–197, 1996.
- [Grü59] B. Grünbaum. On intersections of similar sets. *Portugal Math.*, 18:155–164, 1959.
- [Grü64] B. Grünbaum. Common secants for families of polyhedra. *Arch. Math.*, 15:76–80, 1964.
- [Had57] H. Hadwiger. Über Eibereiche mit gemeinsamer Treffgeraden. *Portugal Math.*, 6:23–29, 1957.
- [HD57] H. Hadwiger and H. Debrunner. Über eine Variante zum Helly'schen Satz. *Arch. Math.*, 8:309–313, 1957.

- [HDK64] H. Hadwiger, H. Debrunner, and V. Klee. *Combinatorial Geometry in the Plane*. Holt, Rinehart & Winston, New York, 1964.
- [Hel23] E. Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresber. Deutsch. Math.-Verein.*, 32:175–176, 1923.
- [Hel30] E. Helly. Über Systeme abgeschlossener Mengen mit gemeinschaftlichen Punkten. *Monatsh. Math.*, 37:281–302, 1930.
- [Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.
- [HII⁺93] M.E. Houle, H. Imai, K. Imai, J.-M. Robert, and P. Yamamoto. Orthogonal weighted linear L_1 and L_∞ approximation and applications. *Discrete Appl. Math.*, 43:217–232, 1993.
- [HKL03] A. Holmsen, M. Katchalski, and T. Lewis. A Helly-type theorem for line transversals to disjoint unit balls. *Discrete Comput. Geom.*, 29:595–602, 2003.
- [HM] A. Holmsen and J. Matoušek. No Helly theorem for stabbing translates by lines in \mathbb{R}^3 . Unpublished manuscript.
- [Hol03] A. Holmsen. New bounds on the Katchalski-Lewis transversal problem. *Discrete Comput. Geom.*, 29:395–408, 2003.
- [HXC01] Y. Huang, J. Xu, and D.Z. Chen. Geometric permutations of high dimensional spheres. In *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 244–245, 2001.
- [Kal84] G. Kalai. Intersection patterns of convex sets. *Israel J. Math.*, 48:161–174, 1984.
- [Kat71] M. Katchalski. The dimension of intersections of convex sets. *Israel J. Math.*, 10:465–470, 1971.
- [Kat77] M. Katchalski. A Helly type theorem on the sphere. *Proc. Amer. Math. Soc.*, 66:119–122, 1977.
- [KL80] M. Katchalski and T. Lewis. Cutting families of convex sets. *Proc. Amer. Math. Soc.*, 79:457–461, 1980.
- [Kle51] V. Klee. On certain intersection properties of convex sets. *Canad. J. Math.*, 3:272–275, 1951.
- [Kle53] V. Klee. The critical set of a convex body. *Amer. J. Math.*, 75:178–188, 1953.
- [KLL87] M. Katchalski, T. Lewis, and A. Liu. Geometric permutations of disjoint translates of convex sets. *Discrete Math.*, 65:249–259, 1987.
- [KLL92] M. Katchalski, T. Lewis, and A. Liu. The different ways of stabbing disjoint convex sets. *Discrete Comput. Geom.*, 7:197–206, 1992.
- [KS] V. Koltun and M. Sharir. The partition technique for overlays of envelopes. Unpublished manuscript.
- [KSZ03] M. Katchalski, S. Suri, and Y. Zhou. A constant bound for geometric permutations of disjoint unit balls. *Discrete Comput. Geom.*, 29:161–173, 2003.
- [KV01] M.J. Katz and K.R. Varadarajan. A tight bound on the number of geometric permutations of convex fat objects in \mathbb{R}^d . *Discrete Comput. Geom.*, 26:543–548, 2001.
- [Mat97] J. Matoušek. A Helly-type theorem for unions of convex sets. *Discrete Comput. Geom.*, 18:1–12, 1997.
- [Mat02] J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Math.* Springer-Verlag, New York, 2002.
- [Mor73] H.C. Morris. *Two Pigeon Hole Principles and Unions of Convexly Disjoint Sets*. Ph.D. thesis, California Inst. Tech., Pasadena, CA, 1973.

- [PS89] J. Pach and M. Sharir. The upper envelope of piecewise linear functions and the boundary of a region enclosed by convex plates: combinatorial analysis. *Discrete Comput. Geom.*, 4:291–309, 1989.
- [PS92] M. Pellegrini and P. Shor. Finding stabbing lines in 3-space. *Discrete Comput. Geom.*, 8:191–208, 1992.
- [PW90] R. Pollack and R. Wenger. Necessary and sufficient conditions for hyperplane transversals. *Combinatorica*, 10:307–311, 1990.
- [Rob42] C.V. Robinson. Spherical theorems of Helly type and congruence indices of spherical caps. *Amer. J. Math.*, 64:260–272, 1942.
- [Rob97] J.-M. Robert. Geometric orderings of intersecting translates and their applications. *Comput. Geom. Theory Appl.*, 7:59–72, 1997.
- [SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [San57] R. De Santis. A generalization of Helly's theorem. *Proc. Amer. Math. Soc.*, 8:336–340, 1957.
- [SMS00] S. Smorodinsky, J.S.B. Mitchell, and M. Sharir. Sharp bounds on geometric permutations for pairwise disjoint balls in \mathbb{R}^d . *Discrete Comput. Geom.*, 23:247–259, 2000.
- [SS75] L.G. Sharaburova and Yu.A. Shashkin. Intersections of spherically convex sets. *Math. Notes*, 18:1054–1059, 1975.
- [Tve66] H. Tverberg. A generalization of Radon's theorem. *J. London Math. Soc.*, 41:123–128, 1966.
- [Tve89] H. Tverberg. Proof of Grünbaum's conjecture on common transversals for translates. *Discrete Comput. Geom.*, 4:191–203, 1989.
- [Vin39] P. Vincensini. Sur une extension d'un théorème de M. J. Radon sur les ensembles de corps convexes. *Bull. Soc. Math. France*, 67:115–119, 1939.
- [Wen90a] R. Wenger. Upper bounds on geometric permutations for convex sets. *Discrete Comput. Geom.*, 5:27–33, 1990.
- [Wen90b] R. Wenger. Geometric permutations and connected components. Technical Report TR-90-50, DIMACS, 1990.

5 PSEUDOLINE ARRANGEMENTS

Jacob E. Goodman

INTRODUCTION

Pseudoline arrangements generalize in a natural way arrangements of straight lines, discarding the straightness aspect, but preserving their basic topological and combinatorial properties. Elementary and intuitive in nature, at the same time, by the Folkman-Lawrence topological representation theorem (see [Chapter 6](#)), they provide a concrete geometric model for oriented matroids of rank 3.

After their explicit description by Levi in the 1920's, and the subsequent development of the theory by Ringel in the 1950's, the major impetus was given in the 1970's by Grünbaum's monograph *Arrangements and Spreads*, in which a number of results were collected and a great many problems and conjectures posed about arrangements of both lines and pseudolines. The connection with oriented matroids discovered several years later led to further work. The theory is by now very well developed, with many combinatorial and topological results and relations to other areas, as well as an increasing number of applications in computational geometry.

Section 5.1 is devoted to the basic properties of pseudoline arrangements, and Section 5.2 to related structures, such as arrangements of straight lines, configurations (and generalized configurations) of points, and allowable sequences of permutations. (We do not discuss the connection with oriented matroids, however; that is included in Chapter 6.) In Section 5.3 we discuss the stretchability problem and in Section 5.4 summarize some of the (many) combinatorial results known about line and pseudoline arrangements. Section 5.5 deals with results of a topological nature, Section 5.6 with issues of combinatorial and computational complexity, and Section 5.7 with several applications, including sweeping arrangements and visibility graphs.

Unless otherwise noted, we work in the real projective plane \mathbf{P}^2 .

5.1 BASIC PROPERTIES

GLOSSARY

Arrangement of lines: A labeled set of lines not all passing through the same point (the latter is called a *pencil*).

Pseudoline: A simple closed curve whose removal does not disconnect \mathbf{P}^2 .

Arrangement of pseudolines: A labeled set of pseudolines not a pencil, every pair meeting no more than once (hence exactly once and crossing).

Isomorphic arrangements: Two arrangements such that the mapping induced by their labelings is an isomorphism of the cell complexes into which they parti-

tion \mathbf{P}^2 . (Isomorphism classes of pseudoline arrangements correspond to reorientation classes of oriented matroids of rank 3; see [Chapter 6](#).)

Stretchable: A pseudoline arrangement isomorphic to an arrangement of straight lines. Figure 5.1.1 illustrates what was once believed to be an arrangement of straight lines, but which was later proven not to be stretchable. We will see in [Section 5.6](#) that most pseudoline arrangements, in fact, are not stretchable.

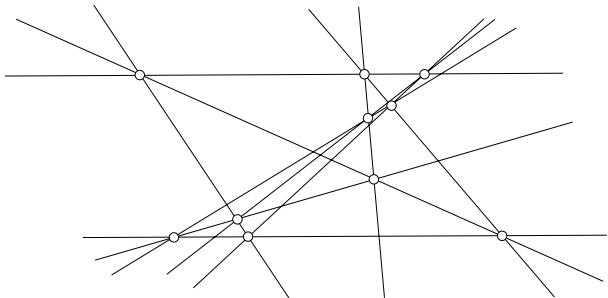


FIGURE 5.1.1

An arrangement of 10 pseudolines,
each containing 3 triple points;
the arrangement is nonstretchable.

Vertex: The intersection of two or more pseudolines in an arrangement.

Ordinary vertex: A vertex at which only two pseudolines meet.

Simple arrangement: An arrangement (of lines or pseudolines) in which each vertex is ordinary.

Euclidean arrangement of pseudolines: An arrangement of x -monotone curves in the Euclidean plane, every pair meeting exactly once and crossing there.

Wiring diagram: A Euclidean arrangement of pseudolines consisting of piecewise linear “wires,” each horizontal except for a short segment where it crosses another wire; see Figure 5.1.2, which shows a wiring diagram labeled $1, \dots, n$ in upward order on the left and in downward order on the right.

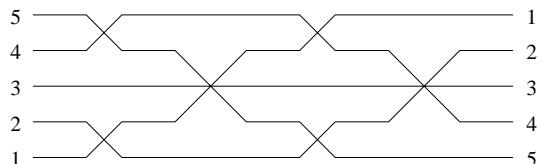


FIGURE 5.1.2

A wiring diagram.

p -convex hull: If \mathcal{A} is an arrangement of pseudolines and p is a point not contained in any member of \mathcal{A} , $L \in \mathcal{A}$ is in the p -convex hull of $\mathcal{B} \subset \mathcal{A}$ if every path from p to a point of L meets some member of \mathcal{B} .

A fundamental tool in working with arrangements of pseudolines, which takes the place of the fact that two points determine a line, is the following.

THEOREM 5.1.1 Levi Enlargement Lemma [Lev26]

If $\mathcal{A} = \{L_1, \dots, L_n\}$ is an arrangement of pseudolines and $p, q \in \mathbf{P}^2$ are two distinct points not on the same member of \mathcal{A} , there is a pseudoline L passing through p and q such that $\mathcal{A} \cup \{L\}$ is an arrangement.

Theorem 5.1.1 has been shown by Goodman and Pollack [GP81b] not to extend to arrangements of pseudohyperplanes. It has, however, been extended by Snoeyink and Hershberger to the case of “2-intersecting curves” (where three points are given) [SH91], and shown by them not to extend to k -intersecting curves and $k + 1$ points for $k > 2$.

The Levi Enlargement Lemma is used to prove extensions to pseudoline arrangements of a number of convexity results on arrangements of straight lines, duals of statements perhaps better known in the setting of configurations of points: Helly’s theorem, Radon’s theorem, Carathéodory’s theorem, Kirchberger’s theorem, the Hahn -Banach theorem, the Krein-Milman theorem, and Tverberg’s generalization of Radon’s theorem (cf. Chapter 4). We state two of these.

THEOREM 5.1.2 *Helly’s Theorem for Pseudoline Arrangements* [GP82a]

If $\mathcal{A}_1, \dots, \mathcal{A}_n$ are subsets of an arrangement \mathcal{A} of pseudolines, and p is a point not on any pseudoline of \mathcal{A} such that, for any i, j, k , \mathcal{A} contains a pseudoline in the p -convex hull of each of $\mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k$, then there is an extension \mathcal{A}' of \mathcal{A} containing a pseudoline lying in the p -convex hull of each of $\mathcal{A}_1, \dots, \mathcal{A}_n$.

THEOREM 5.1.3 *Tverberg’s Theorem for Pseudoline Arrangements* [Rou88b]

If $\mathcal{A} = \{L_1, \dots, L_n\}$ is a pseudoline arrangement with $n \geq 3m - 2$, and p is a point not on any member of \mathcal{A} , then \mathcal{A} can be partitioned into subarrangements $\mathcal{A}_1, \dots, \mathcal{A}_m$ and extended to an arrangement \mathcal{A}' containing a pseudoline lying in the p -convex hull of \mathcal{A}_i for every $i = 1, \dots, m$.

Some of these convexity theorems, but not all, extend to higher dimensional arrangements; see [BLS⁺99, Sections 9.2,10.4], as well as Section 14.3 of this Handbook.

It is not difficult to see that the pseudolines in an arrangement may be drawn as polygonal lines, with bends only at vertices [Grü72]. Related to this is the following representation, which will be discussed further in Section 5.3.

THEOREM 5.1.4 [Goo80]

Every arrangement of pseudolines is isomorphic to a wiring diagram.

Theorem 5.1.4 is used in proving the following duality theorem, which extends to the setting of pseudolines the fundamental duality theorem between lines and points in the projective plane.

THEOREM 5.1.5 [Goo80]

If \mathcal{A} is a pseudoline arrangement and S a point set in \mathbf{P}^2 , and if I is the set of all true statements of the form “ $p \in S$ is incident to $L \in \mathcal{A}$,” then there is a pseudoline arrangement $\hat{\mathcal{A}}$ and a point set \hat{S} such that the set of all incidences holding between members of $\hat{\mathcal{A}}$ and members of $\hat{\mathcal{S}}$ is precisely the dual \hat{I} of I .

THEOREM 5.1.6 [AS02]

For Euclidean arrangements, the result of Theorem 5.1.5 holds with the additional property that the duality preserves above-below relationships as well.

5.2 RELATED STRUCTURES

GLOSSARY

Circular sequence of permutations: A doubly infinite sequence of permutations of $1, \dots, n$ associated with an arrangement \mathcal{A} of lines L_1, \dots, L_n by sweeping a directed line across \mathcal{A} ; see [Figure 5.2.3](#) and the corresponding sequence below.

Local equivalence: Two circular sequences of permutations are locally equivalent if, for each index i , the order in which it switches with the remaining indices is either the same or opposite in the two sequences; see [Figure 5.2.4](#) and Theorem 5.2.2 below.

Local sequence of unordered switches: In a wiring diagram, the permutation α_i given by the order in which the remaining pseudolines cross the i th pseudoline of the arrangement. In [Figure 5.1.2](#), for example, α_2 is $(1, 5, \{3, 4\})$.

Configuration of points: A (labeled) family $\mathcal{S} = \{p_1, \dots, p_n\}$ of points, not all collinear, in \mathbf{P}^2 .

Order type of a configuration \mathcal{S} : The mapping that assigns to each ordered triple i, j, k in $\{1, \dots, n\}$ the orientation of the triple (p_i, p_j, p_k) .

Combinatorial equivalence: Configurations \mathcal{S} and \mathcal{S}' are combinatorially equivalent if the set of permutations of $1, \dots, n$ obtained by projecting \mathcal{S} onto every line in general position agrees with the corresponding set for \mathcal{S}' .

Generalized configuration: A finite set of points in \mathbf{P}^2 , together with a pseudoline joining each pair, the pseudolines forming an arrangement. (Several connecting pseudolines may coincide.) This is sometimes called a **pseudoconfiguration**. An example is shown in [Figure 5.2.1](#).

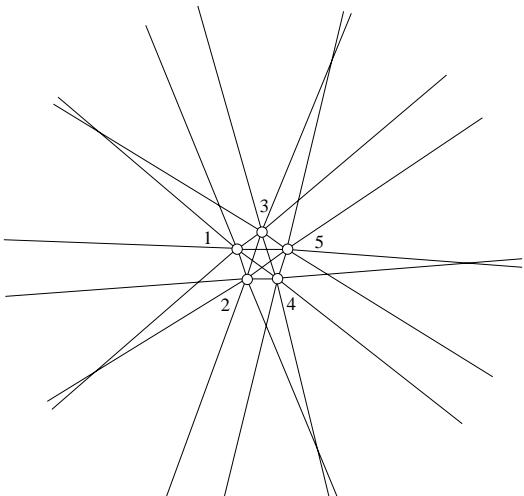


FIGURE 5.2.1
A generalized configuration of 5 points.

Allowable sequence of permutations: A doubly infinite sequence of permutations of $1, \dots, n$ satisfying the three conditions of Theorem 5.2.1. It follows from those conditions that the sequence is periodic of length $\leq n(n - 1)$, and that its period has length $n(n - 1)$ if and only if the sequence is *simple*, i.e., each move consists of the switch of a single pair of indices.

ARRANGEMENTS OF STRAIGHT LINES

Much of the work on pseudoline arrangements has been motivated by problems involving straight-line arrangements. In some cases the question has been whether known results in the case of lines really depended on the *straightness* of the lines; for many (but not all) combinatorial results the answer has turned out to be negative. In other cases, generalization to pseudolines (or, equivalently, reformulations in terms of allowable sequences of permutations—see below) has permitted the solution of a more general problem where none was known previously in the straight case. Finally, pseudolines have turned out to be more useful than lines for certain algorithmic applications; this will be discussed in Section 5.7.

For arrangements of straight lines, there is a rich history of combinatorial results, some of which will be summarized in Section 5.4. Much of this is discussed in [Grü72].

Line arrangements are often classified by isomorphism type. For (unlabeled) arrangements of five lines, for example, Figure 5.2.2 illustrates the four possible isomorphism types, only one of which is simple.

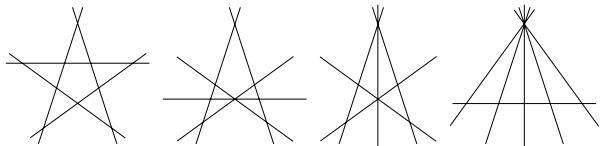


FIGURE 5.2.2
The 4 isomorphism types
of arrangements of 5 lines.

There is a second classification of (numbered) line arrangements, which has proven quite useful for certain problems. If a distinguished point not on any line of the arrangement is chosen to play the part of the “vertical point at infinity,” we can think of the arrangement \mathcal{A} as an arrangement of nonvertical lines in the Euclidean plane, and of P_∞ as the “upward direction.” Rotating a directed line through P_∞ then amounts to sweeping a directed vertical line through \mathcal{A} from left to right (say). We can then note the order in which this directed line cuts the lines of \mathcal{A} , and we arrive at a periodic sequence of permutations of $1, \dots, n$, known as the *circular sequence of permutations* belonging to \mathcal{A} (depending on the choice of P_∞ and the direction of rotation). This sequence is actually doubly infinite, since the rotation of the directed line through P_∞ can be continued in both directions. For the arrangement in Figure 5.2.3, for example, the circular sequence is

$$\mathcal{A} : \dots 12345 \xrightarrow{12,45} 21354 \xrightarrow{135} 25314 \xrightarrow{25,14} 52341 \xrightarrow{234} 54321 \dots$$

Notice how the “moves” between permutations are indicated.

THEOREM 5.2.1 [GP84]

A *circular sequence of permutations* arising from a line arrangement has the following properties:

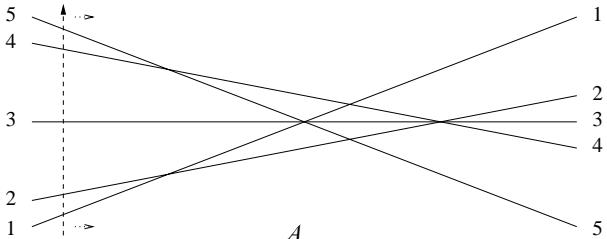


FIGURE 5.2.3

An arrangement of 5 lines.

- (i) *The move from each permutation to the next consists of the reversal of one or more nonoverlapping adjacent substrings;*
- (ii) *After a move in which i and j switch, they do not switch again until every other pair has switched;*
- (iii) *$1, \dots, n$ do not all switch simultaneously with each other.*

If two line arrangements are isomorphic, they may have different circular sequences, depending on the choice of P_∞ (and the direction of rotation). We do have, however:

THEOREM 5.2.2 [GP84]

If \mathcal{A} and \mathcal{A}' are arrangements of lines in \mathbf{P}^2 , and Σ and Σ' are any circular sequences of permutations corresponding to \mathcal{A} and \mathcal{A}' , then \mathcal{A} and \mathcal{A}' are isomorphic if and only if Σ and Σ' are locally equivalent.

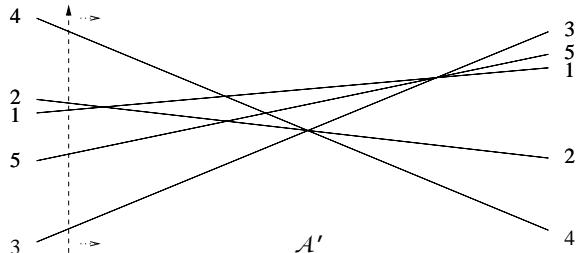


FIGURE 5.2.4

Another arrangement of 5 lines.

Theorem 5.2.2 is illustrated in Figure 5.2.4. Here, the circular sequence of the arrangement \mathcal{A}' , which (as an arrangement in \mathbf{P}^2) is isomorphic to arrangement \mathcal{A} of Figure 5.2.3, is

$$\mathcal{A}' : \dots 35124 \xrightarrow{12} 35214 \xrightarrow{52,14} 32541 \xrightarrow{54} 32451 \xrightarrow{324} 42351 \xrightarrow{351} 42153 \dots$$

Reading off the local sequences of unordered switches of each, we get:

1:	2:	3:	4:	5:
$\mathcal{A} :$... ; 2; 3, 5; 4; ; 1; 5; 3, 4; ; 1, 5; 2, 4; ; 5; 1, 2, 3; ; 4; 1, 3; 2; ...
$\mathcal{A}' :$... ; 2; 4; 3, 5; ; 1; 5; 3, 4; ; 2, 4; 1, 5; ; 1; 5; 2, 3; ; 2; 4; 1, 3; ...

We see that the 2-, 3-, and 5-sequences agree, while the 1- and 4-sequences are reversed.

CONFIGURATIONS OF POINTS

Under projective duality, arrangements of lines in \mathbf{P}^2 correspond to configurations of points. Some questions seem more natural in this setting of points, however, such as the Sylvester-Erdős problem about the existence of an ordinary line in a noncollinear configuration of points, and Scott's conjecture that the minimum number of directions determined by n noncollinear points is $2\lfloor n/2 \rfloor$.

Corresponding to the classification of line arrangements by isomorphism type, it turns out that the “dual” classification of point configurations is by order type.

THEOREM 5.2.3 [GP84]

If \mathcal{A} and \mathcal{A}' are arrangements of lines in \mathbf{P}^2 and \mathcal{S} and \mathcal{S}' the point sets dual to them, then \mathcal{A} and \mathcal{A}' are isomorphic if and only if \mathcal{S} and \mathcal{S}' have the same (or opposite) order types.

From a configuration of points one also derives a circular sequence of permutations in a natural way, by projecting the points onto a rotating line; this gives a finer classification than order type. The sequence for the arrangement in Figure 5.2.3 comes from the configuration in Figure 5.2.5 in this way.

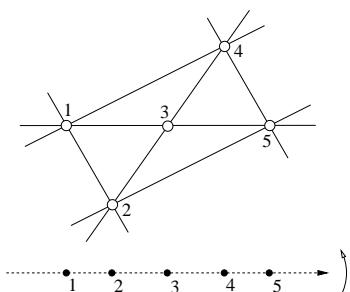


FIGURE 5.2.5
A configuration of 5 points.

In fact, it follows from projective duality that:

THEOREM 5.2.4 [GP82b]

A sequence of permutations is realizable by points if and only if it is realizable by lines.

The circular sequence of a point configuration can be reconstructed from the set of permutations obtained by projecting it onto all lines in general position.

THEOREM 5.2.5 [GP84]

Two configurations have the same circular sequences if and only if they are combinatorially equivalent.

This becomes useful in higher dimensions (where the circular sequence generalizes to a somewhat unwieldy cell decomposition of a sphere with a permutation associated with every cell), since it means that all one really needs to know is the set of permutations; how they fit together can then be determined.

See Chapter 1 of this Handbook for some recent results and some unsolved problems on point configurations.

GENERALIZED CONFIGURATIONS

Just as pseudoline arrangements generalize arrangements of straight lines, generalized configurations provide the corresponding generalization of configurations of points.

The two classifications described above for point configurations, by order type and by circular sequence of permutations, extend in a natural way to generalized configurations. For example, a circular sequence for the generalized configuration in [Figure 5.2.1](#), which is determined by the cyclic order in which the connecting pseudolines meet a distinguished pseudoline (in this case the “pseudoline at infinity”), is

$$\dots 12345^{34}12435^{12}21435^{14}24135^{35}24153^{15}24513^{24}42513^{25}45213^{13}45231^{23}45321^{45}54321 \dots$$

ALLOWABLE SEQUENCES

An allowable sequence of permutations is a combinatorial abstraction of the circular sequence of permutations associated with an arrangement of lines or a configuration of points. We can define, in a natural way, a number of geometric concepts for allowable sequences, such as *collinearity*, *betweenness*, *orientation*, *extreme point*, *convex hull*, *semispace*, *convex n -gon*, *parallel*, etc [GP80a]. Not all allowable sequences are realizable, however, the smallest example being the sequence corresponding to Figure 5.2.1. A realization of this sequence would have to be a drawing of the *bad pentagon* of Figure 5.2.1 with straight lines, and it is not hard to prove that this is impossible.

More generally, we have:

THEOREM 5.2.6 [GP80a]

Suppose Σ is an allowable sequence with extreme points $1, \dots, n$ in counterclockwise order such that, for every i , side $i, i+1$ extended past vertex $i+1$ meets diagonal $i-1, i+2$ extended past vertex $i+2$ (the numbering is modulo n). Then Σ is not realizable by a configuration of points.

Allowable sequences provide a means of rephrasing many geometric problems about point configurations or line arrangements in combinatorial terms. For example, Scott’s conjecture on the minimum number of directions determined by n lines has the simple statement: “Every allowable sequence of permutations of $1, \dots, n$ has at least $2\lfloor n/2 \rfloor$ moves in a half-period.” It was proved in this more general form by Ungar [Ung82], and the proof of the original Scott conjecture follows as a corollary; see also [Jam85], [BLS⁺99, Section 1.11], and [AZ99, Chapter 9].

The Erdős-Szekeres problem (see [Chapter 1](#) of this Handbook) looks as follows in this more general combinatorial formulation:

PROBLEM 5.2.7 Generalized Erdős-Szekeres Problem [GP81a]

What is the minimum n such that for every simple allowable sequence Σ on $1, \dots, n$, there are k indices with the property that each occurs before the other $k-1$ in some term of Σ ?

Allowable sequences arise from pseudoline arrangements by way of wiring diagrams (see Theorem 5.1.4 above), from which they can be read off by sweeping a line across from left to right, just as with an arrangement of straight lines, and they

arise as well from generalized configurations just as from configurations of points. In fact, the following theorem is just a restatement of Theorem 5.1.5.

THEOREM 5.2.8 [GP84]

Every allowable sequence of permutations can be realized both by an arrangement of pseudolines and by a generalized configuration of points.

Allowable sequences have been used to prove the following results, related to the “ k -set” problem (see [Chapter 1](#)):

THEOREM 5.2.9 [Pin03]

Let L be a wiring diagram of size $2n + O(\log \log n)$. Then L has a vertex that is strictly below at least n pseudolines of L and strictly above at least n others.

COROLLARY 5.2.10 [Pin03]

Let L be a wiring diagram of size n . Then L has a vertex P such that the difference between the number of pseudolines strictly above P and the number of those strictly below P is $O(\log \log n)$.

THEOREM 5.2.11 [PP01]

Let L be a simple wiring diagram consisting of n blue and n red pseudolines, and call a vertex P balanced if P is the intersection of a blue and a red pseudoline such that the number of blue pseudolines strictly above P equals the number of red pseudolines strictly above P (and hence the same holds for those strictly below P as well). Then L has at least n balanced vertices, and this result is tight.

WIRING DIAGRAMS

Wiring diagrams provide the simplest “geometric” realizations of allowable sequences. To realize the sequence

$$\mathcal{A} : \dots 12345 \xrightarrow{12,45} 21354 \xrightarrow{135} 25314 \xrightarrow{25,14} 52341 \xrightarrow{234} 54321 \dots,$$

for example, simply start with horizontal “wires” labeled $1, \dots, n$ in (say) increasing order from bottom to top, and, for each move in the sequence, let the corresponding wires cross. This gives the wiring diagram of [Figure 5.1.2](#), and at the end the wires have all reversed order. (It is then easy to extend the curves in both directions to the “line at infinity,” thereby arriving at a pseudoline arrangement in \mathbf{P}^2 .)

We have the following isotopy theorem for wiring diagrams.

THEOREM 5.2.12 [GP85a]

If two wiring diagrams numbered $1, \dots, n$ in order are isomorphic as labeled pseudoline arrangements, then one can be deformed continuously to the other (or to its reflection) through wiring diagrams isomorphic as pseudoline arrangements.

LOCAL SEQUENCES AND CLUSTERS OF STARS

The following theorem (proved independently by Streinu and by Felsner and Weil) solves the “cluster of stars” problem posed in [GP84]; we state it here in terms of local sequences of wiring diagrams, as in [FW01].

THEOREM 5.2.13 [Str97, FW01]

A set $(\alpha_i)_{i=1,\dots,n}$ with each α_i a permutation of $\{1, \dots, i-1, i+1, \dots, n\}$, is the set of local sequences of unordered switches of a simple wiring diagram if and only if for all $i < j < k$ the pairs $\{i, j\}$, $\{i, k\}$, $\{j, k\}$ appear all in natural order or all in inverted order in α_k , α_j , α_i (resp.).

HIGHER DIMENSIONS

Just as isomorphism classes of pseudoline arrangements correspond to oriented matroids of rank 3, the corresponding fact holds for higher-dimensional arrangements, known as arrangements of pseudohyperplanes: they correspond to oriented matroids of rank $d + 1$ (see Theorem 6.2.4 in [Chapter 6](#) of this Handbook).

It turns out, however, that in dimensions > 2 , generalized configurations of points are (surprisingly) more restrictive than such oriented matroids; thus it is only in the plane that “projective duality” works fully in this generalized setting; see [BLS⁺99, Section 5.3].

5.3 STRETCHABILITY

STRETCHABLE AND NONSTRETCHABLE ARRANGEMENTS

Stretchability can be described in either combinatorial or topological terms:

THEOREM 5.3.1 [BLS⁺99, Section 6.3]

Given an arrangement \mathcal{A} or pseudolines in \mathbf{P}^2 , the following are equivalent.

- (i) The cell decomposition induced by \mathcal{A} is combinatorially isomorphic to that induced by some arrangement of straight lines;
- (ii) Some homeomorphism of \mathbf{P}^2 to itself maps every $L_i \in \mathcal{A}$ to a straight line.

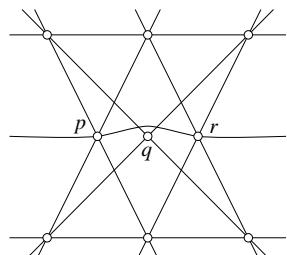


FIGURE 5.3.1

An arrangement that violates the theorem of Pappus.

Among the first examples observed of a nonstretchable arrangement of pseudolines was the non-Pappus arrangement of 9 pseudolines constructed by Levi: see Figure 5.3.1. Since Pappus’s theorem says that points p , q , and r must be collinear if the pseudolines are straight, the arrangement in Figure 5.3.1 is clearly nonstretchable. A second example, involving 10 pseudolines, can be constructed similarly by violating Desargues’s theorem.

Ringel showed how to convert the non-Pappus arrangement into a *simple* arrangement that was still nonstretchable. A symmetric drawing of it is shown in Figure 5.3.2.

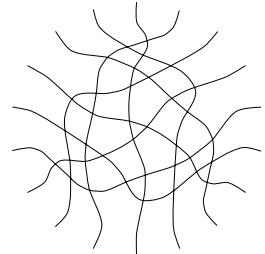


FIGURE 5.3.2
*A simple nonstretchable arrangement
of 9 pseudolines.*

Using allowable sequences, Goodman and Pollack proved the conjecture of Grünbaum that the non-Pappus arrangement has the smallest size possible for a nonstretchable arrangement:

THEOREM 5.3.2 [GP80b]

Every arrangement of 8 or fewer pseudolines is stretchable.

In addition, Richter-Gebert proved that the non-Pappus arrangement is unique among simple arrangements of the same size.

THEOREM 5.3.3 [Ric89]

Every simple arrangement of 9 pseudolines is stretchable, with the exception of the simple non-Pappus arrangement.

The “bad pentagon” of Figure 5.2.1, with extra points inserted to “pin down” the intersections of the sides and corresponding diagonals, provides another example of a nonstretchable arrangement; and Theorem 5.2.6, with extra points, provides, after dualizing, an infinite family of nonstretchable arrangements that were proved, by Bokowski and Sturmfels [BS89a], to be “minor-minimal.” This shows that stretchability of simple arrangements cannot be guaranteed by the exclusion of a finite number of “forbidden” subarrangements. A similar example was found by Haiman and Kahn; see [BLS⁺99, Section 8.3].

As for arrangements of more than 8 pseudolines, we have:

THEOREM 5.3.4 [GPWZ94]

Let \mathcal{A} be an arrangement of n pseudolines. If some face of \mathcal{A} is bounded by at least $n - 1$ pseudolines, then \mathcal{A} is stretchable.

Finally, Shor shows in [Sho91] that even if a stretchable pseudoline arrangement has a symmetry, it may be impossible to realize this symmetry in any stretching.

THEOREM 5.3.5 [Sho91]

There exists a stretchable, simple pseudoline arrangement with a combinatorial symmetry such that no isomorphic arrangement of straight lines has the same combinatorial symmetry.

GENERALIZATIONS OF STRETCHABILITY

While not every pseudoline arrangement is isomorphic to an arrangement of straight lines, every pseudoline arrangement is ***d-stretchable***, i.e., realizable by an arrangement of graphs of polynomial functions of sufficiently high degree d . The following result gives the best bounds known on this degree.

THEOREM 5.3.6 [GP85b]

Let d_n be the smallest number d such that every simple arrangement of n pseudolines is d -stretchable. Then, for appropriate $c_1, c_2 > 0$, we have $c_1\sqrt{n} \leq d_n \leq c_2n^2$.

In several papers [PV94, PV96], Pocchiola and Vegter explore another kind of realizability of pseudoline arrangements, by what they call arrangements of pseudotriangles. A ***pseudotriangle*** is a simply connected, bounded subset T of \mathbb{R}^2 , bounded by 3 convex arcs pairwise tangent at their endpoints, such that T is contained in the triangle formed by these endpoints. The set T^* of directed tangent lines to the boundary of T can be identified by duality with a pseudoline in \mathbf{P}^2 . Because two disjoint pseudotriangles share exactly one common tangent, if $\mathcal{T} = \{T_1, \dots, T_n\}$ is an arrangement of pairwise disjoint pseudotriangles, the curves T_1^*, \dots, T_n^* form an arrangement of pseudolines which is “realized” by the arrangement \mathcal{T} . They prove:

THEOREM 5.3.7 [PV94]

- (i) Every arrangement of straight lines is isomorphic to one realizable by an arrangement of disjoint pseudotriangles.
- (ii) Every arrangement of pseudolines is isomorphic to one realizable by an arrangement of pseudotriangles.

CONJECTURE 5.3.8 [PV94]

Every arrangement of pseudolines is isomorphic to one realizable by disjoint pseudotriangles.

5.4 COMBINATORIAL RESULTS

Although there are exceptions (see below), most combinatorial results known for line arrangements hold for pseudoline arrangements as well. We survey these in this section, including a number of results that update Grünbaum’s comprehensive 1972 survey [Grü72]. For a discussion of *levels in arrangements* (dually, *k-sets*), see [Chapters 24](#) and [1](#), respectively.

GLOSSARY

Simplicial arrangement: An arrangement of lines or pseudolines in which every cell is a triangle.

Near-pencil: An arrangement with all but one line (or pseudoline) concurrent.

Projectively unique: A line arrangement \mathcal{A} with the property that every isomorphic line arrangement is the image of \mathcal{A} under a projective transformation.

x -monotone path: In an arrangement of lines in \mathbb{R}^2 , or in a wiring diagram, a path monotonic in the first coordinate, each step following a line (or wire) from one vertex to another. The *length* of an x -monotone path is one more than the number of turns from one (pseudo)line to another.

SYLVESTER-TYPE RESULTS

CONJECTURE 5.4.1 [Grü69]

Every arrangement of n pseudolines has at least $\lfloor n/2 \rfloor$ ordinary vertices.

The strongest result to date on Conjecture 5.4.1 is the following theorem of Csima and Sawyer (cf. [Chapter 1](#)), which uses previous work of Hansen and improves a long-standing result of Kelly and Moser.

THEOREM 5.4.2 [CS93]

Every arrangement of n pseudolines, with the exception of the one shown in [Figure 1.1.1\(b\)](#), has at least $6n/13$ ordinary vertices.

The arrangement shown in Figure 1.1.1(a) shows that this result is sharp (see Chapter 1 of this Handbook for more details).

Using (complex) algebro-geometric methods, Hirzebruch was able to prove the following result about the number t_i of vertices of multiplicity exactly i in an arrangement of *straight* lines.

THEOREM 5.4.3 [Hir83]

If an arrangement of n lines is not a near-pencil, then

$$t_2 + \frac{3}{4}t_3 \geq n + t_5 + 2t_6 + 3t_7 + \dots$$

RELATIONS AMONG NUMBERS OF VERTICES, EDGES, AND FACES

THEOREM 5.4.4 Euler

If $f_i(\mathcal{A})$ is the number of faces of dimension i in the cell decomposition of \mathbf{P}^2 induced by an arrangement \mathcal{A} , then $f_0(\mathcal{A}) - f_1(\mathcal{A}) + f_2(\mathcal{A}) = 1$.

In addition to **Euler's formula**, the following inequalities are satisfied for arbitrary pseudoline arrangements (here, $n(\mathcal{A})$ is the number of pseudolines in the arrangement \mathcal{A}).

THEOREM 5.4.5 [Grü72, SE88]

- (i) $1 + f_0(\mathcal{A}) \leq f_2(\mathcal{A}) \leq 2f_0(\mathcal{A}) - 2$, with equality on the left for precisely the simple arrangements, and on the right for precisely the simplicial arrangements;
- (ii) $n(\mathcal{A}) \leq f_0(\mathcal{A}) \leq \binom{n(\mathcal{A})}{2}$, with equality on the left for precisely the near-pencils, and on the right for precisely the simple arrangements;
- (iii) For $n \gg 0$, every f_0 satisfying $n^{3/2} \leq f_0 \leq \binom{n}{2}$, with the exceptions of $\binom{n}{2} - 3$ and $\binom{n}{2} - 1$, is the number of vertices of some arrangement of n pseudolines (in fact, of straight lines);

- (iv) $2n(\mathcal{A}) - 2 \leq f_2(\mathcal{A}) \leq \binom{n(\mathcal{A})}{2} + 1$, with equality on the left for precisely the near-pencils, and on the right for precisely the simple arrangements;
- (v) $f_2(\mathcal{A}) \geq 3n(\mathcal{A}) - 6$ if \mathcal{A} is not a near-pencil.

There are gaps in the possible values for $f_2(\mathcal{A})$, as shown by Theorem 5.4.6, which proves a conjecture posed by Grünbaum and generalized by Purdy, refining Theorem 5.4.5(iv).

THEOREM 5.4.6 [Mar93]

There exists an arrangement \mathcal{A} of n pseudolines with $f_2(\mathcal{A}) = f$ if and only if, for some integer k with $1 \leq k \leq n-2$, we have $(n-k)(k+1) + \binom{k}{2} - \min(n-k, \binom{k}{2}) \leq f \leq (n-k)(k+1) + \binom{k}{2}$. Moreover, if \mathcal{A} exists, it can be chosen to consist of straight lines.

Finally, the following result (proved in the more general setting of geometric lattices) gives a complete set of inequalities for the flag vectors $(n(\mathcal{A}), f_0(\mathcal{A}), i(\mathcal{A}))$ of pseudoline arrangements; here $(i(\mathcal{A}))$ is the number of vertex-pseudoline incidences determined by the arrangement \mathcal{A} .

THEOREM 5.4.7 [Nym01]

The closed convex set generated by all the flag vectors of pseudoline arrangements is determined by the inequalities $i \leq 3f_0 - 3$, $i \geq 2f_0$, $f_0 \geq n$, $n \geq 3$, and, for all $k \in \mathbb{Z}_+$, $(k-1)i - kn - (2k-3)f_0 + \binom{k+1}{2} \geq 0$. This set is minimal for $k \geq 3$.

THE NUMBER OF CELLS OF DIFFERENT SIZES

It is easy to see by induction that a *simple* arrangement of more than 3 pseudolines must have at least one nontriangular cell. This observation leads to many questions about numbers of cells of different types in both simple and nonsimple arrangements, some of which have not yet been answered satisfactorily.

The best result on the maximum number of triangles is the following.

THEOREM 5.4.8 [Grü72, Har85, Rou96, FR98]

The maximum number of triangles in an arrangement of $n \geq 9$ pseudolines is bounded above by $\lfloor n(n-1)/3 \rfloor$, with this bound achieved for infinitely many values of n , even for simple straight line arrangements.

For an algorithm to generate all pseudoline arrangements with a maximal number of triangles, and for connections with other combinatorial structures, as well as a generalization of pseudoline arrangements in \mathbb{R}^2 to closed curve arrangements on other surfaces, see [BRS97] and [BP].

PROBLEM 5.4.9 [Grü72]

What is the maximum number of k -sided cells in an arrangement of n pseudolines, for $k > 3$?

On the minimum number of triangles, we have:

THEOREM 5.4.10 [Lev26]

In any arrangement of pseudolines, every pseudoline borders at least 3 triangles. Hence every arrangement of n pseudolines determines at least n triangles.

This minimum is achieved by the “cyclic arrangements” of lines generated by regular polygons, as in Figure 5.4.1.

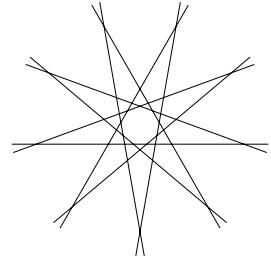


FIGURE 5.4.1
A cyclic arrangement of 9 lines.

For arrangements in the Euclidean plane \mathbb{R}^2 , on the other hand, we have:

THEOREM 5.4.11 [FK99]

- (i) *Every simple arrangement of n pseudolines in \mathbb{R}^2 contains at least $n - 2$ triangles, with equality achieved for all $n \geq 3$.*
- (ii) *Every arrangement of n pseudolines in \mathbb{R}^2 contains at least $2n/3$ triangles, with equality achieved for all $n \equiv 0 \pmod{3}$.*
- (iii) *Every arrangement of n pseudolines in \mathbb{R}^2 contains at most $n(n - 2)/3$ triangles, with equality achieved for infinitely many values of n .*

The following result distinguishes line from pseudoline arrangements.

THEOREM 5.4.12 [Rou88a]

An arrangement of n lines with only n triangles is simple. However, there exist nonsimple arrangements of n pseudolines with only n triangles.

An example of the second assertion of Theorem 5.4.12 is obtained by “collapsing” the central triangle in Figure 5.3.2.

A similar result for quadrilaterals is the following.

THEOREM 5.4.13 [Grü72, Rou87, FR01]

- (i) *Every arrangement of $n \geq 5$ pseudolines contains at most $n(n - 3)/2$ quadrilaterals. For straight-line arrangements, this bound is achieved by a unique simple arrangement for each n .*
- (ii) *A pseudoline arrangement containing $n(n - 3)/2$ quadrilaterals must be simple.*

There are infinitely many simple pseudoline arrangements with no quadrilaterals, contrary to what was once believed. The following result implies, however, that there must be many quadrilaterals or pentagons in *every* simple arrangement.

THEOREM 5.4.14 [Rou87]

Every pseudoline in a simple arrangement of $n > 3$ pseudolines borders at least 3 quadrilaterals or pentagons. Hence, if p_4 is the number of quadrilaterals and p_5 the number of pentagons in a simple arrangement, we must have $4p_4 + 5p_5 \geq 3n$.

The following result was proved after the opposite had been conjectured.

THEOREM 5.4.15 [LRS89]

There is a simple arrangement of straight lines containing no two adjacent triangles.

The proof involved finding a pseudoline arrangement with this property, then showing (algebraically, using Bokowski's "inequality reduction method"—see [Section 5.6](#)) that the arrangement, which consists of 12 pseudolines, is stretchable.

SIMPLICIAL ARRANGEMENTS

In addition to 91 "sporadic" examples of simplicial arrangements of straight lines, the following infinite families are known.

THEOREM 5.4.16 [Grü72]

Each of the following arrangements is simplicial:

- (i) *the near-pencil of n lines;*
- (ii) *the sides of a regular n -gon, together with its n axes of symmetry;*
- (iii) *the arrangement in (ii), together with the line at infinity, for n even.*

On the other hand, additional infinite families of (nonstretchable) simplicial arrangements of pseudolines are known, which are constructible from regular polygons by extending sides, diagonals, and axes of symmetry and modifying the resulting arrangement appropriately. For example, Figure 5.4.2 shows a member of such a family having 31 pseudolines, constructed from a decagon in this way.

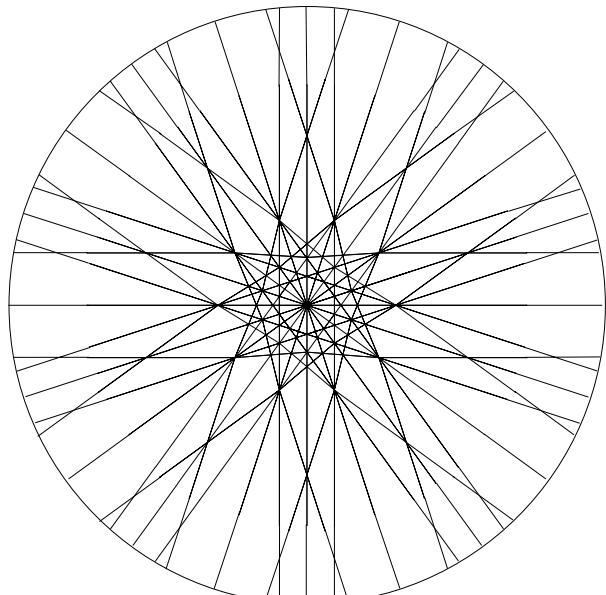


FIGURE 5.4.2
A simplicial arrangement of 31 pseudolines. (The line at infinity, where "parallel" lines meet, is shown as a circle.)

One of the most important problems on arrangements is the following.

PROBLEM 5.4.17 [Grü72]

Classify all simplicial arrangements of pseudolines. Which of these are stretchable? In particular, are there any infinite families of simplicial line arrangements besides the three of Theorem 5.4.16?

It has apparently not been disproved that every (pseudo)line arrangement is a subarrangement of a simplicial (pseudo)line arrangement.

CONJECTURE 5.4.18 [Grü72]

Except for near-pencils, every simplicial arrangement of straight lines is projectively unique.

Finally, putting together results of Strommer and of Csima and Sawyer, we get the following theorem; part (ii) is only a slight improvement over the corresponding result for nonsimplicial arrangements.

THEOREM 5.4.19 [Str77, CS93]

- (i) *For every even n , there is a simplicial arrangement of n lines with a total of $(n^2 + 10n - 8)/8$ cells;*
- (ii) *Except for the arrangement of Figure 1.1.1(b), the number of cells in a simplicial arrangement of n pseudolines is $\leq n(n - 1)/3 + 4 - 4n/13$.*

PATHS IN PSEUDOLINE ARRANGEMENTS

The following result is most easily stated in terms of wiring diagrams.

THEOREM 5.4.20 [Mat91, RT03]

The maximum length of an x -monotone path in a wiring diagram of size n is $\Omega(n^2/\log n)$, and in an arrangement of n lines is $\Omega(n^{7/4})$.

The only upper bound known for the lengths of such paths is the trivial one, $O(n^2)$ (refined to $5n^2/12$ in [RT03]).

For related results on k -levels in arrangements, see [Chapter 24](#).

COMPLEXITY OF SETS OF CELLS IN AN ARRANGEMENT

For cells that “line up” in an arrangement, the best result is:

THEOREM 5.4.21 Zone Theorem [BEPY91]

The sum of the numbers of sides in all the cells of an arrangement of $n + 1$ pseudolines that are supported by one of the pseudolines is $\leq 19n/2 - 1$; this bound is tight.

For general sets of faces, on the other hand, Canham proved:

THEOREM 5.4.22 [Can69]

If F_1, \dots, F_k are any k distinct faces of an arrangement of n pseudolines, then $\sum_{i=1}^k p(F_i) \leq n + 2k(k - 1)$, where $p(F)$ is the number of sides of a face F . This is tight for $2k(k - 1) \leq n$.

For $2k(k - 1) > n$, this was improved by Clarkson et al. to the following result, with simpler proofs later found by Székely and by Dey and Pach; the tightness follows from a result of Szemerédi and Trotter, proved independently by Edelsbrunner and Welzl.

THEOREM 5.4.23 [ST83, EW86, CEG⁺90, Szé97, DP98]

The total number of sides in any k distinct cells of an arrangement of n pseudolines is $O(k^{2/3}n^{2/3} + n)$. This bound is (asymptotically) tight in the worst case.

There are a number of results of this kind for arrangements of objects in the plane and in higher dimensions; see [Chapter 24](#), as well as [CEG⁺90].

SEPARATING POINTS BY LINES AND PSEUDOLINES

Da Silva and Fukuda [DF98] say that a set L of (pseudo)lines *isolates* a set P consisting of n points in the plane if each point of P lies in a distinct cell of L . They give an algorithm to determine the smallest possible size of an isolating set, and prove:

THEOREM 5.4.24 [DF98]

Let $r(P)$ be the largest number of collinear points of P , and $l(P)$ (resp. $l'(P)$) the smallest possible size of an isolating set of lines (resp. pseudolines) for P . If $r(P) > \lceil n/2 \rceil$, then $l(P) = r(P) - 1$. If $r(P) \leq \lceil n/2 \rceil$, then $\max\{\lceil(-1 + \sqrt{8n - 7})/2\rceil, r(P) - 1\} \leq l(P) \leq \lceil n/2 \rceil$. Moreover, $l'(P) = \lceil(-1 + \sqrt{8n - 7})/2\rceil$.

5.5 TOPOLOGICAL PROPERTIES

GLOSSARY

Spread: Given the projective plane \mathbf{P}^2 with a distinguished line L_∞ , a spread of pseudolines is a family $\mathcal{L} = \{L_x\}_{x \in L_\infty}$ of pseudolines varying continuously with $x = L_x \cap L_\infty$, any two of which meet at a single point (at finite distance).

Topological projective plane: \mathbf{P}^2 , with a distinguished family \mathcal{L} of pseudolines (its “lines”), is a topological projective plane if, for each $p, q \in \mathbf{P}^2$, exactly one $L_{p,q} \in \mathcal{L}$ passes through p and q , with $L_{p,q}$ varying continuously with p and q .

(There are other notions of both “spread” and “projective plane” [Grü72], but the ones defined here have the closest connection with pseudoline arrangements.)

Isomorphism of topological projective planes: A homeomorphism that maps “lines” to “lines.”

Universal topological projective plane: One containing an isomorphic copy of every pseudoline arrangement.

Topological sweep: If \mathcal{A} is a pseudoline arrangement in the Euclidean plane and $L \in \mathcal{A}$, a topological sweep of \mathcal{A} “starting at L ” is a continuous family of pseudolines including L , each compatible with \mathcal{A} , which forms a partition of the plane.

Basic semialgebraic set: The set of solutions to a finite number of polynomial equations and strict polynomial inequalities in \mathbb{R}^d . (This term is sometimes used even if the inequalities are not necessarily strict.)

Stable equivalence: A relation on semialgebraic sets that preserves homotopy type. A precise definition appears in [RZ95] and in [Ric96a].

GRAPH-THEORETIC PROPERTIES

THEOREM 5.5.1 [FHNS00]

The graph of a simple projective arrangement of $n \geq 4$ pseudolines is 4-connected.

Using wiring diagrams, the same authors prove:

THEOREM 5.5.2 [FHNS00]

Every projective arrangement with an odd number of pseudolines can be decomposed into two edge-disjoint Hamiltonian paths (plus two unused edges), and the decomposition can be found efficiently.

CONJECTURE 5.5.3 [FHNS00]

All projective arrangements admit decompositions into two Hamiltonian cycles.

EMBEDDING IN LARGER STRUCTURES

In [Grü72], Grünbaum asked a number of questions about extending pseudoline arrangements to more elaborate structures, in particular to spreads and topological planes. The strongest result known about such extendibility is the following, which extends results of Goodman, Pollack, Wenger, and Zamfirescu [GPWZ94].

THEOREM 5.5.4 [GPW96]

There exist uncountably many pairwise nonisomorphic universal topological projective planes.

In particular, this implies the following statements, together with the corresponding statements about spreads, all of which had been conjectured in [Grü72].

- (i) Every pseudoline arrangement can be extended to a topological projective plane.
- (ii) There exists a universal topological projective plane.
- (iii) There are nonisomorphic topological projective planes such that every arrangement in each is isomorphic to some arrangement in the other.

Theorem 5.5.4 also implies the following result, established earlier by Snoeyink and Hershberger (and implicitly by Edmonds, Fukuda, and Mandel—see [BLS⁺99, Section 10.5]).

THEOREM 5.5.5 *Sweeping Theorem* [SH91]

A pseudoline arrangement \mathcal{A} in the Euclidean plane can be swept by a pseudoline, starting at any $L \in \mathcal{A}$.

PROBLEM 5.5.6 [Grü72]

Which arrangements are present (up to isomorphism) in every topological projective plane?

MOVING FROM ONE ARRANGEMENT TO ANOTHER

In [Rin56], Ringel asked whether an arrangement \mathcal{A} of straight lines could always be moved continuously to a given isomorphic arrangement \mathcal{A}' (or to its reflection) so that all intermediate arrangements remained isomorphic. This question, which became known as the “isotopy problem” for arrangements, was eventually solved by Mnëv, and (independently, since news of Mnëv’s results had not yet reached the West) by White in the nonsimple case, then by Jaggi and Mani-Levitska in the simple case [BLS⁺99]. Mnëv’s results are, however, far stronger.

THEOREM 5.5.7 *Mnëv’s Universality Theorem* [Mnë85]

If V is any basic semialgebraic set defined over \mathbb{Q} , there is a configuration \mathcal{S} of points in the plane such that the space of all configurations of the same order type as \mathcal{S} is stably equivalent to V . If V is open in some \mathbb{R}^n , then there is a simple configuration \mathcal{S} with this property.

From this it follows that the space of line arrangements isomorphic to a given one may have the homotopy type of *any* semialgebraic variety, and in particular may be disconnected, which gives a (very strongly) negative answer to the isotopy question. For a further generalization of Theorem 5.5.7, see [Ric96a].

The line arrangement of smallest size known for which the isotopy conjecture fails consists of 14 lines in general position and was found by Suvorov [Suv88]; see also [Ric96b].

Special cases where the isotopy conjecture *does* hold include:

- (i) every arrangement of 9 or fewer lines in general position [Ric89], and
- (ii) an arrangement of n lines containing a cell bounded by at least $n - 1$ of them.

There are also results of a more combinatorial nature about the possibility of transforming one pseudoline arrangement to another. In [Rin56, Rin57], Ringel proved

THEOREM 5.5.8 *Ringel’s Homotopy Theorem*

If \mathcal{A} and \mathcal{A}' are simple arrangements of pseudolines, then \mathcal{A} can be transformed to \mathcal{A}' by a finite sequence of steps each consisting of moving one pseudoline continuously across the intersection of two others. If \mathcal{A} and \mathcal{A}' are simple arrangements of lines, this can be done within the space of line arrangements.

The second part of Theorem 5.5.8 has been generalized by Roudneff and Sturmfels [RS88] to arrangements of planes; the first half is still open in higher dimensions.

Ringel also observed that the isotopy property does hold for pseudoline arrangements.

THEOREM 5.5.9 [Rin56]

If \mathcal{A} and \mathcal{A}' are isomorphic arrangements of pseudolines, then \mathcal{A} can be deformed continuously to \mathcal{A}' through isomorphic arrangements.

Ringel did not provide a proof of this observation, but one method of proving it is via Theorem 5.2.12, together with the following isotopy result.

THEOREM 5.5.10 [GP84]

Every arrangement of pseudolines can be continuously deformed (through isomorphic arrangements) to a wiring diagram.

5.6 COMPLEXITY ISSUES

GLOSSARY

λ -matrix: The matrix with entries $\lambda_{ij} =$ the number of points of the (generalized) configuration $\{p_1, \dots, p_n\}$ to the left of the directed (pseudo)line $\vec{p_i p_j}$. (λ_{ii} is undefined.)

THE NUMBER OF ARRANGEMENTS

Various exact values, as well as bounds, are known for the number of equivalence classes of the structures discussed in this chapter. For low values of n , some of these are given in Table 5.6.1 [Grü72, GP80a, Ric89, Knu92, Fel97, BLS⁺99, AAK01, BKLR, Fin].

TABLE 5.6.1 Exact numbers known for low n .

EQUIVALENCE CLASS	3	4	5	6	7	8	9	10	11	12	13	14	15
Isom classes of arr's of n lines	1	2	4	17	143	4890							
" " " simple " " "	1	1	1	4	11	135	4381	312114	41693377				
" " " simplicial " " "	1	1	1	2	2	2	2	4	2	4	5	5	6
" " " arr's of n pseudolines	1	2	4	17	143	4890	461053	95052532					
" " " simple " " "	1	1	1	4	11	135	4382	312356	41848591				
" " " simplicial " " "	1	1	1	2	2	2							
Isom classes of simple Eucl config's	1	2	3	16	135	3315	158817	14309547	2334512907				
" " " " gen'd config's	1	2	3	16	135	3315	158830	14320182	2343203071				
Comb'l equiv classes of allow seq's	1	2	20										
" " " realizable " "	1	2	19										
Simple allow seq's cont'ng 123...n	2	16	768	...			[see Theorem 5.6.1]						
Simple allow seq's	2	32	4608	...			[see Corollary 5.6.2]						

The only exact formula known for arbitrary n follows from Stanley's formula:

THEOREM 5.6.1 [Sta84]

The number of simple allowable sequences on $1, \dots, n$ containing the permutation $123 \dots n$ is

$$\frac{\binom{n}{2}!}{1^{n-1} 3^{n-2} 5^{n-3} \dots (2n-3)^1}.$$

COROLLARY 5.6.2

The total number of simple allowable sequences on $1, \dots, n$ is

$$\frac{(n-2)! \binom{n}{2}!}{1^{n-1} 3^{n-2} 5^{n-3} \dots (2n-3)^1}.$$

For n arbitrary, Table 5.6.2 indicates the known asymptotic bounds [BLS⁺99, Fel97, GP91, GP93, Knu92].

TABLE 5.6.2 Asymptotic bounds for large n (all logarithms are base 2).

EQUIVALENCE CLASS	LOWER BOUND	UPPER BOUND
Isom classes of (labeled) arr's of n pseudolines	$2^{n^2/6 - 5n/2}$	$2^{1.0850n^2}$
" " " simple " " "	"	$2^{.6974n^2}$
Order types of (labeled) n pt configs (simple or not)	$2^{4n \log n + \Omega(n)}$	$2^{4n \log n + O(n)}$
Isotopy classes of (labeled) n pt configs	"	"
Comb'l equiv classes of (labeled) n pt configs	$2^{7n \log n}$	$2^{8n \log n}$

CONJECTURE 5.6.3 [Knu92]

The number of isomorphism classes of simple pseudoline arrangements is $\leq 2^{\binom{n}{2}}$.

HOW MUCH SPACE IS NEEDED TO SPECIFY AN ARRANGEMENT?

A configuration or generalized configuration \mathcal{S} is described, up to isomorphism, by the set of points lying to the left (say) of each line or pseudoline joining a pair of points. The following theorem, which extends to higher dimensions, allows one to encode the order type of \mathcal{S} in essentially one order of magnitude less space.

THEOREM 5.6.4 [GP83, Cor83]

If \mathcal{S} is a configuration or generalized configuration in the plane, the order type of \mathcal{S} is determined by its λ -matrix.

COROLLARY 5.6.5

The order type of an arrangement of pseudolines can be encoded in space $O(n^2 \log n)$.

A modification by Felsner of the λ -matrix encoding for planar arrangements improves this, giving an encoding of wiring diagrams in space $O(n^2)$:

THEOREM 5.6.6 [Fel97]

Given a wiring diagram $\mathcal{A} = \{L_1, \dots, L_n\}$, let $t_j^i = 1$ if the j th crossing along L_i is with L_k for $k > i$, 0 otherwise. Then the mapping that associates to each wiring diagram \mathcal{A} the binary $n \times (n-1)$ matrix (t_j^i) is injective.

The number of stretchable pseudoline arrangements is much smaller than the total number, which suggests that it should be possible to encode these more compactly. The following result of Goodman, Pollack, and Sturmfels (stated here for the dual case of point configurations) shows, however, that the “naive” encoding, by coordinates of an integral representative, is doomed to be inefficient.

THEOREM 5.6.7 [GPS89]

For each configuration \mathcal{S} of points (x_i, y_i) in the integer grid \mathbb{Z}^2 , let

$$\nu(\mathcal{S}) = \min \max\{|x_1|, \dots, |x_n|, |y_1|, \dots, |y_n|\},$$

the minimum being taken over all configurations \mathcal{S}' of the same order type as \mathcal{S} , and let $\nu^*(n) = \max \nu(\mathcal{S})$ over all n -point configurations. Then, for some $c_1, c_2 > 0$,

$$2^{2^{c_1 n}} \leq \nu^*(n) \leq 2^{2^{c_2 n}}.$$

REALIZABILITY

Along with the Universality Theorem of Section 5.5, Mnëv proved that the problem of determining whether a given pseudoline arrangement is stretchable is NP-hard, in fact as hard as the problem of solving general systems of polynomial equations and inequalities over \mathbb{R} (cf. Chapter 33 of this Handbook):

THEOREM 5.6.8 [Mnë85, Mnë88]

The stretchability problem for pseudoline arrangements is polynomially equivalent to the “existential theory of the reals” decision problem.

Shor [Sho91] presents a more compact proof of the NP-hardness result, by encoding a so-called “monotone 3-SAT” formula in a family of suitably modified Pappus and Desargues configurations that turn out to be stretchable if and only if the corresponding formula is satisfiable. (See also [Ric96a].)

The following result provides an upper bound for the realizability problem.

THEOREM 5.6.9 [BLS⁺99, Sections 8.4,A.5]

The stretchability problem for pseudoline arrangements can be decided in singly exponential time and polynomial space in the Turing machine model of complexity. The number of arithmetic operations needed is bounded above by $2^{4n \log n + O(n)}$.

The NP-hardness does not mean, however, that it is pointless to look for algorithms to determine stretchability, particularly in special cases. Indeed, a good deal of work has been done on this problem by Bokowski, in collaboration with Guedes de Oliveira, Pock, Richter-Gebert, Scharnbacher, and Sturmfels. Four main algorithmic methods have been developed to test for the realizability (or nonrealizability) of an oriented matroid, i.e., in the rank 3 case, the stretchability (respectively nonstretchability) of a pseudoline arrangement:

- (i) The **inequality reduction method**: this attempts to find a relatively small system of inequalities that still carries all the information about a given oriented matroid;
- (ii) The **solvability sequence method**: this attempts to find an elimination order with special properties for the coordinates in a potential realization of an order type;
- (iii) The **final polynomial method**: this attempts to find a bracket polynomial (cf. Chapter 59) whose existence will imply the *nonrealizability* of an order type;
- (iv) Bokowski’s **rubber-band method**: an elementary heuristic that has proven surprisingly effective in finding realizations [Poc91].

Not every realizable order type has a solvability sequence, but it turns out that every nonrealizable one does have a final polynomial, and an algorithm due to Lombardi can be used to find one [Lom90].

All of these methods extend to higher dimensions. For details about the first three, see [BS89b].

CONSTRUCTING ARRANGEMENTS

An $O(n^2)$ algorithm is given in [EOS86, ESS93] to “construct” an arrangement \mathcal{A} of lines (hyperplanes, in general, in time $O(n^d)$), i.e., to construct its face lattice. This algorithm is used as a subroutine in a number of other algorithms in computational geometry (see [Ede87]). From this one can find the λ -matrix of \mathcal{A} in time $O(n^2)$, which is optimal.

SORTING INTERSECTIONS OF LINES OR PSEUDOLINES

Steiger and Streinu consider the problem of x -sorting line or pseudoline intersections, i.e., determining the order of the x -coordinates of the intersections of the lines or pseudolines in a Euclidean arrangement. They prove:

THEOREM 5.6.10 [SS94]

- (i) *There is a decision tree of depth $O(n^2)$ to x -sort the vertices of a simple arrangement of n lines;*
- (ii) *$\Omega(n^2 \log n)$ comparisons are necessary to x -sort the vertices of a simple arrangement of n pseudolines.*

(The second statement is a corollary of Theorem 5.6.1 above.)

Even though this is only a “pseudo-algorithmic” distinction, since it holds in the decision-tree model of computation, nevertheless this result is one of the few known instances where there is a clear computational difference between lines and pseudolines.

5.7 APPLICATIONS

Planar arrangements of lines and pseudolines, as well as point configurations, arise in many problems of computational geometry. Here we describe several such applications involving pseudolines in particular.

GLOSSARY

Tangent visibility graph of a set of pairwise disjoint convex objects: The graph formed by the tangents to pairs of objects, cut off at their points of tangency (provided these segments do not meet any other objects) and by the arcs into which they divide the boundaries of the objects.

Pseudoline graph: Given a Euclidean pseudoline arrangement Γ and a subset E of its vertices, the graph $G = (\Gamma, E)$ whose vertices are the members of Γ , with

two vertices joined by an edge whenever the intersection of the corresponding pseudolines belongs to E .

Extendible set of pseudosegments: A set of Jordan arcs, each chosen from a different pseudoline belonging to a simple Euclidean arrangement.

Diamond: Two pairs $\{l_1, l_2\}, \{l_3, l_4\}$ of pseudolines in a Euclidean arrangement form a diamond if the intersection of one pair lies above each member of the second and the intersection of the other pair below each member of the first.

TOPOLOGICAL SWEEP

The original idea behind what has come to be known as *topologically sweeping an arrangement* was applied, by Edelsbrunner and Guibas, to the case of an arrangement of straight lines. In order to construct the arrangement, rather than using a line to sweep it, they used a pseudoline, and achieved a saving of a factor of $\log n$ in the time required, while keeping the storage linear.

THEOREM 5.7.1 [EG89]

The cell complex of an arrangement of n lines in the plane can be computed in $O(n^2)$ time and $O(n)$ space by sweeping a pseudoline across it.

This result can be applied to a number of problems, and results in an improvement of known bounds on each: minimum area triangle spanned by points, visibility graph of segments, and (in higher dimensions) enumerating faces of a hyperplane arrangement and testing for degeneracies in a point configuration.

The idea of a topological sweep was then generalized, by Snoeyink and Hershberger, to sweeping a pseudoline across an arrangement of *pseudolines*; they prove the possibility of such a sweep (Theorem 5.5.5), and show that it can be performed in the same time and space as in Theorem 5.7.1. They also apply this result to finding a short Boolean formula for a polygon with curved edges.

The topological sweep method was also used by Chazelle and Edelsbrunner [CE92] to report all k -segment intersections in an arrangement of n line segments in (optimal) $O(n \log n + k)$ time, and has been generalized to higher dimensions.

APPLICATIONS OF DUALITY

Theorem 5.1.6, and the algorithm used to compute the dual arrangement, are used by Agarwal and Sharir to compute incidences between points and pseudolines and to compute a subset of faces in a pseudoline arrangement [AS02]. An additional application is due to Sharir and Smorodinsky.

THEOREM 5.7.2 [SS03]

Let Γ be a simple Euclidean pseudoline arrangement, E a subset of vertices of Γ , and $G = (\Gamma, E)$ the corresponding pseudoline graph. Then there is a drawing of G in the plane, with the edges constituting an extendible set of pseudosegments, such that for any two edges e, e' of G , e and e' form a diamond if and only if their corresponding drawings cross.

Conversely, for any graph $G = (V, E)$ drawn in the plane with its edges constituting an extendible set of pseudosegments, there is a simple Euclidean arrangement Γ of pseudolines and a one-to-one mapping ϕ from V onto Γ with each edge $uv \in E$

mapped to the vertex $\phi(u) \cap \phi(v)$ of Γ , such that two edges in E cross if and only if their images are two vertices of Γ forming a diamond.

This can then be used to provide a simple proof of the Tamaki-Tokuyama theorem:

THEOREM 5.7.3 [TT97]

Let Γ and G be as in Theorem 5.7.2. If G is diamond-free, then G is planar, and hence $|E| \leq 3n - 6$.

PSEUDOTRIANGULATIONS

Pocchiola and Vegter introduced the concept of a pseudotriangulation (see [Section 5.3](#) above) in order to compute the visibility graph of a collection of pairwise disjoint convex obstacles. Then they showed that a collection of disjoint pseudotriangles dualizes to a pseudoline arrangement, and that certain pseudoline arrangements could be realized in this way by collections of pseudotriangles. This enables them to generalize certain algorithms for configurations of points to configurations of more general convex objects.

Their results include the following.

THEOREM 5.7.4 [PV94]

Given a collection of n disjoint convex objects in the plane, a pseudotriangulation can be computed in $O(n \log n)$ time, the dual arrangement in $O(n^2)$ time and space, and the tangent visibility graph in $O(n^2)$ time and linear space.

Streinu has modified the notions of pseudotriangle and pseudotriangulation as follows in order to give an algorithmic solution of the **Carpenter's Rule problem** previously settled existentially by Connelly, Demaine, and Rote [CDR03]: A **pseudotriangle** is a planar polygon with precisely three vertices having internal angles less than π , and a **pseudotriangulation** of a point set P in the plane is a partition of the convex hull of P into pseudotriangles whose vertex set is precisely P . She proves:

THEOREM 5.7.5 [Str00]

Every planar polygon can be convexified in $O(n^2)$ motions, each consisting of a one-degree-of-freedom mechanism constructed from a pseudotriangulation with a single convex-hull edge removed, which is moved until two of its adjacent edges align, followed by a local flip of diagonals to restore a pseudotriangulation. A starting pseudotriangulation can be computed in time $O(n^2)$ and subsequently updated in linear time per step.

With the same definitions, Kettner et al. prove:

THEOREM 5.7.6 [KKM⁺03]

Every planar point set in general position has a pseudotriangulation every vertex of which has degree at most 5, and this bound is tight.

If a pseudotriangulation is such that no edge can be removed and leave a pseudotriangulation, it is called *minimal*; in that case it must have exactly $n - 2$ pseudotriangles. Brönnimann et al. have adduced some experimental evidence for:

CONJECTURE 5.7.7 [BKPS01, RRSS01]

For any set S of points in general position in the plane, there are at least as many minimal pseudotriangulations of S as triangulations, with equality if and only if S is in convex position.

PSEUDOVISIBILITY

In a series of papers, O'Rourke and Streinu introduce what they call the “vertex-edge visibility graph” of a polygon, which encodes more information than the standard vertex visibility graph, and use it to study the visibility problem in the polygon. They then generalize this concept to **pseudopolypgons**, whose vertices and edges come from generalized configurations of points (see [Section 5.2](#)), and show that the reconstruction problem for vertex-edge visibility graphs can be solved as long as pseudopolypgons are permitted. They prove:

THEOREM 5.7.8 [OS96]

There is a polynomial-time algorithm for the problem of deciding whether a graph is the vertex-edge pseudovisibility graph of a pseudopolygon.

COROLLARY 5.7.9 [OS96]

The decision problem for vertex visibility graphs of pseudopolypgons is in NP.

(This last result is in contrast to the fact that the same problem with straight-edge visibility is only known to be in PSPACE.)

Finally, Streinu has used Theorem 5.2.6 above to construct examples of non-stretchable pseudopolypgons and of nonstretchable pseudovisibility graphs [Str03].

5.8 SOURCES AND RELATED MATERIAL

FURTHER READING

[BLS⁺99]: A comprehensive account of oriented matroid theory, including a great many references; most references not given explicitly in this chapter can be traced through this book.

[Ede87]: An introduction to computational geometry, focusing on arrangements and their algorithms.

[GP91, GP93]: Two surveys on allowable sequences and order types and their complexity.

[Grü72]: A monograph on planar arrangements and their generalizations, with excellent problems (many still unsolved) and a very complete bibliography up to 1972.

RELATED CHAPTERS

[Chapter 1](#): Finite point configurations

[Chapter 4](#): Helly-type theorems and geometric transversals

- Chapter 6: Oriented matroids
 - Chapter 9: Geometry and topology of polygonal linkages
 - Chapter 24: Arrangements
 - Chapter 33: Computational real algebraic geometry
-

REFERENCES

- [AS02] P.K. Agarwal and M. Sharir. Pseudoline arrangements: duality, algorithms, and applications. *Proc. 13th Annu. ACM-SIAM Sympos. Discr. Algorithms*, 2002, pages 781–790.
- [AAK01] O. Aichholzer, F. Aurenhammer, and H. Krasser. Enumerating order types for small point sets, with applications. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 2001, pages 11–18. See also <http://www.cis.TUGraz.at/igi/oaich/triangulations/ordertypes.html>.
- [AZ99] M. Aigner and G.M. Ziegler. *Proofs from THE BOOK*, 2nd Ed. Springer-Verlag, Heidelberg, 1999.
- [BEPY91] M. Bern, D. Eppstein, P. Plassmann, and F. Yao. Horizon theorems for lines and polygons. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 45–66, volume 6 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.
- [BLS⁺99] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*, 2nd Ed. Volume 46 of *Encyclopedia of Mathematics*. Cambridge University Press, 1999.
- [BKLR] J. Bokowski, U. Kortenkamp, G. Laffaille, and J. Richter-Gebert. Classification of non-stretchable pseudoline arrangements and related properties. In preparation.
- [BP] J. Bokowski and T. Pisanski. Oriented matroids and complete graph embeddings on surfaces. Manuscript.
- [BRS97] J. Bokowski, J.-P. Roudneff, and T.-K. Strempel. Cell decompositions of the projective plane with Petrie polygons of constant length. *Discrete Comput. Geom.*, 17:377–392, 1997.
- [BS89a] J. Bokowski and B. Sturmfels. An infinite family of minor-minimal nonrealizable 3-chirotopes. *Math. Zeitschrift*, 200:583–589, 1989.
- [BS89b] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry*. Volume 1355 of *Lecture Notes in Math.* Springer-Verlag, Heidelberg, 1989.
- [BKPS01] H. Brönnimann, L. Kettner, M. Pocchiola, and J. Snoeyink. Enumerating and counting pseudo-triangulations with the greedy flip algorithm. 2001, manuscript.
- [Can69] R.J. Canham. A theorem on arrangements of lines in the plane. *Israel Math. J.*, 7:393–397, 1969.
- [CE92] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. Assoc. Comput. Mach.*, 39:1–54, 1992.
- [CEG⁺90] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5:99–160, 1990.
- [CDR03] R. Connelly, E.D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete Comput. Geom.*, 30:205–239, 2003.

- [Cor83] R. Cordovil. Oriented matroids and geometric sorting. *Canad. Math. Bull.*, 26:351–354, 1983.
- [CS93] J. Csima and E.T. Sawyer. There exist $6n/13$ ordinary points. *Discrete Comput. Geom.*, 9:187–202, 1993.
- [DF98] I. Da Silva and K. Fukuda. Isolating points by lines in the plane. *J. Geom.*, 62:48–65, 1998.
- [DP98] T. Dey and J. Pach. Extremal problems for geometric hypergraphs. *Discrete Comput. Geom.*, 19:473–484, 1998.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin, 1987.
- [EG89] H. Edelsbrunner and L.J. Guibas. Topologically sweeping an arrangement. *J. Comput. System Sci.*, 38:165–194, 1989; Corrigendum: 42:249–251, 1991.
- [EOS86] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15:341–363, 1986.
- [ESS93] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22:418–429, 1993.
- [EW86] H. Edelsbrunner and E. Welzl. On the maximal number of edges of many faces in arrangements. *J. Combinatorial Th. Ser. A*, 41:159–166, 1986.
- [Fel97] S. Felsner. On the number of arrangements of pseudolines. *Discrete Comput. Geom.*, 18:257–267, 1997.
- [FHNS00] S. Felsner, F. Hurtado, M. Noy, and I. Streinu. Hamiltonicity and colorings of arrangement graphs. *Proc. 11th Annu. ACM-SIAM Sympos. Discrete Algorithms*, 2000, pages 155–164.
- [FK99] S. Felsner and K. Kriegel. Triangles in Euclidean arrangements. *Discrete Comput. Geom.*, 22:429–438, 1999.
- [FW01] S. Felsner and H. Weil. Sweeps, arrangements, and signotopes. *Discrete Appl. Math.*, 109:67–94, 2001.
- [Fin] L. Finschi. Homepage of Oriented Matroids. <http://www.om.math.ethz.ch>.
- [FR98] D. Forge and J.L. Ramírez Alfonsín. Straight line arrangements in the real projective plane. *Discrete Comput. Geom.*, 20:155–161, 1998.
- [FR01] D. Forge and J.L. Ramírez Alfonsín. On counting the k -face cells of cyclic arrangements. *European J. Combin.*, 22:307–312, 2001.
- [Goo80] J.E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Math.*, 32:27–35, 1980.
- [GP80a] J.E. Goodman and R. Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *J. Combin. Theory Ser. A*, 29:220–235, 1980.
- [GP80b] J.E. Goodman and R. Pollack. Proof of Grünbaum's conjecture on the stretchability of certain arrangements of pseudolines. *J. Combinatorial Theory Ser. A*, 29:385–390, 1980.
- [GP81a] J.E. Goodman and R. Pollack. A combinatorial perspective on some problems in geometry. *Congressus Numerantium*, 32:383–394, 1981.
- [GP81b] J.E. Goodman and R. Pollack. Three points do not determine a (pseudo-) plane. *J. Combinatorial Theory Ser. A*, 31:215–218, 1981.
- [GP82a] J.E. Goodman and R. Pollack. Helly-type theorems for pseudoline arrangements in P^2 . *J. Combin. Theory Ser. A*, 32:1–19, 1982.

- [GP82b] J.E. Goodman and R. Pollack. A theorem of ordered duality. *Geom. Dedicata*, 12:63–74, 1982.
- [GP83] J.E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Computing*, 12:484–503, 1983.
- [GP84] J.E. Goodman and R. Pollack. Semispaces of configurations, cell complexes of arrangements. *J. Combin. Theory Ser. A*, 37:257–293, 1984.
- [GP85a] J.E. Goodman and R. Pollack. A combinatorial version of the isotopy conjecture. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, pages 12–19, volume 440 of *Ann. New York Acad. Sci.*, 1985.
- [GP85b] J.E. Goodman and R. Pollack. Polynomial realizations of pseudolines arrangements. *Comm. Pure Applied Math.*, 38:725–732, 1985.
- [GP91] J.E. Goodman and R. Pollack. The complexity of point configurations. *Discrete Appl. Math.*, 31:167–180, 1991.
- [GP93] J.E. Goodman and R. Pollack. Allowable sequences and order types in discrete and computational geometry. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 103–134, volume 10 of *Algorithms Combin.*, Springer-Verlag, Berlin/Heidelberg, 1993.
- [GPS89] J.E. Goodman, R. Pollack, and B. Sturmfels. Coordinate representation of order types requires exponential storage. *Proc. 21st Annu. ACM Sympos. Theory Comput.*, Seattle 1989, 405–410.
- [GPW96] J.E. Goodman, R. Pollack, and R. Wenger. There are uncountably many universal topological planes. *Geom. Dedicata*, 59:157–162, 1996.
- [GPWZ94] J.E. Goodman, R. Pollack, R. Wenger, and T. Zamfirescu. Arrangements and topological planes. *Amer. Math. Monthly*, 101:866–878, 1994.
- [Grü69] B. Grünbaum. The importance of being straight. In *Proc. 12th Biannual Intern. Seminar of the Canadian Math. Congress* (Vancouver, 1969), pages 243–254, 1970.
- [Grü72] B. Grünbaum. *Arrangements and Spreads*. Volume 10 of *CBMS Regional Conf. Ser. in Math.* Amer. Math. Soc., Providence, 1972.
- [GS93] L. Guibas and M. Sharir. Combinatorics and algorithms of arrangements. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 9–36, volume 10 of *Algorithms Combin.* Springer-Verlag, Berlin/Heidelberg, 1993.
- [Har85] H. Harborth. Some simple arrangements of pseudolines with a maximum number of triangles. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, pages 31–33, volume 440 of *Ann. New York Acad. Sci.*, 1985.
- [Hir83] F. Hirzebruch. Arrangements of lines and algebraic surfaces. In M. Artin and J. Tate, editors, *Arithmetic and Geometry*, volume 2, pages 113–140. Birkhäuser, Boston, 1983.
- [Jam85] R.E. Jamison. A survey of the slope problem. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, pages 34–51, volume 440 of *Ann. New York Acad. Sci.*, 1985.
- [KKM⁺03] L. Kettner, D. Kirkpatrick, A. Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi. Tight degree bounds for pseudo-triangulations of points. *Comput. Geom. Theory Appl.*, 25:3–12, 2003.
- [Knu92] D.E. Knuth. *Axioms and Hulls*. Volume 606 of *Lecture Notes in Comput. Sci.* Springer-Verlag, Berlin/Heidelberg, 1992.
- [Lev26] F. Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss.*, 78:256–267, 1926.

- [LRS89] D. Ljubić, J.-P. Roudneff, and B. Sturmfels. Arrangements of lines and pseudolines without adjacent triangles. *J. Combinatorial Theory Ser. A*, 50:24–32, 1989.
- [Lom90] H. Lombardi. Nullstellensatz réel effectif et variantes. *C. R. Acad. Sci. Paris Sér. I*, 310:635–640, 1990.
- [Mar93] N. Martinov. Classification of arrangements by the number of their cells. *Discrete Comput. Geom.*, 9:39–46, 1993.
- [Mat91] J. Matoušek. Lower bounds on the length of monotone paths in arrangements. *Discrete Comput. Geom.*, 6:129–134, 1991.
- [Mnë85] N.E. Mnëv. On manifolds of combinatorial types of projective configurations and convex polyhedra. *Soviet Math. Dokl.*, 32:335–337, 1985.
- [Mnë88] N.E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, pages 527–544, volume 1346 of *Lecture Notes in Math.* Springer-Verlag, Berlin, 1988.
- [Nym01] K. Nyman. *Enumeration in Geometric Lattices and the Symmetric Group*. Ph.D. Thesis, Cornell University, Ithaca, 2001.
- [OS96] J. O'Rourke and I. Streinu. Pseudo-visibility graphs in pseudo-polygons: Part II. Preprint, Smith College, 1996.
- [PP01] J. Pach and R. Pinchasi. On the number of balanced lines. *Discrete Comput. Geom.*, 25:611–628, 2001.
- [Pin03] R. Pinchasi. Lines with many points on both sides. *Discrete Comput. Geom.*, 30:415–435, 2003.
- [PV94] M. Pocchiola and G. Vegter. Order types and visibility types of configurations of disjoint convex plane sets. Extended abstract, Tech. Report 94-4, Labo. d'Inf. de l'ENS, Paris, 1994.
- [PV96] M. Pocchiola and G. Vegter. Pseudo-triangulations: Theory and applications. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, 1996, pages 291–300.
- [Poc91] K.P. Pock. *Entscheidungsmethoden zur Realisierbarkeit orientierter Matroide*. Diplomarbeit, TH Darmstadt, 1991.
- [RT03] R. Radoičić and G. Tóth. Monotone paths in line arrangements. *Comput. Geom. Theory Appl.*, 24:129–134, 2003.
- [RRSS01] D. Randall, G. Rote, F. Santos, and J. Snoeyink. Counting triangulations and pseudotriangulations of wheels. In *Proc. 13th Annu. Canad. Conf. Comput. Geom.*, 2001, pages 149–152.
- [Ric89] J. Richter. Kombinatorische Realisierbarkeitskriterien für orientierte Matroide. *Mitt. Math. Sem. Gießen*, 194:1–112, 1989.
- [Ric96a] J. Richter-Gebert. *Realization Spaces of Polytopes*. Volume 1643 of *Lecture Notes in Math.* Springer-Verlag, Berlin/Heidelberg, 1996.
- [Ric96b] J. Richter-Gebert. Two interesting oriented matroids. *Documenta Math.*, 1:137–148, 1996.
- [RZ95] J. Richter-Gebert and G.M. Ziegler. Realization spaces of 4-polytopes are universal. *Bull. Amer. Math. Soc.*, 95:403–412, 1995.
- [Rin56] G. Ringel. Teilungen der Ebene durch Geraden oder topologische Geraden. *Math. Z.*, 64:79–102, 1956.
- [Rin57] G. Ringel. Über Geraden in allgemeiner Lage. *Elem. Math.*, 12:75–82, 1957.

- [Rou87] J.-P. Roudneff. Quadrilaterals and pentagons in arrangements of lines. *Geom. Dedicata*, 23:221–227, 1987.
- [Rou88a] J.-P. Roudneff. Arrangements of lines with a minimal number of triangles are simple. *Discrete Comput. Geometry*, 3:97–102, 1998.
- [Rou88b] J.-P. Roudneff. Tverberg-type theorems for pseudoconfigurations of points in the plane. *European J. Combin.*, 9:189–198, 1988.
- [Rou96] J.-P. Roudneff. The maximum number of triangles in arrangements of (pseudo-) lines. *J. Combin. Theory Ser. B*, 66:44–74, 1996.
- [RS88] J.-P. Roudneff and B. Sturmfels. Simplicial cells in arrangements and mutations of oriented matroids. *Geom. Dedicata*, 27:153–170, 1988.
- [SE88] P. Salamon and P. Erdős. The solution to a problem of Grünbaum. *Canad. Math. Bull.*, 31:129–138, 1988.
- [SS03] M. Sharir and S. Smorodinsky. Extremal configurations and levels in pseudoline arrangements. In *Proc. Workshop Data Struct. Algor.*, Ottawa, 2003.
- [Sho91] P. Shor. Stretchability of pseudolines is *NP*-hard. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, pages 531–554, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.
- [SH91] J. Snoeyink and J. Hershberger. Sweeping arrangements of curves. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 309–349, volume 6 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.
- [Sta84] R.P. Stanley. On the number of reduced decompositions of elements of Coxeter groups. *European J. Combin.*, 5:359–372, 1984.
- [SS94] W. Steiger and I. Streinu. A pseudo-algorithmic separation of lines from pseudo-lines. *Proc. 6th Annu. Canad. Conf. Comput. Geom.*, 1994, pages 7–11.
- [Str97] I. Streinu. Clusters of stars. *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 1997, pages 439–441.
- [Str03] I. Streinu. Non-stretchable pseudo-visibility graphs. *Comput. Geom. Theory Appl.*, 2003, to appear.
- [Str00] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. *Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci.*, 2000, pages 443–453.
- [Str77] T. Strommer. Triangles in arrangements of lines. *J. Combinatorial Theory Ser. A*, 23:314–320, 1977.
- [Suv88] P. Suvorov. Isotopic but not rigidly isotopic plane systems of straight lines. In *Topology and Geometry — Rohlin Seminar*, O.Ya. Viro, editor, Volume 1346 of *Lecture Notes in Math.* Springer-Verlag, Heidelberg, 1988, pages 545–556.
- [Szé97] L.A. Székely. Crossing numbers and hard Erdős problems in discrete geometry. *Combin. Probab. Comput.*, 6:353–358, 1997.
- [ST83] E. Szemerédi and W.T. Trotter, Jr. Extremal problems in discrete geometry. *Combinatorica*, 3:381–392, 1983.
- [TT97] H. Tamaki and T. Tokuyama. A characterization of planar graphs by pseudo-line arrangements. In *Proc. 8th Annu. Internat. Sympos. Algorithms Comput.* Volume 1350 of *Lecture Notes in Comput. Sci.* Springer-Verlag, Heidelberg, 1997, pages 133–142.
- [Ung82] P. Ungar. $2N$ noncollinear points determine at least $2N$ directions. *J. Combin. Theory Ser. A*, 33:343–347, 1982.

6 ORIENTED MATROIDS

Jürgen Richter-Gebert and Günter M. Ziegler

INTRODUCTION

The theory of *oriented matroids* provides a broad setting in which to model, describe, and analyze combinatorial properties of geometric configurations. Mathematical objects of study that appear to be disjoint and independent, such as *point and vector configurations*, *arrangements of hyperplanes*, *convex polytopes*, *directed graphs*, and *linear programs* find a common generalization in the language of oriented matroids.

The oriented matroid of a finite set of points P extracts relative position and orientation information from the configuration; for example, it can be given by a list of signs that encodes the orientations of all the bases of P . In the passage from a concrete point configuration to its oriented matroid, metrical information is lost, but many structural properties of P have their counterparts at the—purely combinatorial—level of the oriented matroid.

We first introduce oriented matroids in the context of several models and motivations (Section 6.1). Then we present some equivalent axiomatizations (Section 6.2). Finally, we discuss concepts that play central roles in the theory of oriented matroids (Section 6.3), among them *duality*, *realizability*, the study of *simplicial cells*, and the treatment of *convexity*.

6.1 MODELS AND MOTIVATIONS

This section discusses geometric examples that are usually treated on the level of concrete coordinates, but where an “oriented matroid point of view” gives deeper insight. We also present these examples as standard models that provide intuition for the behavior of general oriented matroids.

6.1.1 ORIENTED BASES OF VECTOR CONFIGURATIONS

GLOSSARY

Vector configuration: A matrix $X = (x_1, \dots, x_n) \in (\mathbb{R}^d)^n$, usually assumed to have full rank d .

Matroid of X : The pair $M_X = (E, \mathcal{B}_X)$, where $E := \{1, 2, \dots, n\}$ and \mathcal{B}_X is the set of all (column index d -sets) of bases of X .

Matroid: A pair $M = (E, \mathcal{B})$, where E is a finite set, and $\mathcal{B} \subset 2^E$ is a nonempty

collection of subsets of E (the **bases** of M) that satisfies the **Steinitz exchange axiom**: For all $B_1, B_2 \in \mathcal{B}$ and $e \in B_1 \setminus B_2$, there exists an $f \in B_2 \setminus B_1$ such that $(B_1 \setminus e) \cup f \in \mathcal{B}$.

Signs: Elements of the set $\{-, 0, +\}$, used as a shorthand for the corresponding elements of $\{-1, 0, +1\}$.

Chirotope of X : The map

$$\begin{aligned}\chi_X: \quad E^d &\rightarrow \{-, 0, +\} \\ (\lambda_1, \dots, \lambda_d) &\mapsto \text{sign}(\det(x_{\lambda_1}, \dots, x_{\lambda_d})).\end{aligned}$$

Ordinary (unoriented) *matroids*, as introduced in 1935 by Whitney (see Kung [Kun86], Oxley [Oxl92]), can be considered as an abstraction of vector configurations in finite dimensional vector spaces over arbitrary fields. All the bases of a matroid M have the same cardinality d , which is called the **rank** of the matroid. Equivalently, we can identify M with the characteristic function of the bases $B_M: E^d \rightarrow \{0, 1\}$, where $B_M(\lambda) = 1$ if and only if $\{\lambda_1, \dots, \lambda_d\} \in \mathcal{B}$.

One can obtain examples of matroids as follows: Take a finite set of vectors

$$X = \{x_1, x_2, \dots, x_n\} \subset K^d$$

of rank d in a finite-dimensional vector space K^d and consider the set of bases of K^d formed by subsets of the points in X . In other words, the pair

$$M_X = (E, \mathcal{B}_X) = \left(\{1, \dots, n\}, \{ \{\lambda_1, \dots, \lambda_d\} \mid \det(x_{\lambda_1}, \dots, x_{\lambda_d}) \neq 0 \} \right)$$

forms a matroid.

The basic information about the incidence structure of the points in X is contained in the underlying matroid M_X . However, the matroid alone presents only a weak model of a geometric configuration; for example, all configurations of n points in **general position** in the plane (i.e., no three points on a line) have the same matroid $M = U_{3,n}$: here no information beyond the dimension and size of the configuration, and the fact that it is in general position, is retained for the matroid.

In contrast to matroids, the theory of **oriented matroids** considers the structure of dependencies in vector spaces over *ordered* fields. Roughly speaking, an oriented matroid is a matroid where in addition every basis is equipped with an orientation. These oriented bases have to satisfy an oriented version of the Steinitz exchange axiom (to be described later). In other words, oriented matroids not only describe the incidence structure between the points of X and the hyperplanes spanned by points of X (this is the matroid information); they also encode the positions of the points relative to the hyperplanes: “Which points lie on the positive side of a hyperplane, which points lie on the negative side, and which lie on the hyperplane?” If $X \in (K^d)^n$ is a configuration of n points in a d -dimensional vector space K^d over an ordered field K , we can describe the corresponding oriented matroid χ_X by the function:

$$\begin{aligned}\chi_X: \quad E^d &\rightarrow \{-, 0, +\} \\ (\lambda_1, \dots, \lambda_d) &\mapsto \text{sign}(\det(x_{\lambda_1}, \dots, x_{\lambda_d})).\end{aligned}$$

This map χ_X is called the **chirotope** of X and is very closely related to the oriented matroid of X . It encodes much more information than the corresponding matroid, including orientation and convexity information about the underlying configuration.

6.1.2 CONFIGURATIONS OF POINTS

GLOSSARY

Affine point configuration: A matrix $P = (p_1, \dots, p_n) \in (\mathbb{R}^{d-1})^n$, usually assumed to have full rank $d-1$, i.e., to affinely span \mathbb{R}^{d-1} .

Associated vector configuration: The matrix $X \in (\mathbb{R}^d)^n$ obtained from a point configuration by adding a row of ones. This corresponds to the embedding of the affine space \mathbb{R}^{d-1} into the linear vector space \mathbb{R}^d via $p \mapsto x = \begin{pmatrix} p \\ 1 \end{pmatrix}$.

Oriented matroid of an affine point configuration: The oriented matroid of the associated vector configuration.

Covector of a vector configuration X : Partition of $X = (x_1, \dots, x_n)$ induced by a linear hyperplane, into points on the hyperplane, on its positive side, and on its negative side.

Oriented matroid of X : The collection $\mathcal{L} \subset \{-, 0, +\}^n$ of all covectors of X .

Let $X := (x_1, \dots, x_n) \in (\mathbb{R}^d)^n$ be an $n \times d$ matrix and let $E := \{1, \dots, n\}$. We interpret the columns of X as n vectors in the d -dimensional real vector space \mathbb{R}^d . For a linear functional $y^T \in (\mathbb{R}^d)^*$ we set

$$C_X(y) = (\text{sign}(y^T x_1), \dots, \text{sign}(y^T x_n)).$$

Such a sign vector is called a **covector** of X . We denote the collection of all covectors of X by

$$\mathcal{L}_X := \{C_X(y) \mid y \in \mathbb{R}^d\}.$$

The pair $\mathcal{M}_X = (E, \mathcal{L}_X)$ is called the **oriented matroid** of X . Here each sign vector $C_X(y) \in \mathcal{L}_X$ describes the positions of the vectors x_1, \dots, x_n relative to the linear hyperplane $H_y = \{x \in \mathbb{R}^d \mid y^T x = 0\}$: the sets

$$\begin{aligned} C_X(y)^0 &:= \{e \in E \mid C_X(y)_e = 0\} \\ C_X(y)^+ &:= \{e \in E \mid C_X(y)_e > 0\} \\ C_X(y)^- &:= \{e \in E \mid C_X(y)_e < 0\} \end{aligned}$$

describe how H_y partitions the set of points X . Here $C_X(y)^0$ contains the points on H_y , while $C_X(y)^+$ and $C_X(y)^-$ contain the points on the positive and on the negative side of H_y , respectively. In particular, if $C_X(y)^- = \emptyset$, then all points not on H_y lie on the positive side of H_y . In other words, in this case H_y determines a face of the positive cone

$$\text{pos}(x_1, \dots, x_n) := \left\{ \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n \mid 0 \leq \lambda_i \in \mathbb{R} \text{ for } 1 \leq i \leq n \right\}$$

of all points of X . The face lattice of the cone $\text{pos}(X)$ can be recovered from \mathcal{L}_X . It is simply the set $\mathcal{L}_X \cap \{+, 0\}^E$, partially ordered by the order induced from the relation “ $0 < +$.”

If, in the configuration X , we have $x_{i,d} = 1$ for all $1 \leq i \leq n$, then we can consider X as representing homogeneous coordinates of an *affine* point set X' in \mathbb{R}^{d-1} .

Here the affine points correspond to the original points x_i after removal of the d th coordinate. The face lattice of the convex polytope $\text{conv}(X') \subset \mathbb{R}^{d-1}$ is then identical to the face lattice of $\text{pos}(X)$. Hence, \mathcal{M}_X can be used to recover the *convex hull* of X' .

Thus oriented matroids are generalizations of point configurations in linear or affine spaces. For general oriented matroids we weaken the assumption that the hyperplanes spanned by points of the configuration are flat to the assumption that they only satisfy certain topological incidence properties. Nonetheless, this kind of picture is sometimes misleading since not all oriented matroids have this type of representation (compare the “Type II representations” of [BLS⁺93, Section 5.3]).

6.1.3 ARRANGEMENTS OF HYPERPLANES AND OF HYPERSPHERES

GLOSSARY

Hyperplane arrangement \mathcal{H} : Collection of (oriented) linear hyperplanes in \mathbb{R}^d , given by normal vectors x_1, \dots, x_n .

Hypersphere arrangement induced by \mathcal{H} : Intersection of \mathcal{H} with the unit sphere S^{d-1} .

Covectors of \mathcal{H} : Sign vectors of the cells in \mathcal{H} ; equivalently, $\mathbf{0}$ together with the sign vectors of the cells in $\mathcal{H} \cap S^{d-1}$.

We obtain a different picture if we polarize the situation and consider *hyperplane arrangements* rather than configurations of points. For a real matrix $X := (x_1, \dots, x_n) \in (\mathbb{R}^d)^n$ consider the system of hyperplanes $\mathcal{H}_X := (H_1, \dots, H_n)$ with

$$H_i := \{y \in \mathbb{R}^d \mid y^T x_i = 0\}.$$

Each vector x_i induces an orientation on H_i by defining

$$H_i^+ := \{y \in \mathbb{R}^d \mid y^T x_i > 0\}$$

to be the *positive side* of H_i . We define H_i^- analogously to be the *negative side* of H_i . To avoid degenerate cases we assume that X contains at least one proper basis (i.e., the matrix X has rank d). The hyperplane arrangement \mathcal{H}_X subdivides \mathbb{R}^d into polyhedral cones. Without loss of information we can intersect with the unit sphere S^{d-1} and consider the sphere system

$$\mathcal{S}_X := (H_1 \cap S^{d-1}, \dots, H_n \cap S^{d-1}) = \mathcal{H}_X \cap S^{d-1}.$$

Our assumption that X contains at least one proper basis translates to the fact that the intersection of all $H_1 \cap \dots \cap H_n \cap S^{d-1}$ is empty. \mathcal{H}_X induces a cell decomposition $\Gamma(\mathcal{S}_X)$ on S^{d-1} . Each face of $\Gamma(\mathcal{S}_X)$ corresponds to a sign vector in $\{-, 0, +\}^E$ that indicates the position of the cell with respect to the $(d-2)$ -spheres $H_i \cap S^{d-1}$ (and therefore with respect to the hyperplanes H_i) of the arrangement. The list of all these sign vectors is exactly the set \mathcal{L}_X of covectors of \mathcal{H}_X .

While the visualization of oriented matroids by sets of points in \mathbb{R}^n does not fully generalize to the case of nonrepresentable oriented matroids, the picture of

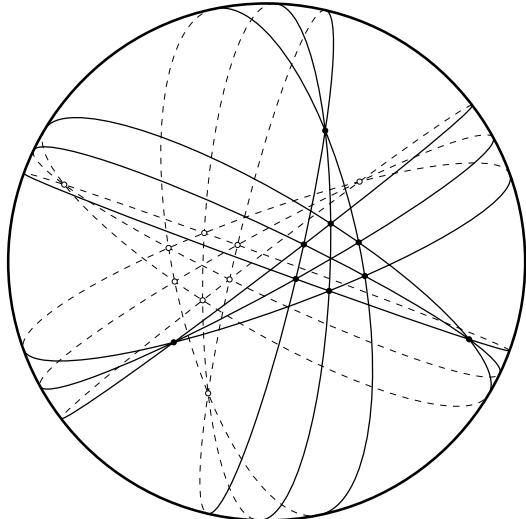


FIGURE 6.1.1

An arrangement of nine great circles on S^2 . The arrangement corresponds to a Pappus configuration.

hyperplane arrangements has a well-defined extension that also covers all the non-realizable cases. We will see that as a consequence of the topological representation theorem of Folkman and Lawrence (Section 6.2.4) every rank- d oriented matroid can be represented as an arrangement of oriented *pseudospheres* (or pseudohyperplanes) embedded in the S^{d-1} (resp. in \mathbb{R}^d). Arrangements of pseudospheres are systems of topological $(d-2)$ -spheres embedded in S^{d-1} that satisfy certain intersection properties that clearly hold in the case of “straight” arrangements.

6.1.4 ARRANGEMENTS OF PSEUDOLINES

GLOSSARY

Pseudoline: Simple closed curve p in the projective plane \mathbb{RP}^2 that is topologically equivalent to a line (i.e., there is a self-homeomorphism of \mathbb{RP}^2 mapping p to a straight line).

Arrangement of pseudolines: Collection of pseudolines $\mathcal{P} := (p_1, \dots, p_n)$ in the projective plane, any two of them intersecting exactly once.

Simple arrangement: No three pseudolines meet in a common point. (Equivalently, the associated oriented matroid is *uniform*.)

Equivalent arrangements: Arrangements \mathcal{P}_1 and \mathcal{P}_2 that generate isomorphic cell decompositions of \mathbb{RP}^2 . (In this case there exists a self-homeomorphism of \mathbb{RP}^2 mapping \mathcal{P}_1 to \mathcal{P}_2 .)

Stretchable arrangement of pseudolines: An arrangement that is equivalent to an arrangement of projective lines.

An *arrangement of pseudolines* in the projective plane is a collection of pseudolines such that any two pseudolines intersect in exactly one point, where they

cross. (See Grünbaum [Grü72] and Richter [Ric89].) We will always assume that \mathcal{P} is *essential*, i.e., that the intersection of all the pseudolines p_i is empty.

An arrangement of pseudolines behaves in many respects just like an arrangement of n lines in the projective plane. (In fact, there are only very few combinatorial theorems known that are true for straight arrangements, but not true in general for pseudoarrangements.) Figure 6.1.2 shows a small example of a nonstretchable arrangement of pseudolines. (It is left as a challenging exercise to the reader to prove the nonstretchability.) Up to isomorphism this is the only simple nonstretchable arrangement of 9 pseudolines [Ric89, Knu92]; every arrangement of 8 (or fewer) pseudolines is stretchable [GP80].

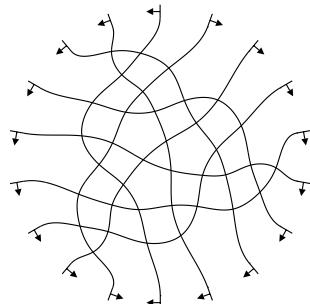


FIGURE 6.1.2

A nonstretchable arrangement of nine pseudolines. It was obtained by Ringel [Rin56] as a perturbation of the Pappus configuration.

To associate with a projective arrangement \mathcal{P} an oriented matroid we represent the projective plane (as customary) by the 2-sphere with antipodal points identified. With this, every arrangement of pseudolines gives rise to an arrangement of *great pseudocircles* on S^2 . For each great pseudocircle on S^2 we choose a positive side. Each cell induced by \mathcal{P} on S^2 now corresponds to a unique sign vector. The collection of all these sign vectors again forms a set of covectors $\mathcal{L}_{\mathcal{P}} \setminus \{\mathbf{0}\}$ of an oriented matroid of rank 3. Conversely, as a special case of the topological representation theorem, *every* oriented matroid of rank 3 has a representation by an *oriented* pseudoline arrangement.

Thus we can use pseudoline arrangements as a standard picture to represent rank-3 oriented matroids. The easiest picture is obtained when we restrict ourselves to the upper hemisphere of S^2 and assume w.l.o.g. that each pseudoline crosses the equator exactly once, and that the crossings are distinct (i.e., no intersection of the great pseudocircles lies on the equator). Then we can represent this upper hemisphere by an arrangement of mutually crossing, oriented affine pseudolines in the plane \mathbb{R}^2 . (We did this implicitly while drawing Figure 6.1.2.) For a recent and reasonably elementary proof of the fact that rank-3 oriented matroids are equivalent to arrangements of pseudolines see Bokowski, Mock, and Streinu [BMS01].

By means of this equivalence, all problems concerning pseudoline arrangements can be translated to the language of oriented matroids. For instance, the problem of stretchability is equivalent to the realizability problem for oriented matroids.

6.2 AXIOMS AND REPRESENTATIONS

In this section we define oriented matroids formally. It is one of the main features of oriented matroid theory that the same object can be viewed under quite dif-

ferent aspects. This results in the fact that there are many different equivalent axiomatizations, and it is sometimes very useful to “jump” from one point of view to another. Statements that are difficult to prove in one language may be easy in another. For this reason we present here several different axiomatizations. We also give a (partial) dictionary that indicates how to translate among them. For a complete version of the basic equivalence proofs—which are highly nontrivial—see [BLS⁺93, Chapters 3 and 5].

We will give axiomatizations of oriented matroids for the following four types of representations:

- Collections of covectors,
- Collections of cocircuits,
- Signed bases,
- Arrangements of pseudospheres.

In the last part of this section these concepts are illustrated by an example.

GLOSSARY

Sign vector: Vector C in $\{-, 0, +\}^E$, where E is a finite index set, usually $\{1, \dots, n\}$. For $e \in E$, the e -component of C is denoted by C_e .

Positive, negative, and zero part of C :

$$\begin{aligned} C^+ &:= \{e \in E \mid C_e = +\}, \\ C^- &:= \{e \in E \mid C_e = -\}, \\ C^0 &:= \{e \in E \mid C_e = 0\}. \end{aligned}$$

Support of C : $\underline{C} := \{e \in E \mid C_e \neq 0\}$.

Zero vector: $\mathbf{0} := (0, \dots, 0) \in \{-, 0, +\}^E$.

Negative of a sign vector: $-C$, defined by $(-C)^+ := C^-$, $(-C)^- := C^+$ and $(-C)^0 = C^0$.

Composition of C and D : $(C \circ D)_e := \begin{cases} C_e & \text{if } C_e \neq 0, \\ D_e & \text{otherwise.} \end{cases}$

Separation set of C and D : $S(C, D) := \{e \in E \mid C_e = -D_e \neq 0\}$.

We partially order the set of sign vectors by “ $0 < +$ ” and “ $0 < -$.” The partial order on sign vectors, denoted by $C \leq D$, is understood componentwise; equivalently, we have

$$C \leq D \iff [C^+ \subset D^+ \text{ and } C^- \subset D^-].$$

For instance, if $C := (+, +, -, 0, -, +, 0, 0)$ and $D := (0, 0, -, +, +, -, 0, -)$, then we have:

$$C^+ = \{1, 2, 6\}, \quad C^- = \{3, 5\}, \quad C^0 = \{4, 7, 8\}, \quad \underline{C} = \{1, 2, 3, 5, 6\},$$

$$C \circ D = (+, +, -, +, -, +, 0, -), \quad C \circ D \geq C, \quad S(C, D) = \{5, 6\}.$$

Furthermore, for $x \in \mathbb{R}^n$, we denote by $\sigma(x) \in \{-, 0, +\}^E$ the image of x under the componentwise sign function σ that maps \mathbb{R}^n to $\{-, 0, +\}^E$.

6.2.1 COVECTOR AXIOMS

Definition: An *oriented matroid* given in terms of its covectors is a pair $\mathcal{M} := (E, \mathcal{L})$, where $\mathcal{L} \in \{-, 0, +\}^E$ satisfies

$$(\text{CV0}) \quad \mathbf{0} \in \mathcal{L}$$

$$(\text{CV1}) \quad C \in \mathcal{L} \implies -C \in \mathcal{L}$$

$$(\text{CV2}) \quad C, D \in \mathcal{L} \implies C \circ D \in \mathcal{L}$$

$$(\text{CV3}) \quad C, D \in \mathcal{L}, e \in S(C, D) \implies$$

there is a $Z \in \mathcal{L}$ with $Z_e = 0$ and with $Z_f = (C \circ D)_f$ for $f \in E \setminus S(C, D)$.

It is not difficult to check that these covector axioms are satisfied by the sign vector system \mathcal{L}_X of the cells in a hyperplane arrangement \mathcal{H}_X , as defined in the last section. The first two axioms are satisfied trivially. For (CV2) assume that x_C and x_D are points in \mathbb{R}^d with $\sigma(x_C^T \cdot X) = C \in \mathcal{L}_X$ and $\sigma(x_D^T \cdot X) = D \in \mathcal{L}_X$. Then (CV2) is implied by the fact that for sufficiently small $\epsilon > 0$ we have $\sigma((x_C + \epsilon x_D)^T \cdot X) = C \circ D$. The geometric content of (CV3) is that if $H_e := \{y \in \mathbb{R}^d \mid y^T x_e = 0\}$ is a hyperplane separating x_C and x_D then there exists a point x_Z on H_e with the property that x_Z is on the same side as x_C and x_D for all hyperplanes not separating x_C and x_D . We can find such a point by intersecting H_e with the line segment that connects x_C and x_D .

As we will see later the partially ordered set (\mathcal{L}, \leq) describes the face lattice of a cell decomposition of the sphere S^{d-1} by pseudohyperspheres. Each sign vector corresponds to a face of the cell decomposition. We define the *rank* d of $\mathcal{M} = (E, \mathcal{L})$ to be the (unique) length of the maximal chains in (\mathcal{L}, \leq) minus one. In the case of realizable arrangements \mathcal{S}_X of hyperspheres, the lattice (\mathcal{L}_X, \leq) equals the face lattice of $\Gamma(\mathcal{S}_X)$.

6.2.2 COCIRCUITS

The covectors of (inclusion-)minimal support in $\mathcal{L} \setminus \{\mathbf{0}\}$ correspond to the 0-faces (= vertices) of the cell decomposition. We call the set $\mathcal{C}^*(\mathcal{M})$ of all such minimal covectors the *cocircuits* of \mathcal{M} . An oriented matroid can be described by its set of cocircuits, as shown by the following theorem.

THEOREM 6.2.1 *Cocircuit Characterization*

A collection $\mathcal{C}^* \in \{-, 0, +\}^E$ is the set of cocircuits of an oriented matroid \mathcal{M} if and only if it satisfies

$$(\text{CC0}) \quad \mathbf{0} \notin \mathcal{C}^*$$

$$(\text{CC1}) \quad C \in \mathcal{C}^* \implies -C \in \mathcal{C}^*$$

$$(\text{CC2}) \quad \text{For all } C, D \in \mathcal{C}^* \text{ we have: } \underline{C} \subset \underline{D} \implies C = D \text{ or } C = -D$$

$$(\text{CC3}) \quad C, D \in \mathcal{C}^*, C \neq -D, \text{ and } e \in S(C, D) \implies$$

there is a $Z \in \mathcal{C}^*$ with $Z^+ \subset (C^+ \cup D^+) \setminus \{e\}$ and $Z^- \subset (C^- \cup D^-) \setminus \{e\}$.

THEOREM 6.2.2 Covector/Cocircuit Translation

For every oriented matroid \mathcal{M} , one can uniquely determine the set \mathcal{C}^* of cocircuits from the set \mathcal{L} of covectors of \mathcal{M} , and conversely, as follows:

- (i) \mathcal{C}^* is the set of vectors with minimal support in $\mathcal{L} \setminus \{\mathbf{0}\}$:

$$\mathcal{C}^* = \{C \in \mathcal{L} \setminus \{\mathbf{0}\} \mid C' \leq C \implies C' \in \{\mathbf{0}, C\}\}$$
- (ii) \mathcal{L} is the set of all sign vectors obtained by successive composition of a finite number of cocircuits from \mathcal{C}^* :

$$\mathcal{L} = \{C_1 \circ \dots \circ C_k \mid k \geq 0, C_1, \dots, C_k \in \mathcal{C}^*\}.$$

6.2.3 CHIROTOPES

GLOSSARY

Alternating sign map: A map $\chi: E^d \rightarrow \{-, 0, +\}$ such that any transposition of two components changes the sign: $\chi(\tau_{ij}(\lambda)) = -\chi(\lambda)$.

Chirotope: An alternating sign map χ that encodes the basis orientations of an oriented matroid \mathcal{M} of rank d .

We now present an axiom system for *chirotopes*, which characterizes oriented matroids in terms of basis orientations. Here an algebraic connection to determinant identities becomes obvious. Chirotopes are the main tool for translating problems in oriented matroid theory to an algebraic setting [BS89a]. They also form a description of oriented matroids that is very practical for many algorithmic purposes (for instance in computational geometry; see Knuth [Knu92]).

Definition: Let $E := \{1, \dots, n\}$ and $0 \leq d \leq n$. A *chirotope of rank d* is an alternating sign map $\chi: E^d \rightarrow \{-, 0, +\}$ that satisfies

(CHI1) The map $|\chi|: E^d \rightarrow \{0, 1\}$, $\lambda \mapsto |\chi(\lambda)|$ is a matroid, and

(CHI2) For every $\lambda \in E^{d-2}$ and $a, b, c, d \in E \setminus \lambda$ the set

$$\left\{ \chi(\lambda, a, b) \cdot \chi(\lambda, c, d), -\chi(\lambda, a, c) \cdot \chi(\lambda, b, d), \chi(\lambda, a, d) \cdot \chi(\lambda, b, c) \right\}$$

either contains $\{-1, +1\}$ or equals $\{0\}$.

Where does the motivation of this axiomatization come from? If we again consider a configuration $X := (x_1, \dots, x_n)$ of vectors in \mathbb{R}^d , we can observe the following identity among the $d \times d$ submatrices of X :

$$\begin{aligned} & \det(x_{\lambda_1}, \dots, x_{\lambda_{d-2}}, x_a, x_b) \cdot \det(x_{\lambda_1}, \dots, x_{\lambda_{d-2}}, x_c, x_d) \\ & - \det(x_{\lambda_1}, \dots, x_{\lambda_{d-2}}, x_a, x_c) \cdot \det(x_{\lambda_1}, \dots, x_{\lambda_{d-2}}, x_b, x_d) \\ & + \det(x_{\lambda_1}, \dots, x_{\lambda_{d-2}}, x_a, x_d) \cdot \det(x_{\lambda_1}, \dots, x_{\lambda_{d-2}}, x_b, x_c) = 0 \end{aligned}$$

for all $\lambda \in E^{d-2}$ and $a, b, c, d \in E \setminus \lambda$. Such a relation is called a *three-term Grassmann-Plücker identity*. If we compare this identity to our axiomatization, we see that (CHI2) implies that

$$\begin{aligned} \chi_X: E^d & \rightarrow \{-, 0, +\} \\ (\lambda_1, \dots, \lambda_d) & \mapsto \text{sign}(\det(x_{\lambda_1}, \dots, x_{\lambda_d})) \end{aligned}$$

is consistent with these identities. More precisely, if we consider χ_X as defined above for a vector configuration X , the above Grassmann-Plücker identities imply that (CHI2) is satisfied. (CHI1) is also satisfied since for the vectors of X the Steinitz exchange axiom holds. (In fact the exchange axiom is a consequence of higher order Grassmann-Plücker identities.)

Consequently, χ_X is a chirotope for every $X \in (\mathbb{R}^d)^n$. Thus chirotopes can be considered as a combinatorial model of the determinant values on vector configurations. The following is not easy to prove, but essential.

THEOREM 6.2.3 *Chirotope/Cocircuit Translation*

For each chirotope χ of rank d on $E := \{1, \dots, n\}$ the set

$$\mathcal{C}^*(\chi) = \left\{ (\chi(\lambda, 1), \chi(\lambda, 2), \dots, \chi(\lambda, n)) \mid \lambda \in E^{d-1} \right\}$$

forms the set of cocircuits of an oriented matroid. Conversely, for every oriented matroid \mathcal{M} with cocircuits \mathcal{C}^* there exists a unique pair of chirotopes $\{\chi, -\chi\}$ such that $\mathcal{C}^*(\chi) = \mathcal{C}^*(-\chi) = \mathcal{C}^*$.

The retranslation of cocircuits into signs of bases is straightforward but needs extra notation. It is omitted here.

6.2.4 ARRANGEMENTS OF PSEUDOSPHERES

GLOSSARY

The $(d-1)$ -sphere: The standard unit sphere $S^{d-1} := \{x \in \mathbb{R}^d \mid \|x\| = 1\}$, or any homeomorphic image of it.

Pseudosphere: The image $s \subset S^{d-1}$ of the equator $\{x \in S^{d-1} \mid x_d = 0\}$ in the unit sphere under a self homeomorphism $\phi: S^{d-1} \rightarrow S^{d-1}$. (This definition describes topologically *tame* embeddings of a $(d-2)$ -sphere in S^{d-1} . Pseudospheres behave “nicely” in the sense that they divide S^{d-1} into two sides homeomorphic to open $(d-1)$ -balls.)

Oriented pseudosphere: A pseudosphere together with a choice of a positive side s^+ and a negative side s^- .

Arrangement of pseudospheres: A set of n pseudospheres in S^{d-1} with the extra condition that any subset of $d+2$ or fewer pseudospheres is *realizable*: it defines a cell decomposition of S^{d-1} that is isomorphic to a decomposition by an arrangement of $d+2$ linear hyperplanes.

Essential arrangement: An arrangement such that the intersection of all the pseudospheres is empty.

Rank: The codimension in S^{d-1} of the intersection of all the pseudospheres. For an essential arrangement in S^{d-1} , the rank is d .

Topological representation of $\mathcal{M} = (E, \mathcal{L})$: An essential arrangement of oriented pseudospheres such that \mathcal{L} is the collection of sign vectors associated with the cells of the arrangement.

One of the most important interpretations of oriented matroids is given by the topological representation theorem of Folkman and Lawrence [FL78]; see also

[BLS⁺93, Chapters 4 and 5] and [BKMS01]. It states that oriented matroids are in bijection to (combinatorial equivalence classes of) *arrangements of oriented pseudospheres*. Arrangements of pseudospheres are a topological generalization of hyperplane arrangements, in the same way in which arrangements of pseudolines generalize line arrangements. Thus every rank- d oriented matroid describes a certain cell decomposition of the $(d-1)$ -sphere. Arrangements of pseudospheres are collections of pseudospheres that have intersection properties just like those satisfied by arrangements of proper subspheres.

Definition: A finite collection $\mathcal{P} = (s_1, s_2, \dots, s_n)$ of pseudospheres in S^{d-1} is an *arrangement of pseudospheres* if the following conditions hold (we set $E := \{1, \dots, n\}$):

(PS1) For all $A \subset E$ the set $S_A = \bigcap_{e \in A} s_e$ is a topological sphere.

(PS2) If $S_A \not\subset s_e$, for $A \subset E, e \in E$, then $S_A \cap s_e$ is a pseudosphere in S_A with sides $S_A \cap s_e^+$ and $S_A \cap s_e^-$.

Notice that this definition permits two pseudospheres of the arrangement to be identical. An entirely different, but equivalent, definition is given in the Glossary.

We see that every essential arrangement of pseudospheres \mathcal{P} partitions the $(d-1)$ -sphere into a regular cell complex $\Gamma(\mathcal{P})$. Each cell of $\Gamma(\mathcal{P})$ is uniquely determined by a sign vector in $\{-, 0, +\}^E$ encoding the relative position with respect to each pseudosphere s_i . Conversely, $\Gamma(\mathcal{P})$ characterizes \mathcal{P} up to homeomorphism. \mathcal{P} is *realizable* if there exists an arrangement of proper spheres \mathcal{S}_X with $\Gamma(\mathcal{P}) \cong \Gamma(\mathcal{S}_X)$.

The translation of arrangements of pseudospheres to oriented matroids is given by the topological representation theorem of Folkman and Lawrence [FL78], as follows. (For the definition of “loop,” see [Section 6.3.1](#).)

THEOREM 6.2.4 *The Topological Representation Theorem (pseudosphere-covector translation)*

If \mathcal{P} is an essential arrangement of pseudospheres on S^{d-1} then $\Gamma(\mathcal{P}) \cup \{\mathbf{0}\}$ forms the set of covectors of an oriented matroid of rank d . Conversely, for every oriented matroid (E, \mathcal{L}) of rank d (without loops) there exists an essential arrangement of pseudospheres \mathcal{P} on S^{d-1} with $\Gamma(\mathcal{P}) = \mathcal{L} \setminus \{\mathbf{0}\}$.

6.2.5 DUALITY

GLOSSARY

Orthogonality: Two sign vectors $C, D \in \{-, 0, +\}^E$ are *orthogonal* if the set

$$\{C_e \cdot D_e \mid e \in E\}$$

either equals $\{0\}$ or contains $\{+, -\}$. We then write $C \perp D$.

Vector of \mathcal{M} : A sign vector that is orthogonal to all covectors of \mathcal{M} ; a covector of the dual oriented matroid \mathcal{M}^* .

Circuit of \mathcal{M} : A vector of minimal nonempty support; a cocircuit of the dual oriented matroid \mathcal{M}^* .

There is a natural duality structure relating oriented matroids of rank d on n elements to oriented matroids of rank $n-d$ on n elements. It is an amazing fact that the existence of such a duality relation can be used to give another axiomatization of oriented matroids (see [BLS⁺93, [Section 3.4](#)]). Here we restrict ourselves to the definition of the dual of an oriented matroid \mathcal{M} .

THEOREM 6.2.5 *Duality*

For every oriented matroid $\mathcal{M} = (E, \mathcal{L})$ of rank d there is a unique oriented matroid $\mathcal{M}^* = (E, \mathcal{L}^*)$ of rank $|E| - d$ given by

$$\mathcal{L}^* = \left\{ D \in \{-, 0, +\}^E \mid C \perp D \text{ for every } C \in \mathcal{L} \right\}.$$

\mathcal{M}^* is called the **dual** of \mathcal{M} . In particular, $(\mathcal{M}^*)^* = \mathcal{M}$.

In particular, the cocircuits of the dual oriented matroid \mathcal{M}^* , which we call the *circuits* of \mathcal{M} , also determine \mathcal{M} . Hence the collection $\mathcal{C}(\mathcal{M})$ of all circuits of an oriented matroid \mathcal{M} , given by

$$\mathcal{C}(\mathcal{M}) := \mathcal{C}^*(\mathcal{M}^*),$$

is characterized by the *the same* cocircuit axioms. Analogously, the *vectors* of \mathcal{M} are obtained as the covectors of \mathcal{M}^* ; they are characterized by the covector axioms.

An oriented matroid \mathcal{M} is realizable if and only if its dual \mathcal{M}^* is realizable. The reason for this is that a matrix $(I_d | A)$ represents \mathcal{M} if and only if $(-A^T | I_{n-d})$ represents \mathcal{M}^* . (Here I_d denotes a $d \times d$ identity matrix, $A \in \mathbb{R}^{d \times (n-d)}$, and $A^T \in \mathbb{R}^{(n-d) \times d}$ denotes the transpose of A .)

Thus for a realizable oriented matroid \mathcal{M}_X the vectors represent the linear dependencies among the columns of X , while the circuits represent minimal linear dependencies. Similarly, in the pseudoarrangements picture, circuits correspond to minimal systems of closed hemispheres that cover the whole sphere, while vectors correspond to consistent unions of such covers that never require the use of both hemispheres determined by a pseudosphere. This provides a direct geometric interpretation of circuits and vectors.

6.2.6 AN EXAMPLE

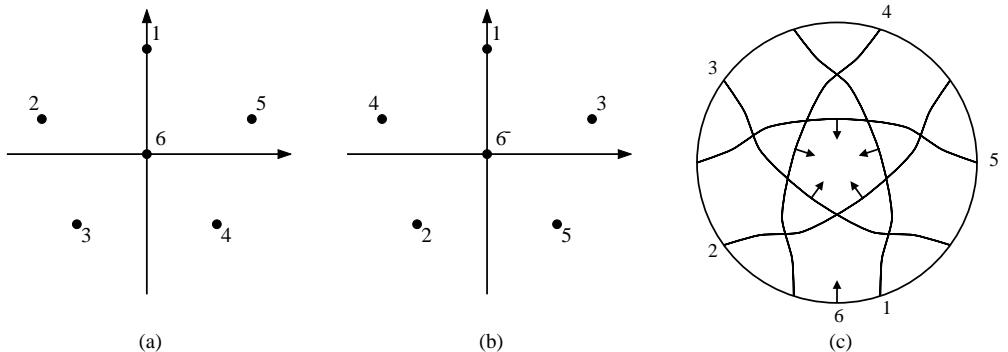
We close this section with an example that demonstrates the different representations of an oriented matroid. Consider the planar point configuration X given in [Figure 6.2.1\(a\)](#).

Homogeneous coordinates for X are given by

$$X := \begin{pmatrix} 0 & 3 & 1 \\ -3 & 1 & 1 \\ -2 & -2 & 1 \\ 2 & -2 & 1 \\ 3 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

FIGURE 6.2.1

An example of an oriented matroid on 6 elements.



The chirotope χ_X of \mathcal{M} is given by the orientations:

$$\begin{array}{lllll} \chi(1, 2, 3) = + & \chi(1, 2, 4) = + & \chi(1, 2, 5) = + & \chi(1, 2, 6) = + & \chi(1, 3, 4) = + \\ \chi(1, 3, 5) = + & \chi(1, 3, 6) = + & \chi(1, 4, 5) = + & \chi(1, 4, 6) = - & \chi(1, 5, 6) = - \\ \chi(2, 3, 4) = + & \chi(2, 3, 5) = + & \chi(2, 3, 6) = + & \chi(2, 4, 5) = + & \chi(2, 4, 6) = + \\ \chi(2, 5, 6) = - & \chi(3, 4, 5) = + & \chi(3, 4, 6) = + & \chi(3, 5, 6) = + & \chi(4, 5, 6) = + \end{array}$$

Half of the cocircuits of \mathcal{M} are given in the table below (the other half is obtained by negating the data):

$$\begin{array}{lll} (0, 0, +, +, +, +) & (0, -, 0, +, +, +) & (0, -, -, 0, +, -) \\ (0, -, -, -, 0, -) & (0, -, -, +, +, 0) & (+, 0, 0, +, +, +) \\ (+, 0, -, 0, +, +) & (+, 0, -, -, 0, -) & (+, 0, -, -, +, 0) \\ (+, +, 0, 0, +, +) & (+, +, 0, -, 0, +) & (+, +, 0, -, -, 0) \\ (+, +, +, 0, 0, +) & (-, +, +, 0, -, 0) & (-, -, +, +, 0, 0) \end{array}$$

Observe that the cocircuits correspond to the point partitions produced by hyperplanes spanned by points. Half of the circuits of \mathcal{M} are given in the next table. The circuits correspond to sign patterns induced by minimal linear dependencies on the rows of the matrix X . It is easy to check that every pair consisting of a circuit and a cocircuit fulfills the orthogonality condition.

$$\begin{array}{lll} (+, -, +, -, 0, 0) & (+, -, +, 0, -, 0) & (+, -, +, 0, 0, -) \\ (+, -, 0, +, -, 0) & (+, +, 0, +, 0, -) & (+, -, 0, 0, -, +) \\ (+, 0, -, +, -, 0) & (+, 0, +, +, 0, -) & (+, 0, +, 0, +, -) \\ (+, 0, 0, +, -, -) & (0, +, -, +, -, 0) & (0, +, -, +, 0, -) \\ (0, +, +, 0, +, -) & (0, +, 0, +, +, -) & (0, 0, +, -, +, -) \end{array}$$

An affine picture of a realization of the dual oriented matroid is given in Figure 6.2.1(b). The minus-sign at point 6 indicates that a reorientation at point 6 has taken place. It is easy to check that the circuits and the cocircuits interchange their roles when dualizing the oriented matroid.

Figure 6.2.1(c) shows the corresponding arrangement of pseudolines. The circle bounding the configuration represents the projective line at infinity representing line 6.

6.3 IMPORTANT CONCEPTS

In this section we briefly introduce some very basic concepts in the theory of oriented matroids. The list of topics treated here is tailored toward some areas of oriented matroid theory that are particularly relevant for applications. Thus many other topics of great importance are left out. In particular, see [BLS⁺93, Section 3.3] for minors of oriented matroids, and [BLS⁺93, Chapter 7] for basic constructions.

6.3.1 SOME BASIC CONCEPTS

In the following glossary, we list some fundamental concepts of oriented matroid theory. Each of them can be expressed in terms of any one of the representations of oriented matroids that we have introduced (covectors, cocircuits, chirotopes, pseudoarrangements), but for each of these concepts some representations are much more convenient than others. Also, each of these concepts has some interesting properties with respect to the duality operator—which may be more or less obvious, depending on the representation that one uses.

GLOSSARY

Direct sum: An oriented matroid $\mathcal{M} = (E, \mathcal{L})$ has a *direct sum decomposition*, denoted by $\mathcal{M} = \mathcal{M}(E_1) \oplus \mathcal{M}(E_2)$, if E has a partition into nonempty subsets E_1 and E_2 such that $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2$ for two oriented matroids $\mathcal{M}_1 = (E_1, \mathcal{L}_1)$ and $\mathcal{M}_2 = (E_2, \mathcal{L}_2)$. If \mathcal{M} has no direct sum decomposition, then it is *irreducible*.

Loops and coloops: A loop of $\mathcal{M} = (E, \mathcal{L})$ is an element $e \in E$ that satisfies $C_e = 0$ for all $C \in \mathcal{L}$. A coloop satisfies $\mathcal{L} \cong \mathcal{L}' \times \{-, 0, +\}$, where \mathcal{L}' is obtained by deleting the e -components from the vectors in \mathcal{L} . If \mathcal{M} has a direct sum decomposition with $E_2 = \{e\}$, then e is either a loop or a coloop.

Acyclic oriented matroid: An oriented matroid $\mathcal{M} = (E, \mathcal{L})$ for which $(+, \dots, +)$ is a covector in \mathcal{L} ; equivalently, the union of the supports of all nonnegative cocircuits is E .

Totally cyclic oriented matroid: An oriented matroid without nonnegative cocircuits; equivalently, $\mathcal{L} \cap \{0, +\}^E = \{\mathbf{0}\}$.

Uniform: An oriented matroid \mathcal{M} of rank d on E is *uniform* if all of its cocircuits have size $|E| - d + 1$. Equivalently, \mathcal{M} is uniform if it has a chirotope with values in $\{+, -\}$.

\mathcal{M} is realizable: There is a vector configuration X with $\mathcal{M}_X = \mathcal{M}$.

Realization of \mathcal{M} : A vector configuration X with $\mathcal{M}_X = \mathcal{M}$.

THEOREM 6.3.1 Duality II

Let \mathcal{M} be an oriented matroid on the ground set E , and \mathcal{M}^* its dual.

- \mathcal{M} is acyclic if and only if \mathcal{M}^* is totally cyclic. (However, “most” oriented matroids are neither acyclic nor totally cyclic!)

- $e \in E$ is a loop of \mathcal{M} if and only if it is a coloop of \mathcal{M}^* .
- \mathcal{M} is uniform if and only if \mathcal{M}^* is uniform.
- \mathcal{M} is a direct sum $\mathcal{M}(E) = \mathcal{M}(E_1) \oplus \mathcal{M}(E_2)$ if and only if \mathcal{M}^* is a direct sum $\mathcal{M}^*(E) = \mathcal{M}^*(E_1) \oplus \mathcal{M}^*(E_2)$.

Duality of oriented matroids captures, among other things, the concepts of linear programming duality [BK92] [BLS⁺93, [Chapter 10](#)] and the concept of Gale diagrams for polytopes [Grü67, Section 5.4] [Zie95, Lecture 6]. For the latter, we note here that the vertex set of a d -dimensional convex polytope P with $d+k$ vertices yields a configuration of $d+k$ vectors in \mathbb{R}^{d+1} , and thus an oriented matroid of rank $d+1$ on $d+k$ points. Its dual is a realizable oriented matroid of rank $k-1$, the **Gale diagram** of P . It can be modeled by an affine point configuration of dimension $k-2$, called an **affine Gale diagram** of P . Hence, for “small” k , we can represent a (possibly high-dimensional) polytope with “few vertices” by a low-dimensional point configuration. In particular, this is beneficial in the case $k=4$, where polytopes with “universal” behavior can be analyzed in terms of their 2-dimensional affine Gale diagrams. For further details, see [Chapter 16](#) of this Handbook.

6.3.2 REALIZABILITY AND REALIZATION SPACES

GLOSSARY

Realization space: Let $\chi: E^d \rightarrow \{-, 0, +\}$ be a chirotope with $\chi(1, \dots, d) = +$.

The realization space $\mathcal{R}(\chi)$ is the set of all matrices $X \in \mathbb{R}^{d \times n}$ with $\chi_X = \chi$ and $x_i = e_i$ for $i = 1, \dots, d$, where e_i is the i th unit vector. If \mathcal{M} is the corresponding oriented matroid, we write $\mathcal{R}(\mathcal{M}) = \mathcal{R}(\chi)$.

Rational realization: A realization $X \in \mathbb{Q}^{d \times n}$; that is, a point in $\mathcal{R}(\chi) \cap \mathbb{Q}^{d \times n}$.

Basic primary semialgebraic set: The (real) solution set of an arbitrary finite system of polynomial equations and strict inequalities with integer coefficients.

Existential theory of the reals: The problem of solving arbitrary systems of polynomial equations and inequalities with integer coefficients.

Stable equivalence: A strong type of arithmetic and homotopy equivalence. Two semialgebraic sets are stably equivalent if they can be connected by a sequence of rational coordinate changes, together with certain projections with contractible fibers. (See [RZ95], and [Ric96a] for details.) In particular, two stably equivalent semialgebraic sets have the same number of components, they are homotopy-equivalent, and either both or neither of them have rational points.

One of the main problems in oriented matroid theory is to design algorithms that find a realization of a given oriented matroid if it exists. However, for oriented matroids with large numbers of points, one cannot be too optimistic, since the realizability problem for oriented matroids is NP-hard. This is one of the consequences of Mnëv’s universality theorem below. An upper bound for the worst-case complexity of the realizability problem is given by the following theorem. It follows

from general complexity bounds for algorithmic problems about semialgebraic sets by Basu, Pollack, and Roy [BPR96] (see also [Chapter 33](#) of this Handbook).

THEOREM 6.3.2 *Complexity of the Best General Algorithm Known*

The realizability of a rank- d oriented matroid on n points can be decided by solving a system of $S = \binom{n}{d}$ real polynomial equations and strict inequalities of degree at most $D = d - 1$ in $K = (n - d - 1)(d - 1)$ variables. Thus, with the algorithms of [BPR96], the number of bit operations needed to decide realizability is (in the Turing machine model of complexity) bounded by $(S/K)^K \cdot S \cdot D^{O(K)}$.

THE UNIVERSALITY THEOREM

A basic observation is that all oriented matroids of rank 2 are realizable. In particular, up to change of orientations and permuting the elements in E there is only one uniform oriented matroid of rank 2. The realization space of an oriented matroid of rank 2 is always stably equivalent to $\{0\}$; in particular, if \mathcal{M} is uniform of rank 2 on n elements, then $\mathcal{R}(\mathcal{M})$ is isomorphic to an open subset of \mathbb{R}^{2n-4} .

In contrast to the rank-2 case, Mnëv's universality theorem states that for oriented matroids of rank 3, the realization space can be “arbitrarily complicated.” Here is the first glimpse of this:

- The realization spaces of all realizable uniform oriented matroids of rank 3 and at most 9 elements are contractible (Richter [Ric89]).
- There is a realizable rank-3 oriented matroid on 9 elements that has no realization with rational coordinates (Perles [Grü67, p. 93]).
- There is a realizable rank-3 oriented matroid on 14 elements with disconnected realization space (Suvorov [Suv88]; see also Richter-Gebert [Ric96b]).

The universality theorem is a fundamental statement with various implications for the configuration spaces of various types of combinatorial objects.

THEOREM 6.3.3 *Mnëv's Universality Theorem* [Mnë88]

For every basic primary semialgebraic set V defined over \mathbb{Z} there is a chirotope χ of rank 3 such that V and $\mathcal{R}(\chi)$ are stably equivalent.

Although some of the facts in the following list were proved earlier than Mnëv's universality theorem, they all can be considered as consequences of the construction techniques used by Mnëv.

CONSEQUENCES OF THE UNIVERSALITY THEOREM

1. The full field of algebraic numbers is needed to realize all oriented matroids of rank 3.
2. The realizability problem for oriented matroids is NP-hard (Mnëv [Mnë88], Shor [Sho91]).
3. The realizability problem for oriented matroids is (polynomial-time-)equivalent to the “Existential Theory of the Reals” (Mnëv [Mnë88]).

4. For every finite simplicial complex Δ , there is an oriented matroid whose realization space is homotopy-equivalent to Δ .
 5. Realizability of rank-3 oriented matroids cannot be characterized by excluding a finite set of “forbidden minors” (Bokowski and Sturmfels [BS89b]).
 6. In order to realize all combinatorial types of integral rank-3 oriented matroids on n elements, even uniform ones, in the integer grid $\{1, 2, \dots, f(n)\}^3$, the “coordinate size” function $f(n)$ has to grow doubly exponentially in n (Goodman, Pollack, and Sturmfels [GPS90]).
 7. The ***isotopy problem*** for oriented matroids (Can one given realization of \mathcal{M} be continuously deformed, through realizations, to another given one?) has a negative solution in general, even for uniform oriented matroids of rank 3 [JMSW89].
-

6.3.3 TRIANGLES AND SIMPLICIAL CELLS

There is a long tradition of studying *triangles* in arrangements of pseudolines. In his 1926 paper [Lev26], Levi already considered them to be important structures. There are good reasons for this. On the one hand, they form the simplest possible cells of full dimension, and are therefore of basic interest. On the other hand, if the arrangement is simple, triangles locate the regions where a “smallest” local change of the combinatorial type of the arrangement is possible. Such a change can be performed by taking one side of the triangle and “pushing” it over the vertex formed by the other two sides. It was observed by Ringel [Rin56] that any two simple arrangements of pseudolines can be deformed into one another by performing a sequence of such “triangle flips.”

Moreover, the realizability of a pseudoline arrangement may depend on the situation at the triangles. For instance, if any one of the triangles in the nonrealizable example of [Figure 6.1.2](#) other than the central one is flipped, the whole configuration becomes realizable.

TRIANGLES IN ARRANGEMENTS OF PSEUDOLINES

Let \mathcal{P} be any arrangement of n pseudolines.

1. For any pseudoline ℓ in \mathcal{P} there are at least 3 triangles adjacent to ℓ . Either the $n - 1$ pseudolines different from ℓ intersect in one point (i.e., \mathcal{P} is a ***near-pencil***), or there are at least $n - 3$ triangles that are not adjacent to ℓ . Thus \mathcal{P} contains at least n triangles (Levi [Lev26]).
2. \mathcal{P} is ***simplicial*** if all its regions are bounded by exactly 3 (pseudo)lines. Except for the near-pencils, there are two infinite classes of simplicial line arrangements and 91 additional “sporadic” simplicial line arrangements (and many more simplicial pseudoarrangements) known (Grünbaum [Grü71]).
3. If \mathcal{P} is simple, then it contains at most $\frac{n(n-1)}{3}$ triangles. For infinitely many values of n , there exists a simple arrangement with $\frac{n(n-1)}{3}$ triangles (Roudneff, Harborth).
4. Any two simple arrangements \mathcal{P}_1 and \mathcal{P}_2 can be deformed into one another by a sequence of simplicial flips (Ringel [Rin56]).

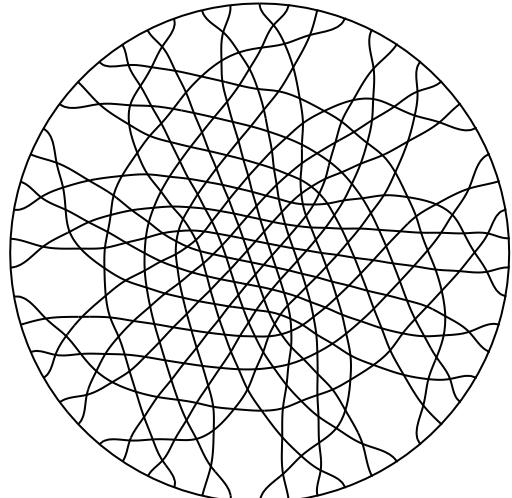


FIGURE 6.3.1
*A simple arrangement of 28 pseudolines
with a maximal number of 252 triangles.*

Every arrangement of pseudospheres in S^{d-1} has a centrally symmetric representation. Thus we can always derive an arrangement of projective pseudohyperplanes (pseudo $(d-2)$ -planes in \mathbb{RP}^{d-1}) by identifying antipodal points. The proper analogue for the triangles in rank 3 are the $(d-1)$ -simplices in projective arrangements of pseudohyperplanes in rank d , i.e., the regions bounded by the minimal number, d , of pseudohyperplanes. We call an arrangement *simple* if no more than $d - 1$ planes meet in a point.

It was conjectured by Las Vergnas in 1980 [Las80] that (as in the rank-3 case) any two simple arrangements can be transformed into each other by a sequence of flips of simplicial regions. In particular this requires that every simple arrangement contain *at least one* simplicial region (which was also conjectured by Las Vergnas). If we consider the case of realizable arrangements only, it is not difficult to prove that any two members in this subclass can be connected by a sequence of flips of simplicial regions and that each realizable arrangement contains at least one simplicial cell. In fact, Shannon [Sha79] proved that every arrangement (even the nonsimple ones) of n projective hyperplanes in rank d contains at least n simplicial regions. More precisely, for every hyperplane h there are at least d simplices adjacent to h and at least $n - d$ simplices not adjacent to h . The contrast between the Las Vergnas conjecture and the results known for the nonrealizable case is dramatic:

SIMPLICIAL CELLS IN PSEUDOARRANGEMENTS

1. There is an arrangement of 8 pseudoplanes in rank 4 having only 7 simplicial regions (Altshuler and Bokowski [ABS80], Roudneff and Sturmefels [RS88]).
2. Every rank-4 arrangement with $n < 13$ pseudoplanes has at least one simplicial region (Bokowski and Rohlfs [BR01]).
3. For every $k > 2$ there is a rank-4 arrangement of $4k$ pseudoplanes having only $3k + 1$ simplicial regions. (This result of Richter-Gebert [Ric93] was improved

by Bokowski and Rohlfs [BR01] to arrangements of $5k$ pseudoplanes with $7k - c$ simplicial regions.)

4. There is a rank-4 arrangement consisting of 20 pseudoplanes for which one plane is not adjacent to any simplicial region (Richter-Gebert [Ric93]; improved to 17 pseudoplanes by Bokowski and Rohlfs [BR01]).
-

OPEN PROBLEMS

The topic of simplicial cells is interesting and rich in structure even in rank 3. The case of higher dimensions is full of unsolved problems and challenging conjectures. These problems are relevant for various problems of great geometric and topological interest, such as the structure of spaces of triangulations. Three key problems are:

1. Classify simplicial arrangements. Is it true, at least, that there are only finitely many types of simplicial arrangements of straight lines outside the three known infinite families?
 2. Does every arrangement of pseudohyperplanes contain at least one simplicial region?
 3. Is it true that any two simple arrangements of pseudospheres can be transformed into one another by a sequence of triangle flips?
-

6.3.4 MATROID POLYTOPES

The convexity properties of a point configuration X are modeled superbly by the oriented matroid \mathcal{M}_X . The combinatorial versions of many theorems concerning convexity also hold on the level of general (including nonrealizable) oriented matroids. For instance, there are purely combinatorial versions of Carathéodory's, Radon's, and Helly's theorems [BLS⁺93, Section 9.2].

In particular, oriented matroid theory provides us with an entirely combinatorial model of convex polytopes, known as “matroid polytopes.” The following definition provides this context in terms of face lattices.

Definition: The face lattice of an acyclic oriented matroid $\mathcal{M} = (E, \mathcal{L})$ is the set

$$\text{FL}(\mathcal{M}) := \{C^0 \mid C \in \mathcal{L} \cap \{0, +\}^E\},$$

partially ordered by inclusion. The elements of $\text{FL}(\mathcal{M})$ are the *faces* of \mathcal{M} . \mathcal{M} is a **matroid polytope** if $\{e\}$ is a face for every $e \in E$.

Every polytope gives rise to a matroid polytope: if $P \subset \mathbb{R}^d$ is a d -polytope with n vertices, then the canonical embedding $x \mapsto \binom{x}{1}$ creates a vector configuration X_P of rank $d+1$ from the vertex set of P . The oriented matroid of X_P is a matroid polytope \mathcal{M}_P , whose face lattice $\text{FL}(\mathcal{M})$ is canonically isomorphic to the face lattice of P .

Matroid polytopes provide a very precise model of (the combinatorial structure of) convex polytopes. In particular, the topological representation theorem implies that *every* matroid polytope of rank d is the face lattice of a regular piecewise linear (PL) cell decomposition of a $(d-2)$ -sphere. Thus matroid polytopes form an

excellent combinatorial model for convex polytopes: in fact, much better than the model of PL spheres (which does not have an entirely combinatorial definition).

However, the construction of a polar fails in general for matroid polytopes. The cellular spheres that represent matroid polytopes have dual cell decompositions (because they are piecewise linear), but this dual cell decomposition is not in general a matroid polytope, even in rank 4 (Billera and Munson [BM84]; Bokowski and Schuchert [BS95]). In other words, the order dual of the face lattice of a matroid polytope (as an abstract lattice) is *not in general* the face lattice of a matroid polytope. (Matroid polytopes form an important tool for polytope theory, not only because of the parts of polytope theory that work for them, but also because of those that fail.)

For every matroid polytope one has the dual oriented matroid (which is totally cyclic, hence not a matroid polytope). In particular, the setup for Gale diagrams generalizes to the framework of matroid polytopes; this makes it possible to also include nonpolytopal spheres in a discussion of the realizability properties of polytopes. This amounts to perhaps the most powerful single tool ever developed for polytope theory. It leads to, among other things, the classification of d -dimensional polytopes with at most $d + 3$ vertices, the proof that all matroid polytopes of rank $d + 1$ with at most $d + 3$ vertices are realizable, the construction of nonrational polytopes as well as of nonpolytopal spheres with $d + 4$ vertices, etc.

ALGORITHMIC APPROACH TO POLYTOPE CLASSIFICATION

A powerful approach, via matroid polytopes, to the problem of classifying all convex polytopes with given parameters is largely due to Bokowski and Sturmfels [BS89a]. Here we restrict our attention to the simplicial case—there are additional technical problems to deal with in the nonsimplicial case, and very little work has been done there as yet. However, the program has been successfully completed for the classification of all simplicial 3-spheres with 9 vertices (Altshuler, Bokowski, and Steinberg [ABS80]) and of all neighborly 5-spheres with 10 vertices (Bokowski and Shemer [BS87]) into polytopes and nonpolytopes. At the core of the matroidal approach lies the following hierarchy:

$$\left(\begin{array}{c} \text{simplicial} \\ \text{spheres} \end{array} \right) \supset \left(\begin{array}{c} \text{uniform} \\ \text{matroid polytopes} \end{array} \right) \supset \left(\begin{array}{c} \text{convex} \\ \text{polytopes} \end{array} \right).$$

The plan of attack is the following. First, one enumerates all isomorphism types of simplicial spheres with given parameters. Then, for each sphere, one computes *all* (uniform) matroid polytopes that have the given sphere as their face lattices. Finally, for each matroid polytope, one tries to decide realizability.

At both of the steps of this hierarchy there are considerable subtleties involved that lead to important insights. For a given simplicial sphere, there may be

- *no* matroid polytope that supports it. In this case the sphere is called ***non-matroidal***. The Barnette sphere [BLS⁺93, Proposition 9.5.3] is an example.
- *exactly one* matroid polytope. In this (important) case the sphere is called ***rigid***. That is, a matroid polytope \mathcal{M} is rigid if $\text{FL}(\mathcal{M}') = \text{FL}(\mathcal{M})$ already implies $\mathcal{M}' = \mathcal{M}$. For rigid matroid polytopes the face lattice uniquely defines the oriented matroid, and thus every statement about the matroid polytope yields a statement about the sphere. In particular, the matroid polytope and the sphere have the same realization space.

Rigid matroid polytopes are a priori rare; however, the *Lawrence construction* [BLS⁺93, Section 9.3] [Zie95, Section 6.6] associates with every oriented matroid \mathcal{M} on n elements in rank d a rigid matroid polytope $\Lambda(\mathcal{M})$ with $2n$ vertices of rank $n + d$. The realizations of $\Lambda(\mathcal{M})$ can be retranslated into realizations of \mathcal{M} .

- or *many* matroid polytopes.

The situation is similarly complex for the second step, from matroid polytopes to convex polytopes. In fact, for each matroid polytope there may be

- *no* convex polytope—this is the case for a nonrealizable matroid polytope. These exist already with relatively few vertices; namely in rank 5 with 9 vertices [BS95], and in rank 4 with 10 vertices [BLS⁺93, Proposition 9.4.5].
- essentially *only one*—this is the rare case where the matroid polytope is “projectively unique.”
- or *many* convex polytopes—the space of all polytopes for a given matroid polytope is the realization space of the oriented matroid, and this may be arbitrarily complicated. In fact, a combination of Mnëv’s universality theorem, the Lawrence construction, and a scattering technique [BS89a, Theorem 6.2] (in order to handle the simplicial case) yields the following amazing universality theorem.

THEOREM 6.3.4 *Mnëv’s Universality Theorem for Polytopes* [Mnë88]

For every [open] basic primary semialgebraic set V defined over \mathbb{Z} there is an integer d and a [simplicial] d -dimensional polytope P on $d + 4$ vertices such that V and the realization space of P are stably equivalent.

6.4 SOURCES AND RELATED MATERIAL

FURTHER READING

The basic theory of oriented matroids was introduced in two fundamental papers, Bland and Las Vergnas [BL78] and Folkman and Lawrence [FL78]. We refer to the monograph by Björner, Las Vergnas, Sturmfels, White, and Ziegler [BLS⁺93] for a broad introduction, and for an extensive development of the theory of oriented matroids. Other introductions and basic sources of information include Bachem and Kern [BK92], Bokowski [Bok93], Bokowski and Sturmfels [BS89a], and Ziegler [Zie95, Lectures 6 and 7].

RELATED CHAPTERS

- [Chapter 5: Pseudoline arrangements](#)
- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 24: Arrangements](#)
- [Chapter 33: Computational real algebraic geometry](#)
- [Chapter 46: Mathematical programming](#)
- [Chapter 59: Geometric applications of the Grassmann-Cayley algebra](#)

REFERENCES

- [ABS80] A. Altshuler, J. Bokowski, and L. Steinberg. The classification of simplicial 3-spheres with nine vertices into polytopes and non-polytopes. *Discrete Math.*, 31:115–124, 1980.
- [BK92] A. Bachem and W. Kern. *Linear Programming Duality: An Introduction to Oriented Matroids*. Universitext. Springer-Verlag, Berlin, 1992.
- [BPR96] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. Assoc. Comput. Mach.*, 43:1002–1045, 1996.
- [BM84] L.J. Billera and B.S. Munson. Polarity and inner products in oriented matroids. *European J. Combin.*, 5:293–308, 1984.
- [BLS⁺93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*. Volume 46 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993; second ed. 1999.
- [BL78] R.G. Bland and M. Las Vergnas. Orientability of matroids. *J. Combin. Theory Ser. B*, 24:94–123, 1978.
- [Bok93] J. Bokowski. Oriented matroids. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 555–602. North-Holland, Amsterdam, 1993.
- [BMS01] J. Bokowski, S. Mock, and I. Streinu. On the Folkman-Lawrence topological representation theorem for oriented matroids of rank 3, *European J. Combin.*, 22:601–615, 2001.
- [BR01] J. Bokowski and H. Rohlfs. On a mutation problem of oriented matroids. *European J. Combin.*, 22:617–626, 2001.
- [BKMS01] J. Bokowski, S. King, S. Mock, and I. Streinu. A topological representation theorem for oriented matroids. Preprint, 21 pages, 2001; [arXiv:math.C0/0209364](https://arxiv.org/abs/math/0209364).
- [BS95] J. Bokowski and P. Schuchert. Altshuler's sphere M_{963}^9 revisited. *SIAM J. Discrete Math.*, 8:670–677, 1995.
- [BS87] J. Bokowski and I. Shemer. Neighborly 6-polytopes with 10 vertices. *Israel J. Math.*, 58:103–124, 1987.
- [BS89a] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry*. Volume 1355 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1989.
- [BS89b] J. Bokowski and B. Sturmfels. An infinite family of minor-minimal nonrealizable 3-chirotopes. *Math. Z.*, 200:583–589, 1989.
- [EM82] J. Edmonds and A. Mandel. *Topology of Oriented Matroids*. Ph.D. thesis of A. Mandel, Univ. of Waterloo, 1982.
- [FL78] J. Folkman and J. Lawrence. Oriented matroids. *J. Combin. Theory Ser. B*, 25:199–236, 1978.
- [GP80] J.E. Goodman and R. Pollack. Proof of Grünbaum's conjecture on the stretchability of certain arrangements of pseudolines. *J. Combin. Theory Ser. A*, 29:385–390, 1980.
- [GPS90] J.E. Goodman, R. Pollack, and B. Sturmfels. The intrinsic spread of a configuration in \mathbb{R}^d . *J. Amer. Math. Soc.*, 3:639–651, 1990.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Interscience, London 1967; second edition edited by V. Kaibel, V. Klee, and G.M. Ziegler, volume 221 of *Graduate Texts in Math.*, Springer-Verlag, New York, 2003.

- [Grü71] B. Grünbaum. Arrangements of hyperplanes. In R.C. Mullin et al., editors, *Proc. Second Louisiana Conference on Combinatorics, Graph Theory and Computing*, Louisiana State University, Baton Rouge, 1971, pages 41–106.
- [Grü72] B. Grünbaum. *Arrangements and Spreads*. Volume 10 of *CBMS Regional Conf. Ser. in Math.*, Amer. Math. Soc., Providence, 1972.
- [JMSW89] B. Jaggi, P. Mani-Levitska, B. Sturmfels, and N. White. Constructing uniform oriented matroids without the isotopy property. *Discrete Comput. Geom.*, 4:97–100, 1989.
- [Knu92] D.E. Knuth. *Axioms and Hulls*. Volume 606 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, Berlin, 1992.
- [Kun86] J.P.S. Kung. *A Source Book in Matroid Theory*. Birkhäuser, Boston 1986.
- [Las80] M. Las Vergnas. Convexity in oriented matroids. *J. Combin. Theory Ser. B*, 29:231–243, 1980.
- [Lev26] F. Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss.*, 78:256–267, 1926.
- [Mnë88] N.E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, pages 527–544, volume 1346 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1988.
- [Oxl92] J. Oxley. *Matroid Theory*. Oxford Univ. Press, 1992.
- [Ric89] J. Richter. Kombinatorische Realisierbarkeitskriterien für orientierte Matroide. *Mitt. Math. Sem. Gießen*, 194:1–112, 1989.
- [Ric93] J. Richter-Gebert. Oriented matroids with few mutations. *Discrete Comput. Geom.*, 10:251–269, 1993.
- [Ric96a] J. Richter-Gebert. *Realization Spaces of Polytopes*. Volume 1643 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1996.
- [Ric96b] J. Richter-Gebert. Two interesting oriented matroids, *Doc. Math.*, 1:137–148, 1996.
- [RZ95] J. Richter-Gebert and G.M. Ziegler. Realization spaces of 4-polytopes are universal. *Bull. Amer. Math. Soc.*, 32:403–412, 1995.
- [Rin56] G. Ringel. Teilungen der Ebene durch Geraden oder topologische Geraden. *Math. Z.*, 64:79–102, 1956.
- [RS88] J.-P. Roudneff and B. Sturmfels. Simplicial cells in arrangements and mutations of oriented matroids. *Geom. Dedicata*, 27:153–170, 1988.
- [Sha79] R.W. Shannon. Simplicial cells in arrangements of hyperplanes. *Geom. Dedicata*, 8:179–187, 1979.
- [Sho91] P. Shor. Stretchability of pseudolines is *NP*-hard. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.*, pages 531–554, Amer. Math. Soc., Providence, 1991.
- [Suv88] P.Y. Suvorov. Isotopic but not rigidly isotopic plane systems of straight lines. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, pages 545–556, volume 1346 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1988.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995; revised edition 1998.
[Updates, corrections, etc. at <http://www.math.tu-berlin.de/~ziegler/>]

7 LATTICE POINTS AND LATTICE POLYTOPES

Alexander Barvinok

INTRODUCTION

Lattice polytopes arise naturally in number theory, algebraic geometry, optimization, combinatorics, probability, and analysis. They possess a very rich structure arising from the interaction of algebraic, convex, analytic, and combinatorial properties. In this chapter, we concentrate on the theory of lattice polytopes and only sketch their numerous applications. We briefly discuss their role in optimization and polyhedral combinatorics (Section 7.1). In Section 7.2 we discuss the *decision problem*, the problem of finding whether a given polytope contains a lattice point. In Section 7.3 we address the *counting problem*, the problem of counting all lattice points in a given polytope. The *asymptotic problem* (Section 7.4) explores the behavior of the number of lattice points in a varying polytope (for example, if a dilatation is applied to the polytope). Finally, in Section 7.5 we discuss *problems with quantifiers*. These problems are natural generalizations of the decision and counting problems. Whenever appropriate we address algorithmic issues. For general references in the area of computational complexity/algorithms see [AHU74]. We summarize the computational complexity status of our problems in Table 7.0.1.

TABLE 7.0.1 Computational complexity of basic problems.

PROBLEM NAME	BOUNDED DIMENSION	UNBOUNDED DIMENSION
Decision problem	polynomial	NP-hard
Counting problem	polynomial	#P-hard
Asymptotic problem	polynomial	#P-hard*
Problems with quantifiers	unknown; polynomial for $\forall \exists$	NP-hard

* in bounded codimension this reduces polynomially to volume computation

7.1 INTEGRAL POLYTOPES IN POLYHEDRAL COMBINATORICS

We describe some combinatorial and computational properties of integral polytopes. General references are [GLS88], [GW93], [Sch86], [Lag95], [DL97], and [Zie00].

GLOSSARY

\mathbb{R}^d : Euclidean d -dimensional space with scalar product $\langle x, y \rangle = x_1y_1 + \dots + x_dy_d$, where $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$.

\mathbb{Z}^d : The subset of \mathbb{R}^d consisting of the points with integral coordinates.

Polytope: The convex hull of finitely many points in \mathbb{R}^d .

Face of a polytope P : The intersection of P and the boundary hyperplane of a halfspace containing P .

Facet: A face of codimension 1.

Vertex: A face of dimension 0; the set of vertices of P is denoted by $\text{Vert } P$.

\mathcal{H} -description of a polytope (\mathcal{H} -polytope): A representation of the polytope as the set of solutions of finitely many linear inequalities.

\mathcal{V} -description of a polytope (\mathcal{V} -polytope): The representation of the polytope by the set of its vertices.

Integral polytope: A polytope with all of its vertices in \mathbb{Z}^d .

(0, 1)-polytope: A polytope P such that each coordinate of every vertex of P is either 0 or 1.

An integral polytope $P \subset \mathbb{R}^d$ can be given either by its \mathcal{H} -description or by its \mathcal{V} -description or (somewhat implicitly) as the convex hull of integral points in some other polytope Q : $P = \text{conv}\{Q \cap \mathbb{Z}^d\}$. In most cases it is difficult to translate one description into another. The following examples illustrate some typical kinds of behavior.

INTEGRALITY OF \mathcal{H} -POLYTOPES

It is an NP-hard problem to decide whether an \mathcal{H} -polytope $P \subset \mathbb{R}^d$ is integral. However, if the dimension d is fixed then the straightforward procedure of generating all the vertices of P and checking their integrality has polynomial time complexity. A rare case where an \mathcal{H} -polytope P is a priori integral is known under the general name of “total unimodularity.” Let A be an $n \times d$ integral matrix such that every minor of A is either 0 or 1 or -1 . Such a matrix A is called **totally unimodular**. If $b \in \mathbb{Z}^n$ is an integral vector then the set of solutions to the system of linear inequalities $Ax \leq b$ is an integral polytope in \mathbb{R}^d , provided this set is bounded. Examples of totally unimodular matrices include matrices of vertex-edge incidences of oriented graphs and of bipartite graphs. A complete characterization of totally unimodular matrices and a polynomial time algorithm for recognizing a totally unimodular matrix is provided by a theorem of P. Seymour (see [Sch86]). A family of integral polytopes, called **transportation polytopes**, were intensively studied in the literature (see [EKK84]). An example of a transportation polytope is provided by the set of $m \times n$ nonnegative matrices $x = (x_{ij})$ whose row and column sums are given positive integers. Integral points in this polytope are called **contingency tables**; they play an important role in statistics. A particular transportation polytope, called the **Birkhoff polytope**, is the set B_n of $n \times n$ nonnegative matrices with all row and column sums equal to 1. Alternatively, it may be described as the convex hull of the $n!$ permutation matrices $\pi(\sigma)_{ij} = \delta_{j\sigma(j)}$ for all permutations σ of the set $\{1, \dots, n\}$.

The notion of total unimodularity has been generalized in various directions, thus leading to new classes of integral polytopes (see [Cor01]).

\mathcal{V} -POLYTOPES WITH MANY VERTICES

There are several important situations where the explicit \mathcal{V} -description of an integral polytope is too long and a shorter description is desirable although not always

available. For example, a $(0, 1)$ -polytope may be given as the convex hull of the characteristic vectors

$$\chi_S(i) = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise} \end{cases}$$

for some combinatorially interesting family \mathcal{S} of subsets $S \subset \{1, \dots, d\}$ (see [GLS88] for various examples). The most famous example is the *traveling salesman polytope*, the convex hull TSP_n of the $(n - 1)!$ permutation matrices $\pi(\sigma)$ where σ is a permutation of the set $\{1, \dots, n\}$ consisting of precisely one cycle (cf. the Birkhoff polytope B_n above). The problem of the \mathcal{H} -description of the traveling salesman polytope has attracted a lot of attention (see [GW93] and [EKK84] for some references) because of its relevance to combinatorial optimization. C.H. Papadimitriou proved that it is a co-NP-complete problem to establish whether two given vertices of TSP_n are adjacent, i.e., connected by an edge. L. Billera and A. Sarangarajan proved that every $(0, 1)$ -polytope can be realized as a face of TSP_n for sufficiently large n (see [BS96]). Thus the combinatorics of TSP_n contrasts with the combinatorics of the Birkhoff polytope B_n .

Another important polytope arising in this way is the *cut polytope*, the famous counterexample to the Borsuk conjecture (see [DL97]). It is defined as the convex hull of the set of $n \times n$ matrices x_S , where

$$x_S(i, j) = \begin{cases} 1 & \text{if } |\{i, j\} \cap S| = 1 \text{ and } i \neq j, \\ 0 & \text{otherwise,} \end{cases}$$

where S ranges over all subsets of the set $\{1, \dots, n\}$.

CONVEX HULL OF INTEGRAL POINTS

Let $P \subset \mathbb{R}^d$ be a polytope. Then the convex hull P_I of the set $P \cap \mathbb{Z}^d$, if nonempty, is an integral polytope. Generally, the number of facets or vertices of P_I depends not only on the number of facets or vertices of P but also on the actual numerical size of the description of P (see [CHKM92]). Furthermore, it is an NP-complete problem to check whether a given point belongs to P_I , where P is given by its \mathcal{H} -description. If, however, the dimension d is fixed then the complexity of the facial description of the polytope P_I is polynomial in the complexity of the description of P . In particular, the number of vertices of P_I is bounded by a polynomial of degree $d - 1$ in the input size of P (see [CHKM92]).

Integrality imposes some restrictions on the combinatorial structure of a polytope. It is known that the combinatorial type of any 2- or 3-dimensional polytope can be realized by an integral polytope. J. Richter-Gebert constructed a 4-dimensional polytope with a nonintegral (and, therefore, nonrational) combinatorial type [Ric96]. Earlier, N. Mnëv had shown that for sufficiently large d there exist nonrational d -polytopes with $d + 4$ vertices. The number $N_d(V)$ of classes of integral d -polytopes having volume V and nonisomorphic with respect to affine transformations of \mathbb{R}^d preserving the integral lattice \mathbb{Z}^d has logarithmic order

$$c_1(d)V^{\frac{d-1}{d+1}} \leq \log N_d(V) \leq c_2(d)V^{\frac{d-1}{d+1}}$$

for some nonzero constants $c_1(d), c_2(d)$ [BV92].

7.2 DECISION PROBLEM

We consider the following general decision problem: Given a polytope $P \subset \mathbb{R}^d$ and a lattice $\Lambda \subset \mathbb{R}^d$, decide whether $P \cap \Lambda = \emptyset$ and, if the intersection is nonempty, find a point in $P \cap \Lambda$. We describe the main structural and algorithmic results for this problem. General references are [GL87], [GLS88], [GW93], [Sch86], and [Lag95].

GLOSSARY

Lattice: A discrete additive subgroup Λ of \mathbb{R}^d , i.e., $x - y \in \Lambda$ for any $x, y \in \Lambda$ and Λ does not contain limit points.

Basis of a lattice: A set of linearly independent vectors u_1, \dots, u_k such that every vector $y \in \Lambda$ can be (uniquely) represented in the form $y = m_1 u_1 + \dots + m_k u_k$ for some integers m_1, \dots, m_k .

Rank of a lattice: The cardinality of any basis of the lattice. If $\Lambda \subset \mathbb{R}^d$ has rank d , Λ is said to be of **full rank**.

Determinant of a lattice: For a lattice of rank k the k -volume of the parallelepiped spanned by any basis of the lattice.

Reciprocal lattice: For a full rank lattice $\Lambda \subset \mathbb{R}^d$, the lattice $\Lambda^* = \{x \in \mathbb{R}^d \mid \langle x, y \rangle \in \mathbb{Z} \text{ for all } y \in \Lambda\}$.

Polyhedron: An intersection of finitely many halfspaces in \mathbb{R}^d .

Convex body: A compact convex set in \mathbb{R}^d with nonempty interior.

Lattice Polytope: For a given lattice Λ , a polytope with all of its vertices in Λ .

Applying a suitable linear transformation one can reduce the decision problem to the case in which $\Lambda = \mathbb{Z}^k$ and $P \subset \mathbb{R}^k$ is a full-dimensional polytope, $k = \text{rank } \Lambda$.

The decision problem is known to be NP-complete for \mathcal{H} -polytopes as well as for \mathcal{V} -polytopes, although some special cases admit a polynomial time algorithm. In particular, if one fixes the dimension d then the decision problem becomes polynomially solvable. The main tool is provided by the so-called “flatness results.”

FLATNESS THEOREMS

Let $P \subset \mathbb{R}^d$ be a convex body and let $l \in \mathbb{R}^d$ be a nonzero vector. The number

$$\max\{\langle l, x \rangle \mid x \in P\} - \min\{\langle l, x \rangle \mid x \in P\}$$

is called the **width** of P with respect to l . For a full rank lattice $\Lambda \subset \mathbb{R}^d$, the minimum width of P with respect to a nonzero vector $l \in \Lambda^*$ is called the **lattice width** of P .

The following general result is known under the unifying name of “flatness theorem”.

THEOREM 7.2.1

There is a function $f : \mathbb{N} \rightarrow \mathbb{R}$ such that for any full rank lattice $\Lambda \subset \mathbb{R}^d$ and any convex body $P \subset \mathbb{R}^d$ with $P \cap \Lambda = \emptyset$, the lattice width of P does not exceed $f(d)$.

There are two types of results relating to the flatness theorem.

First, one may be interested in making $f(d)$ as small as possible. One can observe that $f(d) \geq d$: for some small $\epsilon > 0$, consider $\Lambda = \mathbb{Z}^d$ and the polytope P defined by the inequalities $x_1 + \dots + x_d \leq d - \epsilon$, $x_i \geq \epsilon$ for $i = 1, \dots, d$. It is known that one can choose $f(d) = O(d^{3/2})$ and it is conjectured that one can choose $f(d)$ as small as $O(d)$. W. Banaszczyk proved that if P is centrally symmetric, then one can choose $f(d) = O(d \log d)$, which is optimal up to a logarithmic factor. For these and related results, see [BLPS99]. There are results regarding the lattice width of some interesting classes of convex sets. Thus, if $P \subset \mathbb{R}^d$ is an ellipsoid which does not contain lattice points, then the lattice width of P is $O(d)$ [BLPS99]. J.-M. Kantor [Kan99] showed that for any $\alpha < 1/e$ one can find a sufficiently large d and a lattice simplex P such that P has no lattice points other than its vertices and such that the lattice width of P is at least αd . If P is a 3-dimensional lattice polytope which does not contain any lattice point other than its vertices, then the lattice width of P is 1 (see [Sca85]).

Second, one may be interested in the best width bound for which the corresponding vector $l \in \Lambda^*$ can be computed in polynomial time. The best bound known is $2^{O(d)}$, where l is polynomially computable even if the dimension d varies; see [GLS88]. J. Håstad proved that there is a polynomial time certificate certifying the distance from a given point $x \in \mathbb{R}^d$ to a given lattice $\Lambda \subset \mathbb{R}^d$ within a factor of $O(d^2)$. Namely, if Λ is a full-dimensional lattice, there exists a vector $l \in \Lambda^*$ with

$$\min_{u \in \Lambda} \|x - u\| \geq \frac{\{\{\langle l, x \rangle\}\}}{\|l\|} \geq \frac{1}{6d^2 + 1} \min_{u \in \Lambda} \|x - u\|,$$

where $\{\cdot\}$ is the distance to the nearest integer.

ALGORITHMS FOR THE DECISION PROBLEMS

Flatness theorems allow one to reduce the dimension in the decision problem: Assuming that $\Lambda = \mathbb{Z}^d$ and that the body P does not contain an integral point, one constructs a vector $l \in \mathbb{Z}^d$ for which P has a small width and reduces the d -dimensional decision problem to a family of $(d-1)$ -dimensional decision problems $P_i = \{x \in P \mid \langle l, x \rangle = i\}$, where i ranges between $\min\{\langle l, x \rangle \mid x \in P\}$ and $\max\{\langle l, x \rangle \mid x \in P\}$. This reduction is the main idea of polynomial time algorithms in fixed dimension. The best complexity known for the decision problem in terms of the dimension d is $d^{O(d)}$.

Constructing l efficiently relies on two major components (see [GLS88]). First, a linear transformation T is computed, such that the image $T(P)$ is “almost round,” meaning that $T(P)$ is sandwiched between a pair of concentric balls with the ratio of their radii bounded by some small constant depending only on the dimension d . At this stage, a linear programming algorithm is used. Second, a reasonably short nonzero vector u is constructed in the lattice Λ^* reciprocal to $\Lambda = T(\mathbb{Z}^d)$. A basis reduction algorithm is used at this stage. Then we let $l = (T^*)^{-1}u$.

One can streamline the process by using the generalized lattice reduction [LS92] tailored to a given polytope. A polynomial time algorithm based on counting lattice points in the polytope and not using the flatness argument is sketched in [BP99].

MINKOWSKI'S CONVEX BODY THEOREM

The following classical result, known as “Minkowski’s convex body theorem,” provides a very useful criterion.

THEOREM 7.2.2

Suppose that $B \subset \mathbb{R}^d$ is a convex body, centrally symmetric about the origin 0, and $\Lambda \subset \mathbb{R}^d$ is a lattice of full rank. If $\text{vol } B \geq 2^d \det \Lambda$ then B contains a nonzero point of Λ .

For the proof and various generalizations see, for example, [GL87]. An important generalization (Minkowski's Second Theorem) concerns the existence of i linearly independent lattice points in a convex body. Namely, if $\lambda_i = \inf \left\{ \lambda > 0 \mid \lambda B \cap \Lambda \text{ contains } i \text{ linearly independent points} \right\}$ is the "ith successive minimum," then $\lambda_1 \dots \lambda_d \leq (2^d \det \Lambda) / (\text{vol } B)$.

If B is a convex body such that $\text{vol } B = 2^d \det \Lambda$ but B does not contain a nonzero lattice point in its interior, then B is called **extremal**. Every extremal body is necessarily a polytope. Moreover, this polytope contains at most $2(2^d - 1)$ facets, and therefore, for every dimension d , there exist only finitely many combinatorially different extremal polytopes. The contracted polytope $P = \{x/2 \mid x \in B\}$ has the property that its lattice translates $P + x \mid x \in \Lambda$ tile the space \mathbb{R}^d . Such a tiling polytope is called a **parallelohedron**. Similarly, for every dimension d there exist only finitely many combinatorially different parallelohedra. Parallelohedra can be characterized intrinsically: a polytope is a parallelohedron if and only if it is centrally symmetric, every facet of it is centrally symmetric, and every class of parallel ridges (($d - 2$)-dimensional faces) consists of four or six ridges. If $q : \mathbb{R}^d \rightarrow \mathbb{R}$ is a positive definite quadratic form, then the **Dirichlet-Voronoi cell** $P_q = \{x \mid q(x) \leq q(x - \lambda) \text{ for any } \lambda \in \Lambda\}$ is a parallelohedron. The problem of finding whether a centrally symmetric polyhedron P contains a nonzero point from a given lattice Λ is known to be NP-complete even in the case of the standard cube $P = \{(x_1, \dots, x_d) \mid -1 \leq x_i \leq 1\}$. For fixed dimension d there exists a polynomial time algorithm since the problem obviously reduces to the decision problem (one can add the extra inequality $x_1 + \dots + x_d \geq 1$).

VOLUME BOUNDS

An integral simplex in \mathbb{R}^d containing no integral points other than its vertices has volume $1/2$ if $d = 2$ but already for $d = 3$ can have an arbitrarily large volume (the smallest possible volume of such a simplex is $1/d!$). On the other hand, if an integral polytope P contains precisely $k > 0$ integral points then its volume is bounded by a function of k and d . The best bound known, $\text{vol } P \leq k(7(k+1))^{2^{d+1}}$, is due to J. Lagarias and G.M. Ziegler (see [Lag95]).

7.3 COUNTING PROBLEM

We consider the following problem: Given a polytope $P \subset \mathbb{R}^d$, compute exactly or approximately the number of integral points $|P \cap \mathbb{Z}^d|$ in P .

For counting in general convex bodies see [CHKM92]. For some applications in the combinatorics of generating functions and representation theory see, for example, [BZ88] and [Sta86]. For applications in statistical physics (computing permanents) and statistics (counting contingency tables), see [JS97]. For general information see the surveys [GW93] and [BP99].

GLOSSARY

Rational polyhedron: The set

$$P = \{x \in \mathbb{R}^d \mid \langle a_i, x \rangle \leq \beta_i, i = 1, \dots, m\},$$

where $a_i \in \mathbb{Z}^d$ and $\beta_i \in \mathbb{Z}$ for $i = 1, \dots, m$.

Polyhedral cone: A set $K \subset \mathbb{R}^d$ of the form $K = \{\sum_{i=1}^k \lambda_i u_i \mid \lambda_i \geq 0, i = 1, \dots, k\}$ for some vectors $u_1, \dots, u_k \in \mathbb{R}^d$. The vectors u_1, \dots, u_k are called *generators* of K .

Rational cone: A polyhedral cone having a set of generators belonging to \mathbb{Z}^d . A rational cone is a rational polyhedron.

Simple cone: A polyhedral cone generated by linearly independent vectors.

Cone of feasible directions at a point: The cone

$$K_v = \{x \mid v + \epsilon x \in P \text{ for all sufficiently small } \epsilon > 0\}$$

for a point v of a polytope P . If v is a vertex, then the cone K_v is generated by the vectors $u_i = v_i - v$, where $[v_i, v]$ is an edge of P .

Fundamental parallelepiped of a simple cone: The set

$$\Pi = \{\lambda_1 u_1 + \dots + \lambda_k u_k \mid 0 \leq \lambda_i < 1, i = 1, \dots, k\},$$

where u_1, \dots, u_k are linearly independent generators of the cone.

Unimodular cone: A rational simple cone $K \subset \mathbb{R}^d$ whose fundamental parallelepiped does not contain points of \mathbb{Z}^d other than 0.

Simple polytope: A polytope P such that the cone K_v of feasible directions is simple for every vertex v of P .

Totally unimodular polytope: An integral polytope P such that the cone K_v of feasible directions is unimodular for every vertex v of P .

GENERAL INFORMATION

The counting problem is known to be $\#P$ -hard even for an integral \mathcal{H} - or \mathcal{V} -polytope.

However, if the dimension d is fixed, one can solve the counting problem in polynomial time (see [BP99]).

SOME EXPLICIT FORMULAS IN LOW DIMENSIONS

The classical Pick formula expresses the number of integral points in a convex integral polygon $P \subset \mathbb{R}^2$ in terms of its area and the number of integral points on the boundary ∂P :

$$|P \cap \mathbb{Z}^2| = \text{area}(P) + \frac{1}{2} \cdot |\partial P \cap \mathbb{Z}^2| + 1$$

(see, for example, [Mor93b], [GW93]). This formula almost immediately gives rise to a polynomial time algorithm for counting integral points in integral polygons.

An important explicit formula for the number of integral points in a lattice tetrahedron of a special kind was proven by L. Mordell. Let a, b, c be pairwise coprime positive integers and $\Delta(a, b, c) \subset \mathbb{R}^3$ be the tetrahedron with vertices $(0, 0, 0)$, $(a, 0, 0)$, $(0, b, 0)$, and $(0, 0, c)$. Then

$$\begin{aligned} |\Delta(a, b, c) \cap \mathbb{Z}^3| &= \frac{abc}{6} + \frac{ab + ac + bc + a + b + c}{4} + \\ &\quad \frac{1}{12} \left(\frac{ac}{b} + \frac{bc}{a} + \frac{ab}{c} + \frac{1}{abc} \right) - s(bc, a) - s(ac, b) - s(ab, c) + 2. \end{aligned} \quad (7.3.1)$$

Here

$$s(p, q) = \sum_{i=1}^q \left(\left(\frac{i}{q} \right) \right) \left(\left(\frac{pi}{q} \right) \right), \quad \text{where } ((x)) = x - 0.5(\lfloor x \rfloor + \lceil x \rceil),$$

is the Dedekind sum. A similar formula was found in dimension 4. The famous reciprocity relation $s(p, q) + s(q, p) = (p/q + q/p + 1/pq - 3)/12$ allows one to compute the Dedekind sum $s(p, q)$ in polynomial time. A version of formula (7.3.1) was used by M. Dyer to construct polynomial time algorithms for the counting problem in dimensions 3 and 4. Formula (7.3.1) was generalized to an arbitrary tetrahedron by J. Pommersheim (see [BP99]). A generalization to higher dimensions was suggested in [CS94].

Computationally efficient formulas for the number of lattice points are known for some particular polytopes, most notably zonotopes. Given integral points v_1, \dots, v_n in \mathbb{R}^d , a **zonotope** spanned by v_1, \dots, v_n is the polytope

$$P = \left\{ \lambda_1 v_1 + \dots + \lambda_n v_n \mid 0 \leq \lambda_i \leq 1 \text{ for } i = 1, \dots, n \right\}.$$

For each subset $S \subset \{v_1, \dots, v_n\}$ of linearly independent points, let a_S be the index of the sublattice generated by S in the lattice $\mathbb{Z}^d \cap \text{span}(S)$, where $a_\emptyset = 1$. Then $|P \cap \mathbb{Z}^d| = \sum_S a_S$ (see [Chapter 4](#), Problem 31 of [Sta86]).

EXPONENTIAL SUMS

A powerful tool for solving the counting problem exactly is provided by **exponential sums**, which may be regarded as generating functions for sets of integral points.

Let $P \subset \mathbb{R}^d$ be a polytope and $c \in \mathbb{R}^d$ be a vector. We consider the exponential sum $\sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\}$. If $c = 0$ we get the number of integral points in P . The reason for introducing the parameter c is that for a “generic” c the exponential sums reveal some nontrivial algebraic properties that become less visible when $c = 0$. To describe these properties we need to consider exponential sums over rational polyhedra and, in particular, over cones.

EXPONENTIAL SUMS OVER RATIONAL POLYHEDRA

Let $K \subset \mathbb{R}^d$ be a rational cone without straight lines generated by vectors u_1, \dots, u_k in \mathbb{Z}^d . Then the series $\sum_{x \in K \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\}$ converges for any c such that $\langle c, u_i \rangle < 0$

for all $i = 1, \dots, k$ and defines a meromorphic function of c which we denote by $f_K(c)$. For a simple rational cone $K \subset \mathbb{R}^d$ with linearly independent generators u_1, \dots, u_k we have

$$f_K(c) = \left(\sum_{x \in \Pi \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\} \right) \cdot \prod_{i=1}^k \frac{1}{1 - \exp\{\langle c, u_i \rangle\}},$$

where Π is the fundamental parallelepiped of K . In particular, if K is unimodular then

$$f_K(c) = \prod_{i=1}^k \frac{1}{1 - \exp\{\langle c, u_i \rangle\}},$$

since the corresponding sum is just the multiple geometric series. Generally speaking, the farther a given cone is from being unimodular, the more complicated the formula for $f_K(c)$ will be.

These results are known in many different forms (see, for example, Section 4.6 of [Sta86]). Furthermore, the function $f_K(c)$ can be extended to a finitely additive measure, defined on rational polyhedra in \mathbb{R}^d and taking its values in the space of meromorphic functions in d variables, so that the measure of a rational polyhedron with a straight line is equal to 0. To state the result precisely, let us associate with every set $A \in \mathbb{R}^d$ its **indicator function** $[A] : \mathbb{R}^d \rightarrow \mathbb{R}$, given by

$$[A](x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise.} \end{cases}$$

The following result was proved by A.G. Khovanskii and A. Pukhlikov [KP92] and, independently, by J. Lawrence [Law91].

THEOREM 7.3.1 *Lawrence-Khovanskii-Pukhlikov Theorem*

There exists a map that associates, to every rational polyhedron $P \subset \mathbb{R}^d$, a meromorphic function $f_P(c)$, $c \in \mathbb{C}^d$, such that:

The correspondence $P \mapsto f_P$ preserves linear dependencies among indicator functions of rational polyhedra:

$$\sum_{i=1}^m \alpha_i [P_i] = 0 \quad \text{implies} \quad \sum_{i=1}^m \alpha_i f_{P_i}(c) = 0$$

for rational polyhedra P_i and integers α_i ;

If P does not contain straight lines, then

$$f_P(c) = \sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\}$$

for all c such that the series converges absolutely;

If P contains a straight line then $f_P(c) \equiv 0$.

If $P + m$ is a translation of P by an integral vector m then

$$f_{P+m}(c) = \exp\{\langle c, v \rangle\} f_P(c).$$

For example, suppose that $d = 1$ and let us choose $P_+ = [0, +\infty)$, $P_- = (-\infty, 0]$, $P_0 = \{0\}$, and $P = (-\infty, +\infty)$. Then

$$f_{P_+}(c) = \sum_{x=0}^{+\infty} \exp\{cx\} = \frac{1}{1 - \exp\{c\}} \quad \text{and} \quad f_{P_-}(c) = \sum_{x=0}^{-\infty} \exp\{cx\} = \frac{1}{1 - \exp\{-c\}}.$$

Moreover, $f_{P_0} = 1$ and $f_P = 0$ since P contains a straight line. We see that $[P] = [P_+] + [P_-] - [P_0]$ and that $f_P = f_{P_+} + f_{P_-} - f_{P_0}$.

Let $P \subset \mathbb{R}^d$ be a rational polytope and let $v \in P$ be its vertex. Let us consider the translation $v + K_v$ of the cone K_v of feasible directions at v . The following crucial result was proved by M. Brion [Bri98].

THEOREM 7.3.2 *Brion's Theorem*

Let $P \subset \mathbb{R}^d$ be a rational polytope. Then

$$\sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle c, x \rangle\} = \sum_{v \in \text{Vert } P} f_{v+K_v}(c).$$

If the polytope is integral, we have $f_{v+K_v}(c) = \exp\{\langle c, v \rangle\} f_{K_v}(c)$. We note that if K is a unimodular cone and v is a rational vector then $f_{K+v} = \exp\{\langle c, w \rangle\} f_K(c)$, where $w \in \mathbb{Z}^d$ is a certain “rounding” of v with respect to K . Namely, assume that K is the conic hull of some integral vectors u_1, \dots, u_d that constitute a basis of \mathbb{Z}^d . Let u_1^*, \dots, u_d^* be the biorthogonal basis such that $\langle u_i^*, u_j \rangle = \delta_{ij}$. Then $w = \sum_{i=1}^d \lceil \langle v, u_i^* \rangle \rceil u_i$.

Essentially, Theorem 7.3.2 can be deduced from Theorem 7.3.1 by noticing that the indicator function of every (rational) polyhedron P can be written as the sum of the indicator functions $[v + K_v]$ modulo indicator functions of (rational) polyhedra with straight lines; see [BP99].

Brion's formula allows one to reduce the counting of integral points in polytopes to the counting of points in polyhedral cones, a much easier problem. Below we discuss two instances where the application of exponential sums and Brion's identities leads to an efficient computational solution of the counting problem.

COUNTING IN FIXED DIMENSION

The following result was obtained by A. Barvinok (see [BP99]).

THEOREM 7.3.3

Let us fix the dimension d . Then there exists a polynomial time algorithm that, for any given rational polytope $P \subset \mathbb{R}^d$, computes the number $|P \cap \mathbb{Z}^d|$ of integral points in P .

THE IDEA OF THE ALGORITHM

We assume that the polytope is given by its \mathcal{V} -description. Let us choose a “generic” $c \in \mathbb{Q}^d$. We can compute the number $|P \cap \mathbb{Z}^d|$ as the limit of the exponential sum

$$\lim_{t \rightarrow 0} \sum_{x \in P \cap \mathbb{Z}^d} \exp\{\langle tc, x \rangle\},$$

where t is a real parameter. Using Brion's Theorem 7.3.2, we reduce the problem to the computation of the constant term in the Laurent expansion of the meromorphic function $f_v(t) = f_{v+K_v}(tc)$, where v is a vertex of P and K_v is the cone of feasible directions at v . If K_v is a unimodular cone, we have an explicit formula for $f_{v+K_v}(c)$

(see above) and thus can easily compute the desired term. However, for $d > 1$ the cone K_v does not have to be unimodular. It turns out, nevertheless, that for any given rational cone K one can construct in polynomial time a decomposition $K = \sum_{i \in I} \epsilon_i K_i$, $\epsilon_i \in \{-1, 1\}$, of the “inclusion-exclusion” type, where the cones K_i are unimodular (see below). Thus one can get an explicit expression $f_{v+K_v}(c) = \sum_{i \in I} \epsilon_i \cdot f_{v+K_i}(c)$ and then compute the constant term of the Laurent expansion of $f_v(t)$. The complexity of the algorithm in terms of the dimension d is $d^{O(d)}$.

COUNTING IN TOTALLY UNIMODULAR POLYTOPES

One can efficiently count the number of integral points in a totally unimodular polytope given by its vertex description even in varying dimension.

THEOREM 7.3.4 [BP99]

There exists an algorithm that, for any d and any given integral vertices $v_1, \dots, v_m \in \mathbb{Z}^d$ such that the polytope $P = \text{conv}\{v_1, \dots, v_m\}$ is totally unimodular, computes the number of integral points of P in time linear in the number m of vertices.

Moreover, the same result holds for rational polytopes with unimodular cones of feasible directions at the vertices. The algorithm uses Brion’s formulas (Theorem 7.3.2) and the explicit formula above for the exponential sum over a unimodular cone.

EXAMPLE: COUNTING CONTINGENCY TABLES

Suppose A is an $n \times d$ totally unimodular matrix (see Section 7.1). Let us choose $b \in \mathbb{Z}^n$ such that the set P_b of solutions to the system $Ax \leq b$ of linear inequalities is a simple polytope. Then P_b is totally unimodular.

For example, if we know all the vertices of a simple transportation polytope P , we can compute the number of integral points of P in time linear in the number of vertices of P .

One can construct an efficient algorithm for counting integral points in a polytope that is somewhat “close” to totally unimodular and for which the explicit formulas for $f_{K_v}(c)$ are therefore not too long.

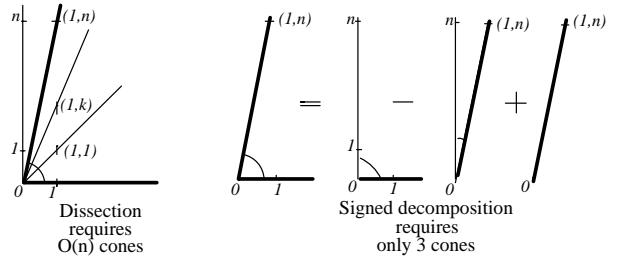
One particular application is counting contingency tables (see Section 7.1). Implementation of the algorithm based on Brion’s formula, codes, and numerical results, as well as other algebraic approaches, are discussed in [DLS03].

CONNECTIONS WITH TORIC VARIETIES

It was first observed by A.G. Khovanskii in the 1970s, and has since then become widely known, that the number of integral points in an integral polytope is related to some algebro-geometric invariants of the associated toric variety (see [Oda88]). Naturally, for smooth toric varieties (they correspond to totally unimodular polytopes) computation is much easier. Various formulas for the number of integral points in polytopes were first obtained for totally unimodular polytopes and then,

by the use of resolution of singularities, generalized to arbitrary integral polytopes (see, for example, [BP99]). Resolution of singularities of toric varieties reduces to dissection of a polyhedral cone into unimodular cones. However, as one can see, it is impossible to subdivide a rational cone into polynomially (in the input) many unimodular cones even in dimension $d = 2$. For example (see Figure 7.3.1), the plane cone K generated by the points $(1, 0)$ and $(1, n)$ cannot be subdivided into fewer than $2n - 1$ unimodular cones, whereas a polynomial time subdivision would give a polynomial in $\log n$ cones. On the other hand, if we allow a signed linear combination of the inclusion-exclusion type, then one can easily represent this cone as a combination of 3 unimodular cones: $[K] = [K_1] - [K_2] + [K_3]$, where K_1 is generated by the basis $(1, 0)$ and $(0, 1)$, K_2 is generated by $(0, 1)$ and $(1, n)$, and K_3 is generated by $(1, n)$. Moreover, modulo rational cones with straight lines (cf. Theorem 7.3.1), we need to use only two unimodular cones: $[K] = [K_3] + [K_4]$ modulo rational cones with straight lines, where K_3 is the cone generated by $(1, n)$ and $(0, -1)$ and K_4 is the cone generated by $(0, 1)$ and $(1, 0)$. Consequently, from Theorem 7.3.1, $f_K(c) = (1 - \exp\{c_1 + nc_2\})^{-1}(1 - \exp\{-c_2\})^{-1} + (1 - \exp\{c_1\})^{-1}(1 - \exp\{c_2\})^{-1}$ for $c = (c_1, c_2)$. As we have mentioned above, once we allow “signed” combinations, any rational polyhedral cone can be decomposed into unimodular cones in polynomial time, provided the dimension is fixed. Moreover, if we allow decompositions modulo rational cones with straight lines, the algorithm can be sped up further: roughly from $2^{O(d^2)}$ to $2^{O(d)}$ (see [BP99]).

FIGURE 7.3.1
Decomposition of a cone
into unimodular cones.



CONNECTIONS WITH VALUATIONS

The number of integral points $\nu(P) = |P \cap \mathbb{Z}^d|$ in an integral polytope $P \subset \mathbb{R}^d$ is a **valuation**, that is, it preserves linear relations among indicator functions of polytopes; and it is lattice-translation-invariant, i.e., $\nu(P + l) = \nu(P)$ for any $l \in \mathbb{Z}^d$. General properties of valuations and the related notion of the “polytope algebra” have been intensively studied (see, for example, [McM93] and [Mor93a]). Various identities discovered in this area might prove useful in dealing with particular counting problems (see [BP99]). For example, if the transportation polytope P_b is not simple, one can apply the following recipe. First, triangulating the normal cone at the vertex, we represent it as a combination of unimodular cones (we discard lower-dimensional cones). Then, passing to the dual cones, we get the desired representation of the cone of feasible directions (we discard cones with straight lines).

ANALYTICAL METHODS

The number $|P \cap \Lambda|$ of lattice points of Λ in the polytope P can be interpreted as the integral over P of the periodic delta-function

$$\sum_{y \in \Lambda} \delta_y(x) = (\det \Lambda)^{-1} \sum_{l \in \Lambda^*} \exp\{2\pi i \langle l, x \rangle\}.$$

Depending on the interpretation of this integral one can get various formulas. For example, if the above series is approximated as $t \rightarrow \infty$ by the theta-series

$$\theta_t(x) = t^{d/2} \sum_{y \in \Lambda} \exp\{-t\pi \|x - y\|^2\} = (\det \Lambda)^{-1} \sum_{l \in \Lambda^*} \exp\{-\pi \|l\|^2/t\} \exp\{2\pi i \langle l, x \rangle\},$$

then as the limit $\lim_{t \rightarrow \infty} \int_P \theta_t(x) dx$ one gets the number of lattice points in P , each lattice point y counted with weight equal to the spherical measure $\gamma(K_y)$ of the cone K_y of feasible directions at y normalized in such a way that the spherical measure of \mathbb{R}^d is equal to 1 (see [GL87] and [BP99] for some information about this weighted counting).

Applying Parseval's theorem one can get the famous Siegel identity (see [GL87])

$$2^d \det \Lambda - \text{vol } B = \frac{1}{\text{vol } B} \sum_{l \in \Lambda^* \setminus 0} \left| \int_B \exp\{-\pi i \langle l, x \rangle\} dx \right|^2,$$

where B is a 0-symmetric convex body not containing nonzero lattice points (cf. Theorem 7.2.2).

R. Diaz and S. Robins [DR97] have obtained nice “cotangent” formulas for the number of integral points in an integral simplex by integrating an appropriately “smoothed out” sum $f(x) = \sum_{l \in \mathbb{Z}^d} \exp\{2\pi i \langle l, x \rangle\}$. Suppose that $P \subset \mathbb{R}^d$ is an integral simplex, that is, the convex hull of $d + 1$ affinely independent integral vectors v_1, \dots, v_{d+1} . Embedding $\mathbb{R}^d \rightarrow \mathbb{R}^d$ as the affine hyperplane $x_{d+1} = 1$, Diaz and Robins express the number of integral points in P in terms of a certain sum over finite abelian groups that are factors of $\mathbb{Z}^{d+1} \cap \text{span}(v_{i_1}, \dots, v_{i_k})$ modulo the sublattice generated by v_{i_1}, \dots, v_{i_k} . Relations of this construction to higher Dedekind sums are discussed in [BP99].

The following simple observation often leads to practically efficient (although theoretically exponential time) algorithms. Suppose we want to count integral points $x = (x_1, \dots, x_d)$ in a polyhedron $P \subset \mathbb{R}^d$ defined by the equations

$$\sum_{j=1}^d a_{ij} x_j = b_i, \quad i = 1, \dots, m$$

and inequalities

$$x_j \geq 0, \quad j = 1, \dots, d,$$

where $A = (a_{ij})$ is a given $m \times d$ integer matrix. Let z_1, \dots, z_m be (complex) variables and let

$$f_A(z_1, \dots, z_m) = \prod_{j=1}^d \sum_{x=0}^{+\infty} z_1^{a_{1j}x} z_2^{a_{2j}x} \cdots z_m^{a_{mj}x} = \prod_{j=1}^d \frac{1}{1 - z_1^{a_{1j}} z_2^{a_{2j}} \cdots z_m^{a_{mj}}}.$$

Thus the number $|P \cap \mathbb{Z}^d|$ is equal to the coefficient of $z_1^{b_1} \cdots z_m^{b_m}$ in $f_A(z_1, \dots, z_m)$ in a neighborhood of $z_1 = \dots = z_m = 0$. This coefficient may be extracted by numerical differentiation, or by (repeated) application of the residue formula, or by numerical integration using the Cauchy or Martinelli-Bochner integral representation for the Taylor coefficients. M. Beck and D. Pixton [BP02] report results on numerical computation for the problem of counting contingency tables using repeated application of the residue formula.

As discussed in [BV97b], various identities relating functions f_A mirror corresponding identities among indicator functions of rational polyhedra. In particular, decompositions of f_A into “simple fractions” correspond to decompositions of P into simple cones.

Quite a few useful inequalities for the number of lattice points can be found in [GW93], [Lag95], and [GL87]. Blichfeldt’s inequality states that

$$|B \cap \Lambda| \leq \frac{d!}{\det \Lambda} \text{vol } B + d,$$

where B is a convex body containing at least $d + 1$ affinely independent lattice points. Davenport’s inequality implies that

$$|B \cap \mathbb{Z}^d| \leq \sum_{i=0}^d \binom{d}{i} V_i(B),$$

where the V_i are the intrinsic volumes. A conjectured stronger inequality, $|B \cap \mathbb{Z}^d| \leq V_0(K) + \dots + V_d(K)$, was shown to be false in dimensions $d \geq 207$, although it is correct for $d = 2, 3$. Furthermore, H. Hadwiger proved that $|B \cap \mathbb{Z}^d| \geq \sum_{i=0}^d (-1)^{d-i} V_i(B)$, provided $B \subset \mathbb{R}^d$ is a convex body having a nonempty interior (see [Lag95]).

PROBABILISTIC METHODS

Often, we need the number of integral points only approximately. Probabilistic methods based on Monte-Carlo methods have turned out to be quite successful. The main idea can be described as follows (see [JS97]). Suppose we want to approximate the cardinality of a finite set X (for example, X may be the set of lattice points in a polytope). Suppose, further, that we can present a “filtration” $X_0 \subset X_1 \subset \dots \subset X_n = X$, where $|X_0| = 1$ (in general, we require $|X_0|$ to be small) and $|X_{i+1}|/|X_i| \leq 2$ (in general, we require the ratio $|X_{i+1}|/|X_i|$ to be reasonably small). Finally, suppose that we have an efficient procedure for sampling an element $x \in X_i$ uniformly at random (in practice, we settle for “almost uniform” sampling). Given an $\epsilon > 0$ and a $\delta > 0$, with probability at least $1 - \delta$ one can estimate the ratio $|X_{i+1}|/|X_i|$, within a relative error ϵ/n , by sampling $O(n\epsilon^{-1} \ln \delta^{-1})$ points at random from X_{i+1} and counting how many times the points end up in X_i . Then, by “telescoping,” with probability at least $(1 - \delta)^n$, we estimate

$$|X| = |X_n| = \frac{|X_n|}{|X_{n-1}|} \cdots \frac{|X_{i+1}|}{|X_i|} \cdots \frac{|X_2|}{|X_1|}$$

within relative error ϵ .

The bottleneck of the method is the ability to sample a point $x \in X_i$ uniformly at random. To achieve that, a Markov chain on X_i is designed, which converges fast (“mixes rapidly”) to the uniform distribution. Usually, there are some natural candidates for such Markov chains and the main difficulty is to establish whether they indeed mix rapidly.

Counting various combinatorial structures can be interpreted as counting vertices in a certain $(0, 1)$ -polytope. For example, computing the number of perfect matchings in a given bipartite graph on $n + n$ vertices, or, equivalently, computing the permanent of a given $n \times n$ matrix of 0’s and 1’s, can be viewed as counting the number of vertices in a particular face of the Birkhoff polytope B_n . M. Jerrum, A. Sinclair, and E. Vigoda [JSV01] have constructed a polynomial-time probabilistic algorithm to approximate the permanent of any given nonnegative matrix. B. Morris and A. Sinclair [MS99] have presented a polynomial-time probabilistic algorithm to compute the number of $(0, 1)$ -vectors (x_1, \dots, x_n) satisfying the inequality $a_1x_1 + \dots + a_nx_n \leq b_n$, where a_i and b are given positive integers.

In the problem of counting contingency tables, the following simple Markov chain was proposed by P. Diaconis to obtain a random contingency table with prescribed row and column sums. Given a contingency table A , we select at random a pair of rows (i, i') and a pair of columns (j, j') and obtain a new table with the same row and column sums by incrementing a_{ij} and $a_{i'j'}$ by one and decrementing $a_{ij'}$ and $a_{i'j}$ by one, provided this leaves all entries nonnegative. This Markov chain is observed to be rapidly mixing in practice (see [JS97]).

One can obtain some crude and quick bounds on the number of vertices of a $(0, 1)$ -polytope by computing the Hamming distance from a random $(0, 1)$ -vector to the nearest vertex of the polytope [BS01]. Often, this distance can be efficiently computed by solving an appropriate combinatorial optimization problem. This way one can determine, for example, whether the number of vertices is exponentially large in the dimension n in some rigorously defined sense.

7.4 ASYMPTOTIC PROBLEMS

If $P \subset \mathbb{R}^d$ is an integral polytope then the number of integral points in the dilated polytope $nP = \{nx \mid x \in P\}$ for a natural number n is a polynomial in n , known as the Ehrhart polynomial. We review several results concerning the Ehrhart polynomial and its generalizations.

GLOSSARY

Todd polynomial: The homogeneous polynomial $\text{td}_k(x_1, \dots, x_m)$ of degree k defined as the coefficient of t^k in the expansion

$$\prod_{i=1}^m \frac{tx_i}{1 - \exp\{-tx_i\}} = \sum_{k=0}^{\infty} t^k \cdot \text{td}_k(x_1, \dots, x_m).$$

Tangent cone at a face of a polytope: The cone K_F of feasible directions at any point in the relative interior of the face $F \subset P$.

Apex of a cone: The largest linear subspace contained in the cone.

Dual cone: The cone $K^* = \{x \in \mathbb{R}^d \mid \langle x, y \rangle \leq 0 \text{ for all } y \in K\}$, where $K \subset \mathbb{R}^d$ is a given cone.

vol_k: The normalized k -volume of a k -dimensional rational polytope $P \subset \mathbb{R}^d$ computed as follows. Let $L \subset \mathbb{R}^d$ be the k -dimensional linear subspace parallel to the affine span of P . Then $\text{vol}_k(P)$ is the Euclidean k -dimensional volume of P in the affine span of P divided by the determinant of the lattice $\Lambda = \mathbb{Z}^d \cap L$.

EHRHART POLYNOMIALS

The following fundamental result was suggested by Ehrhart (see, for example, [Sta86] and [Sta83]).

THEOREM 7.4.1

Let $P \subset \mathbb{R}^d$ be an integral polytope. For a natural number n we denote by $nP = \{nx \mid x \in P\}$ the n -fold dilatation of P . Then the number of integral points in nP is a polynomial in n :

$$|nP \cap \mathbb{Z}^d| = E_P(n) \quad \text{for some polynomial } E_P(x) = \sum_{i=0}^d e_i(P) \cdot x^i.$$

Moreover, for positive integers n the value of $(-1)^{\deg E_P} E_P(-n)$ is equal to the number of integral points in the relative interior of the polytope nP (the “reciprocity law”).

The polynomial E_P is called the **Ehrhart polynomial** and its coefficients $e_i(P)$ are called **Ehrhart coefficients**. For various proofs of Theorem 7.4.1 see, for example, [Sta86], [Sta83] and [BP99]. The existence of the Ehrhart polynomials and the reciprocity law can be derived from the single fact that the number of integral points in a polytope is a lattice-translation-invariant valuation (see [McM93] and Section 7.3 above).

If P is a rational polytope, we define $e_k(P) = n^{-k} e_k(nP)$, where n is a positive integer such that nP is an integral polytope. For an integral polytope $P \subset \mathbb{R}^d$, one has $|P \cap \mathbb{Z}^d| = e_0(P) + e_1(P) + \dots + e_d(P)$. (This formula is no longer true, however, if P is a general rational polytope.) The Ehrhart coefficients constitute a basis of all additive functions (valuations) ν on rational polytopes that are invariant under unimodular transformations (see [McM93] and [GW93]).

GENERAL PROPERTIES

It is known that $e_0(P) = 1$, $e_d(P) = \text{vol}_d(P)$, and $e_{d-1}(P) = \frac{1}{2} \sum_F \text{vol}_{d-1} F$, where the sum is taken over all the facets of P . Thus, computation of the two highest coefficients reduces to computation of the volume. In fact, the computation of any fixed number of the highest Ehrhart coefficients of an \mathcal{H} -polytope reduces in polynomial time to the computation of the volumes of faces; see [BP99] and also below.

EXISTENCE OF LOCAL FORMULAS

The Ehrhart coefficients can be decomposed into a sum of “local” summands. The following theorem was proven by P. McMullen (see [McM93], [Mor93a], and [BP99]).

THEOREM 7.4.2

For any natural numbers k and d there exists a real valued function $\mu_{k,d}$, defined on the set of all rational polyhedral cones $K \subset \mathbb{R}^d$, such that for every rational full-dimensional polytope $P \subset \mathbb{R}^d$ we have

$$e_k(P) = \sum_F \mu_{k,d}(K_F) \cdot \text{vol}_k F,$$

where the sum is taken over all k -dimensional faces F of P and K_F is the tangent cone at the face F . Moreover, one can choose $\mu_{k,d}$ to be an additive measure on polyhedral cones.

The function $\mu_{k,d}$ that satisfies the conditions of Theorem 7.4.2 is not unique and it is a difficult problem to choose a computationally efficient $\mu_{k,d}$ (see also Morelli’s formulas, below). However, for some specific values of k and d a “canonical” choice of $\mu_{k,d}$ has long been known.

EXAMPLE

For a cone $K \subset \mathbb{R}^d$, let $\gamma(K)$ be the spherical measure of K normalized in such a way that $\gamma(\mathbb{R}^d) = 1$. Thus $\gamma(K) = 0.5$ if K is a halfspace. One can choose $\mu_{d,d} = \mu_{d-1,d} = \gamma$ because of the formulas for $e_d(P)$ and $e_{d-1}(P)$ (see above).

On the other hand, one can choose $\mu_{0,d}(K) = \gamma(K^*)$, where K^* is the dual cone, since it is known that $e_0(P) = 1$. We note that if $\mu(K)$ is an additive measure on polyhedral cones then $\nu(K) = \mu(K^*)$ is also an additive measure on polyhedral cones. Moreover, for integral zonotopes (see Section 7.3), one can always choose $\mu_{k,d}(K_F) = \gamma(K_F^*)$ [BP99]. If F is a k -dimensional face of P then K_F^* is a $(d-k)$ -dimensional cone and $\gamma(K_F^*)$ is understood as the spherical measure in the span of K_F^* .

EULER-MACLAURIN FORMULAS

Let $P \subset \mathbb{R}^d$ be a full-dimensional totally unimodular polytope. Let $\{l_i \mid i = 1, \dots, m\}$ be the set of integral outer normals to the facets of P . We assume that the l_i are primitive, i.e., $\alpha l_i \notin \mathbb{Z}^d$ for any i and any $0 < \alpha < 1$. Say $P = \{x \in \mathbb{R}^d \mid \langle l_i, x \rangle \leq b_i \text{ for } i = 1, \dots, m\}$ for some $b_1, \dots, b_m \in \mathbb{Z}$. Let $h = (h_1, \dots, h_m) \in \mathbb{R}^m$ be a vector. If $\|h\|$ is small enough, then the “perturbed” polytope $P_h = \{x \in \mathbb{R}^d \mid \langle l_i, x \rangle \leq b_i + h_i\}$ has the same “shape” as P and the volume of P_h is a polynomial function of h .

The following expression for the Ehrhart coefficient $e_k(P)$ was found in [KP92]:

$$e_{d-k}(P) = \text{td}_k \left(\frac{\partial}{\partial h_1}, \dots, \frac{\partial}{\partial h_m} \right) \text{vol}_d(P_h) \Big|_{h=0}.$$

Thus $\text{td}_0 = 1$, $\text{td}_1(x_1, \dots, x_m) = (x_1 + \dots + x_m)/2$, etc. The formula can be considered as a far-reaching extension of the classical Euler-Maclaurin formula.

If the polytope is simple, one can formally define

$$b_{d-k}(P) = \text{td}_k \left(\frac{\partial}{\partial h_1}, \dots, \frac{\partial}{\partial h_m} \right) \text{vol}_d(P_h) \Big|_{h=0}.$$

However, $b_{d-k}(P)$ are no longer Ehrhart coefficients if P is not totally unimodular. To get $e_{d-k}(P)$, one should introduce a correction term for each face of codimension $k-1$ of P . When $k=2$, such correction terms have been found by A.G. Khovanskii and J.-M. Kantor. These terms involve Dedekind sums (see [Section 7.3](#)) and they are computable in polynomial time (see [BP99]).

The correction terms for an arbitrary k have been suggested by M. Brion and M. Vergne [BV97a].

MORELLI'S FORMULAS

General formulas for $e_k(P)$ were obtained in [Mor93b]. R. Morelli constructed an explicit measure $\mu_{k,d}(K)$ as in [Theorem 7.4.2](#), which, however, is not a real number but a real-valued rational function on the Grassmannian $\mathbf{G}_{k+1}(\mathbb{R}^d)$ of all $(k+1)$ -dimensional subspaces in \mathbb{R}^d . Let K be a full-dimensional cone whose apex is a k -dimensional subspace (if K is not such a cone then $\mu_{k,d}(K) = 0$). There is an explicit formula for $\mu_{k,d}(K) : \mathbf{G}_{k+1}(\mathbb{R}^d) \rightarrow \mathbb{R}$ when the dual k -dimensional cone $K^* \subset \mathbb{R}^d$ is unimodular. If K^* is not unimodular, then we define $\mu_{k,d}(K)$ using the additivity of $\mu_{k,d}$ (cf. the discussion in [Section 7.3](#) about decomposing a polyhedral cone into unimodular cones). The cone K contains $d-k$ $(k+1)$ -dimensional halfspaces (“edges”) whose intersection is the k -dimensional apex V of K . Let E_s , $s = 1, \dots, d-k$, be the linear spans of these edges. For every s we choose an oriented basis $(b_1^s, \dots, b_{k+1}^s)$ of the $(k+1)$ -dimensional lattice $(E_s \cap \mathbb{Z}^d)$, so that all these orientations are coherent with some fixed orientation of the apex V . Let $A \in \mathbf{G}_{k+1}(\mathbb{R}^d)$ be a $(k+1)$ -dimensional subspace. We define the value of $\mu_{k,d}(K)$ on A as follows: Choose any basis u_1, \dots, u_{k+1} of A . Define a $(k+1) \times (k+1)$ matrix M^s by the formula $M_{ij}^s = \langle b_i^s, u_j \rangle$. Let $f_s = \det M^s$ and define $\mu_{k,d}(K)$ on A to be equal to

$$\frac{\text{td}_{d-k}(f_1, \dots, f_{d-k})}{f_1 \cdots f_{d-k}}.$$

If $d-k$ is fixed then the function $\mu_{k,d}(K) : \mathbf{G}_{k+1}(\mathbb{R}^d) \rightarrow \mathbb{R}$ is polynomially computable. Therefore, computation of any fixed number of the highest Ehrhart coefficients reduces in polynomial time to computation of the volumes of faces for an \mathcal{H} -polytope (see [BP99]).

THE h^* -VECTOR

General properties of generating functions (see [Sta86]) imply that for every integral d -dimensional polytope P there exist integers $h_0^*(P), \dots, h_d^*(P)$ such that

$$\sum_{n=0}^{\infty} E_P(n)x^n = \frac{h_0^*(P) + h_1^*(P)x + \dots + h_d^*(P)x^d}{(1-x)^{d+1}}.$$

The $(d+1)$ -vector $h^*(P) = (h_0^*(P), \dots, h_d^*(P))$ is called the **h^* -vector** of P . It is clear that $h^*(P)$ is a (vector-valued) valuation on the set of integral polytopes and

that $h^*(P)$ is invariant under a unimodular transformation of \mathbb{Z}^d . Moreover, the functions $h_k^*(P)$ constitute a basis of all valuations on integral polytopes that are invariant under unimodular transformations. Unlike the Ehrhart coefficients $e_k(P)$, the numbers $h_k^*(P)$ are not homogeneous. However, $h_k^*(P)$ are monotone (and, therefore, nonnegative): if $Q \subset P$ are two integral polytopes then $h_k^*(P) \geq h_k^*(Q)$ [Sta93]. This property follows from the fact that polytopes admit triangulations that are Cohen-Macaulay complexes (see Chapter 18). If these complexes are Gorenstein then one gets the **Dehn-Sommerville equations** $h_k^*(P) = h_{d-k}^*(P)$. For example, the h^* -vector of the Birkhoff polytope B_n (see Section 7.1) satisfies the Dehn-Sommerville equations (see [Sta83]).

In principle, there is a combinatorial way to calculate $h^*(P)$. Namely, let Δ be a triangulation of P such that every d -dimensional simplex of Δ is integral and has volume $1/d!$ (see Section 7.2). Let $f_k(\Delta)$ be the number of k -dimensional faces of the triangulation Δ . Then

$$h_k^*(P) = \sum_{i=0}^k (-1)^{k-i} \binom{d-i}{d-k} f_{i-1}(\Delta),$$

where we let $f_{-1}(\Delta) = 1$. Such a triangulation may not exist for the polytope P but it exists for mP , where m is a sufficiently large integer (see [KKMS73]). Generally, this triangulation Δ would be too big, but for some special polytopes with nice structure (for example, for the so-called *poset polytopes*) it may provide a very good way to compute $h^*(P)$ and hence the Ehrhart polynomial E_P .

Since the number of integral points in a polytope is a valuation, we get the following result proved by P. McMullen (see [McM93]).

THEOREM 7.4.3

Let P_1, \dots, P_m be integral polytopes in \mathbb{R}^d . For an m -tuple of natural numbers $\mathbf{n} = (n_1, \dots, n_m)$, let us define the polytope

$$P(\mathbf{n}) = \{n_1 x_1 + \dots + n_m x_m \mid x_1 \in P_1, \dots, x_m \in P_m\}$$

(using “+” for Minkowski addition one can also write $P(\mathbf{n}) = n_1 P_1 + \dots + n_m P_m$). Then there exists a polynomial $p(x_1, \dots, x_m)$ of degree at most d such that

$$|P(\mathbf{n}) \cap \mathbb{Z}^d| = p(n_1, \dots, n_m).$$

An interpretation of the values $p(n_1, \dots, n_m)$ for nonpositive integer values of n_1, \dots, n_m can be obtained by using the polytope algebra identities (see [McM93]).

More generally, the existence of local formulas for the Ehrhart coefficients implies that the number of integral points in an integral polytope $P_h = \{x \in \mathbb{R}^d \mid Ax \leq b + h\}$ is a polynomial in h provided P_h is an integral polytope combinatorially isomorphic to the integral polytope P_0 . In other words, if we move the facets of an integral polytope so that it remains integral and has the same facial structure, then the number of integral points varies polynomially.

INTEGRAL POINTS IN RATIONAL POLYTOPES

If P is a rational (not necessarily integral) polytope then $|nP \cap \mathbb{Z}^d|$ is not a polynomial but a **quasipolynomial** (a function of n whose value cycles through the values

of a finite list of polynomials). The following result was independently proven by P. McMullen and R. Stanley (see [McM93] and [Sta86]).

THEOREM 7.4.4

Let $P \subset \mathbb{R}^d$ be a rational polytope. For every r , $0 \leq r \leq d$, let ind_r be the smallest natural number k such that all r -dimensional faces of kP are integral polytopes. Then, for every $n \in \mathbb{N}$,

$$|nP \cap \mathbb{Z}^d| = \sum_{r=0}^d e_r(P, n(\text{mod } \text{ind}_r)) \cdot n^r$$

for suitable rational numbers $e_r(P, k)$, $0 \leq k < \text{ind}_r$.

P. McMullen also obtained a generalization of the “reciprocity law” (see [Sta86] and [McM93]).

Let us fix an $n \times d$ integer matrix A such that the set $P_b = \{x \mid Ax \leq b\}$, $b \in \mathbb{Z}^n$, if nonempty, is a rational polytope. Let $B \subset \mathbb{Z}^n$ be a set of right-hand-side vectors b such that the combinatorial structure of P_b is the same for all $b \in B$. In [BP99] it is shown that as long as the dimension d is fixed, one can find a polynomially computable formula $F(b)$ for the number $|P_b \cap \mathbb{Z}^d|$, where F is a polynomial of degree d in integer parts of linear functions of b . It is based on Brion’s Theorem (Theorem 7.3.2) and the “rounding” of rational translations of unimodular cones.

Interestingly, for a “typical” (and, therefore, nonrational) polytope P the difference $|tP \cap \mathbb{Z}^d| - t^d \text{vol } P$ has order $O((\ln t)^{d-1+\epsilon})$ as $t \rightarrow +\infty$ [Skr98].

7.5 PROBLEMS WITH QUANTIFIERS

A natural generalization of the decision problem (see Section 7.2) is a problem with quantifiers. We describe some known results and formulate open questions for this class of problems.

FROBENIUS PROBLEM

The most famous problem from this class is the *Frobenius problem*:

Given k positive integers a_1, \dots, a_k with greatest common divisor 1, find the largest integer m that cannot be represented as an integer combination $a_1n_1 + \dots + a_kn_k$, $n_i \geq 0$.

The problem is known to be NP-hard in general, but a polynomial time algorithm is known for fixed k [Kan92].

PROBLEM WITH QUANTIFIERS

A general *problem with quantifiers* can be formulated as follows. Suppose that P is a Boolean combination of convex polyhedra: we start with some polyhedra $P_1, \dots, P_k \subset \mathbb{R}^d$ given by their facet descriptions and construct P by using the set-theoretical operations of union, intersection, and complement. We want to find

out if the formula

$$\exists x_1 \forall x_2 \exists x_3 \dots \forall x_m : (x_1, \dots, x_m) \in P \quad (7.5.1)$$

is true. Here x_i is an integral vector from \mathbb{Z}^{d_i} , and, naturally, $d_1 + \dots + d_m = d$, $d_i \geq 0$. The parameters that characterize the size of (7.5.1) can be divided into two classes. The first class consists of the parameters characterizing the *combinatorial size* of the formula. These are the dimension d , the number $m - 1$ of quantifier alternations, the number of linear inequalities and Boolean operations that define the polyhedral set P . The parameters from the other class characterize the *numerical size* of the formula. Those are the bit sizes of the numbers involved in the inequalities that define P .

The following fundamental question remains open.

PROBLEM 7.5.1

Let us fix all the combinatorial parameters of the formula (7.5.1). Does there exist a polynomial time algorithm that checks whether this formula is true?

Naturally, “polynomial time” means that the running time of the algorithm is bounded by a polynomial in the numerical size of the formula. The answer to this question is unknown although it is widely believed that such an algorithm indeed exists. A polynomial time algorithm is known if the formula contains not more than 1 quantifier alternation, i.e., if $m \leq 2$ [Kan90]. A related problem is to compute the number of solutions for quantifier-free variables in a formula with quantifiers.

Sets of lattice points described by formulas with existential quantifiers only are studied in [BW03]. Geometrically, such a set S can be viewed as a projection of the set of lattice points in a polyhedron P . Examples include lattice semigroups, (minimal) Hilbert bases of rational cones, and test sets in integer programming. It is shown that if P is bounded and the dimension of P is fixed then the exponential sum over S admits a short (polynomially computable) formula. As a corollary, various counting problems for lattice semigroups, Hilbert bases, and test sets admit polynomial time algorithms in fixed dimension. For a structural theory of lattice semigroups see [K95].

7.6 SOURCES AND RELATED MATERIAL

RELATED CHAPTERS

[Chapter 3: Tilings](#)

[Chapter 16: Basic properties of convex polytopes](#)

[Chapter 17: Subdivisions and triangulations of polyhedra](#)

[Chapter 31: Computational convexity](#)

[Chapter 46: Mathematical programming](#)

REFERENCES

- [AHU74] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, 1974.

- [BLPS99] W. Banaszczyk, A.E. Litvak, A. Pajor, and S.J. Szarek. The flatness theorem for nonsymmetric convex bodies via the local theory of Banach spaces. *Math. Oper. Res.*, 24:728–750, 1999.
- [BP99] A.I. Barvinok and J.E. Pommersheim. An algorithmic theory of lattice points in polyhedra. In *New Perspectives in Algebraic Combinatorics (Berkeley, 1996–97)*, volume 38 of *Math. Sci. Res. Inst. Publ.*, pages 91–147. Cambridge Univ. Press, 1999.
- [BS01] A. Barvinok and A. Samorodnitsky. The distance approach to approximate combinatorial counting. *Geom. Funct. Anal.*, 11:871–899, 2001.
- [BW03] A. Barvinok and K. Woods. Short rational generating functions for lattice point problems. *J. Amer. Math. Soc.*, 16:957–979, 2003.
- [BS96] L.J. Billera and A. Sarangarajan. Combinatorics of permutation polytopes. In L.J. Billera, C. Greene, R. Simion, and R. Stanley, editors, *Formal Power Series and Algebraic Combinatorics*, volume 24 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–23. American Math. Soc., Providence, 1996.
- [BV92] I. Bárány and A.M. Vershik. On the number of convex lattice polytopes. *Geom. Funct. Anal.*, 2:381–393, 1992.
- [BZ88] A.D. Berenstein and A.V. Zelevinsky. Tensor product multiplicities and convex polytopes in the partition space. *J. Geom. Phys.*, 5:453–472, 1988.
- [BP02] M. Beck and D. Pixton. The Ehrhart polynomial of the Birkhoff polytope. *Math ArXiv preprint*, math.CO/0202267, 2002.
- [Bri98] M. Brion. Points entiers dans les polyèdres convexes. *Ann. Sci. Ècole Norm. Sup. (4)*, 21:653–663, 1998.
- [BV97a] M. Brion and M. Vergne. Lattice points in simple polytopes. *J. Amer. Math. Soc.*, 10:371–392, 1997.
- [BV97b] M. Brion and M. Vergne. Residue formulae, vector partition functions and lattice points in rational polytopes. *J. Amer. Math. Soc.*, 10:797–833, 1997.
- [CHKM92] W.J. Cook, M. Hartmann, R. Kannan, and C. McDiarmid. On integer points in polyhedra. *Combinatorica*, 12:27–37, 1992.
- [Cor01] G. Cornuéjols. *Combinatorial Optimization. Packing and Covering*. CBMS-NSF Regional Conference Series in Applied Mathematics, volume 74. SIAM, Philadelphia, 2001.
- [CS94] S.E. Cappell and J.L. Shaneson. Genera of algebraic varieties and counting of lattice points. *Bull. Amer. Math. Soc. (N.S.)*, 30:62–69, 1994.
- [DL97] M.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Volume 15 of *Algorithms Combin.*, Springer-Verlag, Berlin, 1997.
- [DLS03] J.A. De Loera and B. Sturmfels. Algebraic unimodular counting. *Math. Program.*, 96:183–203, 2003.
- [DR97] R. Diaz and S. Robins. The Ehrhart polynomial of a lattice polytope. *Ann. of Math.*, 145:503–518, 1997; Erratum, 146:237, 1997.
- [EKK84] V.A. Emelichev, M.M. Kovalev, and M.K. Kravtsov. *Polytopes, Graphs and Optimization*. Cambridge University Press, 1984.
- [GL87] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. North Holland, Amsterdam, 2nd edition, 1987.
- [GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.
- [GW93] P. Gritzmann and J.M. Wills. Lattice points. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, pages 765–797. Elsevier, Amsterdam, 1993.
- [Hås88] J. Håstad. Dual vectors and lower bounds for the nearest lattice point problem. *Combinatorica*, 8:75–81, 1988.

- [JS97] M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method: an approach to approximate counting and integration. In D.S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 482–520. PWS, Boston, 1997.
- [JSV01] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *Proc. 33d Annu. ACM Symp. Theory Comput.*, pages 712–721, 2001.
- [Kan90] R. Kannan. Test sets for integer programs, $\forall\exists$ sentences. In W. Cook and P.D. Seymour, editors, *Polyhedral Combinatorics*, volume 1 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 39–47. Amer. Math. Soc., Providence, 1990.
- [Kan92] R. Kannan. Lattice translates of a polytope and the Frobenius problem. *Combinatorica*, 12:161–177, 1992.
- [Kan99] J.-M. Kantor. On the width of lattice-free simplices. *Compositio Math.*, 118:235–241, 1999.
- [K95] A.G. Khovanskii. Sums of finite sets, orbits of commutative semigroups and Hilbert functions (Russian). *Funktional. Anal. i Prilozhen.*, 29:36–50, 1995. Translated in *Funct. Anal. Appl.*, 29:102–112, 1995.
- [KKMS73] G. Kempf, F.F. Knudsen, D. Mumford, and B. Saint-Donat. *Toroidal Embeddings I. Lecture Notes in Math.*, volume 339, Springer-Verlag, Berlin-New York, 1973.
- [KP92] A.G. Khovanskii and A.V. Pukhlikov. A Riemann-Roch theorem for integrals and sums of quasipolynomials on virtual polytopes (Russian). *Algebra i Analiz*, 4:188–216, 1992. Translated in *St.-Petersb. Math. J.*, 4:789–812, 1993.
- [Lag95] J.C. Lagarias. Point lattices. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 919–966. North Holland, Amsterdam, 1995.
- [Law91] J. Lawrence. Rational-function-valued valuations on polyhedra. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 199–208, volume 6 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.* Amer. Math. Soc., Providence, 1991.
- [LS92] L. Lovász and H.E. Scarf. The generalized basis reduction algorithm. *Math. Oper. Res.*, 17:751–764, 1992.
- [McM93] P. McMullen. Valuations and dissections. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, volume B, pages 933–988. North-Holland, Amsterdam, 1993.
- [Mor93a] R. Morelli. A theory of polyhedra. *Adv. Math.*, 97:1–73, 1993.
- [Mor93b] R. Morelli. Pick’s theorem and the Todd class of a toric variety. *Adv. Math.*, 100:183–231, 1993.
- [MS99] B. Morris and A. Sinclair. Random walks on truncated cubes and sampling 0-1 knapsack solutions. In *Proc. 40th IEEE Symp. on Foundations of Computer Science*, 230–240, 1999.
- [Oda88] T. Oda. *Convex Bodies and Algebraic Geometry: An Introduction to the Theory of Toric Varieties*. Springer-Verlag, Berlin, 1988.
- [Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes. Lecture Notes in Math.*, volume 1643, Springer-Verlag, Berlin, 1996.
- [Sca85] H.E. Scarf. Integral polyhedra in three space. *Math. Oper. Res.*, 10:403–438, 1985.
- [Sch86] A. Schrijver. *The Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.
- [Skr98] M.M. Skriganov. Ergodic theory on $\text{SL}(n)$, Diophantine approximations and anomalies in the lattice point problem. *Invent. Math.*, 132:1–72, 1998.
- [Sta83] R.P. Stanley. *Combinatorics and Commutative Algebra*, volume 41 of *Progress in Mathematics*. Birkhäuser, Boston, 1983.

- [Sta86] R.P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth and Brooks/Cole, Monterey, 1986.
- [Sta93] R.P. Stanley. A monotonicity property of h -vectors and h^* -vectors. *European J. Combin.*, 14:251–258, 1993.
- [Zie00] G.M. Ziegler. Lectures on 0/1-polytopes. In G. Kalai and G.M. Ziegler, editors, *Polytopes—Combinatorics and Computation (Oberwolfach, 1997)*, pages 1–41, DMV Sem., volume 29, Birkhäuser, Basel, 2000.

8 LOW-DISTORTION EMBEDDINGS OF FINITE METRIC SPACES

Piotr Indyk and Jiří Matoušek

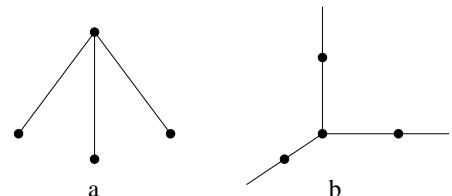
INTRODUCTION

An n -point metric space (X, D) can be represented by an $n \times n$ table specifying the distances. Such tables arise in many diverse areas. For example, consider the following scenario in microbiology: X is a collection of bacterial strains, and for every two strains, one is given their *dissimilarity* (computed, say, by comparing their DNA). It is difficult to see any structure in a large table of numbers, and so we would like to represent a given metric space in a more comprehensible way.

For example, it would be very nice if we could assign to each $x \in X$ a point $f(x)$ in the plane in such a way that $D(x, y)$ equals the Euclidean distance of $f(x)$ and $f(y)$. Such a representation would allow us to see the structure of the metric space: tight clusters, isolated points, and so on. Another advantage would be that the metric would now be represented by only $2n$ real numbers, the coordinates of the n points in the plane, instead of $\binom{n}{2}$ numbers as before. Moreover, many quantities concerning a point set in the plane can be computed by efficient geometric algorithms, which are not available for an arbitrary metric space.

This sounds too good to be generally true: indeed, there are even finite metric spaces that cannot be exactly represented either in the plane or in *any* Euclidean space; for instance, the four vertices of the graph $K_{1,3}$ (a star with 3 leaves) with the shortest-path metric (see Figure 8.0.1a). However, it *is* possible to embed the latter metric in a Euclidean space, if we allow the distances to be distorted somewhat. For example, if we place the center of the star at the origin in \mathbb{R}^3 and the leaves at $(1, 0, 0), (0, 1, 0), (0, 0, 1)$, then all distances are preserved *approximately*, up to a factor of $\sqrt{2}$ (Figure 8.0.1b).

FIGURE 8.0.1
A nonembeddable metric space.



Approximate embeddings have proven extremely helpful for approximate solutions of problems dealing with distances. For many important algorithmic problems, they yield the only known good approximation algorithms.

The normed spaces usually considered for embeddings of finite metrics are the spaces ℓ_p^d , $1 \leq p \leq \infty$, and the cases $p = 1, 2, \infty$ play the most prominent roles.

GLOSSARY

Metric space: A pair (X, D) , where X is a set of *points* and $D: X \times X \rightarrow [0, \infty)$ is a *distance function* satisfying the following conditions for all $x, y, z \in X$:

- (i) $D(x, y) = 0$ if and only if $x = y$,
- (ii) $D(x, y) = D(y, x)$ (symmetry), and
- (iii) $D(x, y) + D(y, z) \geq D(x, z)$ (triangle inequality).

Separable metric space: A metric space (X, D) containing a countable dense set; that is, a countable set Y such that for every $x \in X$ and every $\varepsilon > 0$ there exists $y \in Y$ with $D(x, y) < \varepsilon$.

Pseudometric: Like metric except that (i) is not required.

Isometry: A mapping $f: X \rightarrow X'$, where (X, D) and (X', D') are metric spaces, with $D'(f(x), f(y)) = D(x, y)$ for all x, y .

(Real) normed space: A real vector space Z with a mapping $\|\cdot\|_Z: Z \rightarrow [0, \infty]$, the *norm*, satisfying $\|x\|_Z = 0$ iff $x = 0$, $\|\alpha x\|_Z = |\alpha| \cdot \|x\|_Z$ ($\alpha \in \mathbb{R}$), and $\|x + y\|_Z \leq \|x\|_Z + \|y\|_Z$. The metric on Z is given by $(x, y) \mapsto \|x - y\|_Z$.

ℓ_p^d : The space \mathbb{R}^d with the ℓ_p -norm $\|x\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$, $1 \leq p \leq \infty$ (where $\|x\|_\infty = \max_i |x_i|$).

Finite ℓ_p metric: A finite metric space isometric to a subspace of ℓ_p^d for some d .

ℓ_p : For a sequence (x_1, x_2, \dots) of real numbers we set $\|x\|_p = (\sum_{i=1}^\infty |x_i|^p)^{1/p}$. Then ℓ_p is the space consisting of all x with $\|x\|_p < \infty$, equipped with the norm $\|\cdot\|_p$. It contains every finite ℓ_p metric as a (metric) subspace.

Distortion: A mapping $f: X \rightarrow X'$, where (X, D) and (X', D') are metric spaces, is said to have distortion at most c , or to be a *c-embedding*, where $c \geq 1$, if there is an $r \in (0, \infty)$ such that for all $x, y \in X$,

$$r \cdot D(x, y) \leq D'(f(x), f(y)) \leq cr \cdot D(x, y).$$

If X' is a normed space, we usually require $r = \frac{1}{c}$ or $r = 1$.

Order of congruence: A metric space (X, D) has order of congruence at most m if every finite metric space that is not isometrically embeddable in (X, D) has a subspace with at most m points that is not embeddable in (X, D) .

8.1 THE SPACES ℓ_p

8.1.1 THE EUCLIDEAN SPACES ℓ_2^d

Among normed spaces, the Euclidean spaces are the most familiar, the most symmetric, the simplest in many respects, and the most restricted. Every finite ℓ_2 metric embeds isometrically in ℓ_p for all p . More generally, we have the following Ramsey-type result on the “universality” of ℓ_2 ; see, e.g., [MS86]:

THEOREM 8.1.1 *Dvoretzky's theorem (a finite quantitative version)*

For every d and every $\varepsilon > 0$ there exists $n = n(d, \varepsilon) \leq 2^{O(d/\varepsilon^2)}$ such that ℓ_2^d can be $(1+\varepsilon)$ -embedded in every n -dimensional normed space.

Isometric embeddability in ℓ_2 has been well understood since the classical works of Menger, von Neumann, Schoenberg, and others (see, e.g., [Sch38]). Here is a brief summary:

THEOREM 8.1.2

- (i) (*Compactness*) A separable metric space (X, D) is isometrically embeddable in ℓ_2 iff each finite subspace is so embeddable.
- (ii) (*Order of congruence*) A finite (or separable) metric space embeds isometrically in ℓ_2^d iff every subspace of at most $d+3$ points so embeds.
- (iii) For a finite $X = \{x_0, x_1, \dots, x_n\}$, (X, D) embeds in ℓ_2 iff the $n \times n$ matrix $(D(x_0, x_i)^2 + D(x_0, x_j)^2 - D(x_i, x_j)^2)_{i,j=1}^n$ is positive semidefinite; moreover, its rank is the smallest dimension for such an embedding.
- (iv) (*Schoenberg's criterion*) A separable (X, D) isometrically embeds in ℓ_2 iff the matrix $(e^{-\lambda D(x_i, x_j)^2})_{i,j=1}^n$ is positive semidefinite for all $n \geq 1$, for any points $x_1, x_2, \dots, x_n \in X$, and for any $\lambda > 0$. (This is expressed by saying that the functions $x \mapsto e^{-\lambda x^2}$, for all $\lambda > 0$, are positive definite on ℓ_2 .)

Using similar ideas, the problem of finding the smallest c such that a given finite (X, D) can be c -embedded in ℓ_2 can be formulated as a semidefinite programming problem and thus solved in polynomial time [LLR95] (but no similar result is known for embedding in ℓ_2^d with d given!).

8.1.2 THE SPACES ℓ_1^d

GLOSSARY

Cut metric: A pseudometric D on a set X such that, for some partition $X = A \cup B$, we have $D(x, y) = 0$ if both $x, y \in A$ or both $x, y \in B$, and $D(x, y) = 1$ otherwise.

Hypermetric inequality: A metric space (X, D) satisfies the $(2k+1)$ -point hypermetric inequality (also called the $(2k+1)$ -gonal inequality) if for every multiset A of k points and every multiset B of $k+1$ points in X , $\sum_{a, a' \in A} D(a, a') + \sum_{b, b' \in B} D(b, b') \leq \sum_{a \in A, b \in B} D(a, b)$. (We get the triangle inequality for $k = 1$.)

Hypermetric space: A space that satisfies the hypermetric inequality for all k .

Cocktail-party graph: The complement of a perfect matching in a complete graph K_{2m} ; also called a **hyperoctahedron graph**.

Half-cube graph: The vertex set consists of all vectors in $\{0, 1\}^n$ with an even number of 0's, and edges connect vectors with Hamming distance 2.

Cartesian product of graphs G and H : The vertex set is $V(G) \times V(H)$, and the edge set is $\{\{(u, v), (u, v')\} \mid u \in V(G), \{v, v'\} \in E(H)\} \cup \{\{(u, v), (u', v)\} \mid \{u, u'\} \in E(G), v \in V(H)\}$. The cubes are Cartesian powers of K_2 .

Girth of a graph: The length of the shortest cycle.

The ℓ_1 spaces are important for many reasons, but considerably more complicated than Euclidean spaces; a general reference here is [DL97]. Many important and challenging open problems are related to embeddings in ℓ_1 or in ℓ_1^d .

Unlike the situation in ℓ_2^n , not every n -point ℓ_1 -metric lives in ℓ_1^n ; dimension of order n^2 is sometimes necessary and always sufficient to embed n -point ℓ_1 -metrics isometrically (similarly for the other ℓ_p -metrics with $p \neq 2$).

The ℓ_1 metrics on an n -point set X are precisely the elements of the *cut cone*; that is, linear combinations with nonnegative coefficients of cut metrics on X . Another characterization is this: A metric D on $\{1, 2, \dots, n\}$ is an ℓ_1 metric iff there exist a measure space (Ω, Σ, μ) and sets $A_1, \dots, A_n \in \Sigma$ such that $D(i, j) = \mu(A_i \Delta A_j)$.

Every ℓ_1 metric is a hypermetric space (since cut metrics satisfy the hypermetric inequalities), but for 7 or more points, this condition is not sufficient. Hypermetric spaces have an interesting characterization in terms of *Delaunay polytopes* of lattices; see [DL97].

ISOMETRIC EMBEDDABILITY

Deciding isometric embeddability in ℓ_1 is NP-hard. On the other hand, the embeddability of *unweighted* graphs, both in ℓ_1 and in a Hamming cube, has been characterized and can be tested in polynomial time. In particular, we have:

THEOREM 8.1.3

- (i) An unweighted graph G embeds isometrically in some cube $\{0, 1\}^m$ with the ℓ_1 -metric iff it is bipartite and satisfies the pentagonal inequality.
- (ii) An unweighted graph G embeds isometrically in ℓ_1 iff it is an isometric subgraph of a Cartesian product of half-cube graphs and cocktail-party graphs.

A first characterization of cube-embeddable graphs was given by Djokovic [Djo73], and the form in (i) is due to Avis (see [DL97]). Part (ii) is from Shpectorov [Shp93].

ORDER OF CONGRUENCE

The isometric embeddability in ℓ_1^2 is characterized by 6-point subspaces (6 is best possible here), and can thus be tested in polynomial time (Bandelt and Chepoi [BC96]). The proof uses a result of Bandelt and Dress [BD92] of independent interest, about certain canonical decompositions of metric spaces (see also [DL97]).

On the other hand, for no $d \geq 3$ it is known whether the order of congruence of ℓ_1^d is finite; there is a lower bound of d^2 (for odd d) or $d^2 - 1$ (for d even).

8.1.3 THE OTHER p

The spaces ℓ_∞^d are the richest (and thus generally the most difficult to deal with); every n -point metric space (X, D) embeds isometrically in ℓ_∞^n . To see this, write $X = \{x_1, x_2, \dots, x_n\}$ and define $f: X \rightarrow \ell_\infty^n$ by $f(x_i)_j = D(x_i, x_j)$.

The other $p \neq 1, 2, \infty$ are encountered less often, but it may be useful to know the cases where all ℓ_p metrics embed with bounded distortion in ℓ_q : This happens iff $p = q$, or $p = 2$, or $q = \infty$, or $1 \leq q \leq p \leq 2$. Isometric embeddings exist in all these cases. Moreover, for $1 \leq q \leq p \leq 2$, the whole of ℓ_p^d can be $(1 + \varepsilon)$ embedded

in ℓ_q^{Cd} with a suitable $C = C(p, q, \varepsilon)$ (so the dimension doesn't grow by much); see, e.g., [MS86]. These embeddings are probabilistic. The simplest one is $\ell_2^d \rightarrow \ell_1^{Cd}$, given by $x \mapsto Ax$ for a random ± 1 matrix A of size $Cd \times d$ (surprisingly, no good explicit embedding is known even in this case).

8.2 APPROXIMATE EMBEDDINGS OF GENERAL METRICS IN ℓ_p

8.2.1 BOURGAIN'S EMBEDDING IN ℓ_2

The mother of most embeddings mentioned in the next few sections, from both historical and “technological” points of view, is the following theorem.

THEOREM 8.2.1 *Bourgain* [Bou85]

Any n -point metric space (X, D) can be embedded in ℓ_2 (in fact, in every ℓ_p) with distortion $O(\log n)$.

We describe the embedding, which is constructed probabilistically. We set $m = \lfloor \log_2 n \rfloor$ and $q = \lfloor C \log n \rfloor$ (C a suitable constant) and construct an embedding in ℓ_2^{mq} , with the coordinates indexed by $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, q$. For each such i, j , we select a subset $A_{ij} \subseteq X$ by putting each $x \in X$ into A_{ij} with probability 2^{-j} , all the random choices being mutually independent. Then we set $f(x)_{ij} = D(x, A_{ij})$. We thus obtain an embedding in $\ell_2^{O(\log^2 n)}$ (Bourgain's original proof used exponential dimension; the possibility of reducing it was noted later), and it can be shown that the distortion is $O(\log n)$ with high probability.

This yields an $O(n^2 \log n)$ randomized algorithm for computing the desired embedding. The algorithm can be derandomized (preserving the polynomial time and the dimension bound) using the method of conditional probabilities; this result seems to be folklore. Alternatively, it can be derandomized using small sample spaces [LLR95]; this, however, uses dimension $\Theta(n^2)$. Finally, as was remarked above, an embedding of a given space in ℓ_2 with optimal distortion can be computed by semidefinite programming.

The $O(\log n)$ distortion for embedding a general metric in ℓ_2 is tight [LLR95] (and similarly for ℓ_p , $p < \infty$ fixed). Examples of metrics that cannot be embedded any better are the shortest-path metrics of constant-degree expanders. (An n -vertex graph is a **constant-degree expander** if all degrees are bounded by some constant r and each subset of k vertices has at least βk outgoing edges, for $1 \leq k \leq \frac{n}{2}$ and for some constant $\beta > 0$ independent of n .)

Another interesting lower bound is due to Linial et al. [LMN02]: The shortest-path metric of *any* k -regular graph ($k \geq 3$) of girth g requires $\Omega(\sqrt{g})$ distortion for embedding in ℓ_2 .

8.2.2 THE DIMENSION OF EMBEDDINGS IN ℓ_∞

If we want to embed all n -point metrics in ℓ_∞^d , there is a tradeoff between the dimension d and the worst-case distortion. The following result was proved in [Mat96] by adapting Bourgain's technique.

THEOREM 8.2.2

For an integer $b > 0$ set $c = 2b - 1$. Then any n -point metric space can be embedded in ℓ_∞^d with distortion c , where $d = O(bn^{1/b} \log n)$.

An almost matching lower bound can be proved using graphs without short cycles, an idea also going back to [Bou85]. Let $m(g, n)$ be the maximum possible number of edges of an n -vertex graph of girth $g + 1$. For every fixed $c \geq 1$ and integer $g > c$ there exists an n -point metric space such that any c -embedding in ℓ_∞^d has $d = \Omega(m(g, n)/n)$ [Mat96]. The proof goes by counting: Fix a graph G_0 witnessing $m(g, n)$, and let \mathcal{G} be the set of graphs (considered with the shortest-path metric) that can be obtained from G_0 by deleting some edges. It turns out that if $G, G' \in \mathcal{G}$ are distinct, then they cannot have “essentially the same” c -embeddings in ℓ_∞^d , and there are only “few” essentially different embeddings in ℓ_∞^d if d is small.

It is easy to show that $m(g, n) = O(n^{1+1/\lfloor g/2 \rfloor})$ for all g , and this is conjectured to be the right order of magnitude [Erd64]. This has been verified for $g \leq 7$ and for $g = 10, 11$, while only worse lower bounds are known for the other values of g (with exponent roughly $1 + 4/3g$ for g large). Whenever the conjecture holds for some $g = 2b - 1$, the above theorem is tight up to a logarithmic factor for the corresponding b . Unfortunately, although explicit constructions of graphs of a given girth with many edges are known, the method doesn’t provide explicit examples of badly embeddable spaces.

DISTANCE ORACLES

An interesting algorithmic result, conceptually resembling the above theorem, was obtained by Thorup and Zwick [TZ01]. They showed that for an integer $b > 0$, every n -point metric space can be stored in a data structure of size $O(n^{1+1/b})$ (with preprocessing time of the same order) so that, within time $O(b)$, the distance between any two points can be approximated within a multiplicative factor of $2b - 1$.

LOW DIMENSION

The other end of the tradeoff between distortion and dimension d , where d is fixed (and then all ℓ_p -norms on \mathbb{R}^d are equivalent up to a constant) was investigated in [Mat90]. For all fixed $d \geq 1$, there are n -point metric spaces requiring distortion $\Omega(n^{1/\lfloor(d+1)/2\rfloor})$ for embedding in ℓ_2^d (for $d = 2$, an example is the shortest-path metric of K_5 with every edge subdivided $n/10$ times). On the other hand, every n -point space $O(n)$ -embeds in ℓ_2^1 (the real line), and $O(n^{2/d} \log^{3/2} n)$ -embeds in ℓ_2^d , $d \geq 3$.

8.2.3 THE JOHNSON-LINDENSTRAUSS LEMMA: FLATTENING IN ℓ_2

The n -point ℓ_2 metric with all distances equal to 1 requires dimension $n - 1$ for isometric embedding in ℓ_2 . A somewhat surprising and extremely useful result shows that, in particular, this metric can be embedded in dimension only $O(\log n)$ with distortion close to 1.

THEOREM 8.2.3 Johnson and Lindenstrauss [JL84]

For every $\varepsilon > 0$, any n -point ℓ_2 metric can be $(1+\varepsilon)$ -embedded in $\ell_2^{O(\log n/\varepsilon^2)}$.

There is an almost matching lower bound for the necessary dimension, due to Alon (see [Mat02a]): $\Omega(\log n / (\varepsilon^2 \log(1/\varepsilon)))$.

All known proofs (see, e.g., [Ach01] for references and an insightful discussion) first place the metric under consideration in ℓ_2^n and then map it into ℓ_2^d by a random linear map $A: \ell_2^n \rightarrow \ell_2^d$. Here A can be a random orthogonal projection (as in [JL84]). It can also be given by a random $n \times d$ matrix with independent $N(0, 1)$ entries [IM98], or even one with independent uniform random ± 1 entries. The proof in the last case, due to [Ach01], is considerably more difficult than the previous ones (which use spherically symmetric distributions), but this version has advantages in applications.

An embedding as in the theorem can be computed deterministically in time $O(n^2 d (\log n + 1/\varepsilon)^{O(1)})$ [EIO02] (also see [Siv02]).

Brinkman and Charikar [BC03] proved that no flattening lemma of comparable strength holds in ℓ_1 . Namely, for every fixed $c > 1$, and every n , they exhibit an n -point ℓ_1 -metric that cannot be c -embedded into ℓ_1^d unless $d = n^{\Omega(1/c^2)}$. A simpler alternative proof was found later by Lee and Naor (manuscript).

In contrast, [Ind00] showed that for every $0 < \varepsilon < 1$ and any ℓ_1 -metric over $X \subset \ell_1^d$, there is a $k \times d$ real matrix $[a_1 \dots a_k]^T$, $k = O(\log |X|/\varepsilon^2)$, such that for any $p, q \in X$, $\|p - q\|_1 \leq \text{median}(|a_1(p - q)|, \dots, |a_k(p - q)|) \leq (1 + \varepsilon)\|p - q\|_1$.

8.2.4 VOLUME-RESPECTING EMBEDDINGS

Feige [Fei00] introduced the notion of *volume-respecting* embeddings in ℓ_2 , with impressive algorithmic applications. While the distortion of a mapping depends only on pairs of points, the volume-respecting condition takes into account the behavior of k -tuples. For an arbitrary k -point metric space (S, D) , we set $\text{Vol}(S) = \sup_{\text{nonexpanding } f: S \rightarrow \ell_2} \text{Evol}(f(S))$, where $\text{Evol}(P)$ is the $(k-1)$ -dimensional volume of the convex hull of P (in ℓ_2). Given a nonexpanding $f: X \rightarrow \ell_2$ for some metric space (X, D) with $|X| \geq k$, we define the k -distortion of f to be

$$\sup_{S \subseteq X, |S|=k} \left(\frac{\text{Vol}(S)}{\text{Evol}(f(S))} \right)^{1/(k-1)}$$

If the k -distortion of f is Δ , we call f (k, Δ) -volume-respecting.

If $f: X \rightarrow \ell_2$ is an embedding scaled so that it is nonexpanding but just so, the 2-distortion coincides with the usual distortion. But note that for $k > 2$, the isometric “straight” embedding of a path in ℓ_2 is not volume-respecting at all. In fact, it is known that for any $k > 2$, no $(k, o(\sqrt{\log n}))$ -volume-respecting embedding of a line exists [DV01].

Extending Bourgain’s technique, Feige proved that for every $k > 2$, every n -point metric space has a $(k, O(\log n + \sqrt{k \log n \log k}))$ -volume-respecting embedding in ℓ_2 .

8.3 APPROXIMATE EMBEDDING OF SPECIAL METRICS IN ℓ_p

GLOSSARY

\mathcal{G} -metric: Let \mathcal{G} be a class of graphs and let $G \in \mathcal{G}$. Each positive weight function $w: E(G) \rightarrow (0, \infty)$ defines a metric D_w on $V(G)$, namely, the shortest-

path metric, where the length of a path is the sum of the weights of its edges. A metric space is a \mathcal{G} -metric if it is isometric to a subspace of $(V(G), D_w)$ for some $G \in \mathcal{G}$ and some w .

Tree metric, planar-graph metric: A \mathcal{G} -metric for \mathcal{G} , the class of all trees or all planar graphs, respectively.

Minor: A graph G is a minor of a graph H if it can be obtained from H by repeated deletions of edges and contractions of edges.

8.3.1 TREE METRICS, PLANAR-GRAH METRICS, AND FORBIDDEN MINORS

A major research direction has been improving Bourgain's embedding in ℓ_2 for restricted families of metric spaces.

TREE METRICS

It is easy to show that any tree metric embeds isometrically in ℓ_1 . Any n -point tree metric can also be embedded isometrically in $\ell_\infty^{O(\log n)}$ [LLR95]. For ℓ_p embeddings, the situation is rather delicate:

THEOREM 8.3.1

Distortion of order $(\log \log n)^{\min(1/2, 1/p)}$ is sufficient for embedding any n -vertex tree metric in ℓ_p ($p \in (1, \infty)$ fixed) [Mat99], and it is also necessary in the worst case (for the complete binary tree; [Bou86]).

Gupta [Gup00] proved that any n -point tree metric $O(n^{1/(d-1)})$ -embeds in ℓ_2^d ($d \geq 1$ fixed), and for $d = 2$ and trees with unit-length edges, Babilon et al. [BMMV02] improved this to $O(\sqrt{n})$.

PLANAR-GRAH METRICS AND OTHER CLASSES WITH A FORBIDDEN MINOR

The following result was proved by Rao, building on the work of Klein, Plotkin, and Rao.

THEOREM 8.3.2 Rao [Rao99]

Any n -point planar-graph metric can be embedded in ℓ_2 with distortion $O(\sqrt{\log n})$. More generally, let H be an arbitrary fixed graph and let \mathcal{G} be the class of all graphs not containing H as a minor; then any n -point \mathcal{G} -metric can be embedded in ℓ_2 with distortion $O(\sqrt{\log n})$.

This bound is tight even for series-parallel graphs (no K_4 minor) [NR02]; the example is obtained by starting with a 4-cycle and repeatedly replacing each edge by two paths of length 2.

A challenging conjecture, one that would have significant algorithmic consequences, states that under the conditions of Rao's theorem, all \mathcal{G} -metrics can be c -embedded in ℓ_1 for some c depending only on \mathcal{G} (but not on the number of points). Apparently, this conjecture was first published in [GNRS99], where it was verified for the forbidden minors K_4 (series-parallel graphs) and $K_{2,3}$ (outerplanar graphs).

8.3.2 METRICS DERIVED FROM OTHER METRICS

In this section we focus on metrics derived from other metrics, e.g., by defining a distance between two *sets* or *sequences* of points from the underlying metric.

GLOSSARY

Uniform metric: For any set X , the metric (X, D) is uniform if $D(p, q) = 1$ for all $p \neq q, p, q \in X$.

Hausdorff distance: For a metric space (X, D) , the Hausdorff metric H on the set 2^X of all subsets of X is given by $H(A, B) = \min(\vec{H}(A, B), \vec{H}(B, A))$, where $\vec{H}(A, B) = \sup_{a \in A} \inf_{b \in B} D(a, b)$.

Earth-mover distance: For a metric space (X, D) and an integer $d \geq 1$, the earth-mover distance of two d -element sets $A, B \subseteq X$ is the minimum weight of a perfect matching between A and B ; that is, $\min_{\text{bijective } \pi: A \rightarrow B} \sum_{a \in A} D(a, \pi(a))$.

Levenshtein distance (or **edit distance**): For a metric space $M = (\Sigma, D)$, the distance between two strings $w, w' \in \Sigma^*$ is the minimum cost of a sequence of operations that transforms w into w' . The allowed operations are: character insertion (of cost 1), character deletion (of cost 1), or replacement of a symbol a by another symbol b (of cost $D(a, b)$), where $a, b \in \Sigma$. The total cost of the sequence of operations is the sum of all operation costs.

Fréchet distance: For a metric space $M = (X, D)$, the Fréchet distance (also called the **dogkeeper's distance**) between two functions $f, g: [0, 1] \rightarrow X$ is defined as

$$\inf_{\pi: [0, 1] \rightarrow [0, 1]} \sup_{t \in [0, 1]} D(f(t), g(\pi(t)))$$

where π is continuous, monotone increasing, and such that $\pi(0) = 0, \pi(1) = 1$.

HAUSDORFF DISTANCE

The Hausdorff distance is often used in computer vision for comparing geometric shapes, represented as sets of points. However, even computing a single distance $H(A, B)$ is a nontrivial task. As noted in [FCI99], for any n -point metric space (X, D) , the Hausdorff metric on 2^X can be isometrically embedded in ℓ_∞^n .

The dimension of the host norm can be further reduced if we focus on embedding particular Hausdorff metrics. In particular, let H_M^s be the Hausdorff metric over all s -subsets of M . Farach-Colton and Indyk [FCI99] showed that if $M = (\{1, \dots, \Delta\}^k, \ell_p)$, then H_M^s can be embedded in $\ell_\infty^{d'}$ with distortion $1 + \varepsilon$, where $d' = O(s^2(1/\varepsilon)^{O(k)} \log \Delta)$. For a general (finite) metric space $M = (X, D)$ they show that H_M^s can be embedded in $\ell_\infty^{s^{O(1)} |X|^\alpha \log \Delta}$ for any $\alpha > 0$ with constant distortion, where $\Delta = (\min_{p \neq q \in X} D(p, q)) / (\max_{p, q \in X} D(p, q))$.

EARTH-MOVER DISTANCE (EMD)

A very interesting relation between embedding EMD in normed spaces and embeddings in probabilistic trees (discussed below in Section 8.4.1) was discovered in [Cha02]: If a finite metric space can be embedded in a convex combination of

dominating trees with distortion c (see definitions in Section 8.4), then the EMD over it can be embedded in ℓ_1 with distortion $O(c)$. Consequently, the EMD over any n -point metric can be embedded in ℓ_1 with distortion $O(\log n)$.

LEVENSHTEIN DISTANCE AND ITS VARIANTS

The Levenshtein distance is used in text processing and computational biology. The best algorithm computing the Levenshtein distance of two strings w, w' , even approximately, has running time of order $|w| \cdot |w'|$ (for a constant-size Σ). Not much is known about embeddability of this metric in normed spaces, even in the simplest (but nevertheless quite common) case of the uniform metric over $\Sigma = \{0, 1\}$. It is known, however, that the Levenshtein metric, restricted to a certain set of strings, is isomorphic to the shortest path metric over $K_{2,n}$ [ADG⁺03]; this implies that it cannot be embedded in ℓ_1 (or even the square of ℓ_2) with distortion better than $3/2 - O(1/n)$.

However, if we modify the definition of the distance by permitting the movement of an arbitrarily long contiguous block of characters as a single operation, and if the underlying metric is uniform, then the resulting **block-edit** metric can be embedded in ℓ_1 with distortion $O(\log l \cdot \log^* l)$, where l is the length of the embedded strings (see [MS00, CM02] and references therein). The modified metric has applications in computational biology and in string compression. The embedding of a given string can be computed in almost linear time, which yields a very fast approximation algorithm for computing the distance between two strings (the exact distance computation is NP-hard!).

FRÉCHET METRIC

The Fréchet metric is an interesting metric measuring the distances between two *curves*. From the applications perspective, it is interesting to investigate the case where $M = \ell_2^k$ and f, g are continuous, closed polygonal chains, consisting of (say) at most d segments each. Denote the set of such curves by C_d^k . It is not known whether C_d^k , under Fréchet distance, can be embedded in ℓ_∞ with finite dimension (for infinite dimension, an isometric embedding follows from the universality of the ℓ_∞ norm). On the other hand, it is easy to check that for any bounded set $S \subset \ell_\infty^d$, there is an isometry $f: S \rightarrow C_{3d}^1$.

8.3.3 OTHER SPECIAL METRICS

GLOSSARY

(1, 2)-B metric: A metric space (X, D) such that for any $x \in X$ the number of points y with $D(x, y) = 1$ is at most B , and all other distances are equal to 2.

Transposition distance: The (unfortunately named) metric D_T on the set of all permutations of $\{1, 2, \dots, n\}$; $D_T(\pi_1, \pi_2)$ is the minimum number of moves of contiguous subsequences to arbitrary positions needed to transform π_1 into π_2 .

BOUNDED DISTANCE METRICS

Trevisan [Tre01] considered approximate embeddings of $(1, 2)$ - B metrics in ℓ_p^d (in a sense somewhat different from low-distortion embeddings). Guruswami and Indyk [GI03] proved that any $(1, 2)$ - B metric can be isometrically embedded in $\ell_\infty^{O(B \log n)}$.

PERMUTATION METRICS

It was shown in [CMS01] that D_T can be $O(1)$ -embedded in ℓ_1 ; similar results were obtained for other metrics on permutations, including reversal distance and permutation edit distance.

8.4 APPROXIMATE EMBEDDINGS IN RESTRICTED METRICS

GLOSSARY

Dominating metric: Let D, D' be metrics on the same set X . Then D' dominates D if $D(x, y) \geq D'(x, y)$ for all $x, y \in X$.

Convex combination of metrics: Let X be a set, T_1, T_2, \dots, T_k metrics on it, and $\alpha_1, \dots, \alpha_k$ nonnegative reals summing to 1. The convex combination of the T_i (with coefficients α_i) is the metric D given by $D(x, y) = \sum_{i=1}^k \alpha_i T_i(x, y)$, $x, y \in X$.

Hierarchically well-separated tree (k -HST): A 1-HST is exactly an *ultrametric*; that is, the shortest-path metric on the leaves of a rooted tree T (with weighted edges) such that all leaves have the same distance from the root. For a k -HST with $k > 1$ we require that, moreover, $\Delta(v) \leq \Delta(u)/k$ whenever v is a child of u in T , where $\Delta(v)$ denotes the diameter of the subtree rooted at v (w.l.o.g. we may assume that each non-leaf has degree at least 2, and so $\Delta(v)$ equals the distance of v to the nearest leaves). *Warning:* This is a newer definition introduced in [BBM01]. Older papers, such as [Bar96, Bar98], used another definition, but the difference is merely technical, and the notion remains essentially the same.

8.4.1 PROBABILISTIC EMBEDDINGS IN TREES

A convex combination $\overline{D} = \sum_{i=1}^r \alpha_i T_i$ of some metrics T_1, \dots, T_r on X can be thought of as a **probabilistic metric** (this concept was suggested by Karp). Namely, $\overline{D}(x, y)$ is the expectation of $T_i(x, y)$ for $i \in \{1, 2, \dots, r\}$ chosen at random according to the distribution given by the α_i . Of particular interest are embeddings in convex combinations of *dominating* metrics. The domination requirement is crucial for many applications. In particular, it enables one to solve many problems over the original metric (X, D) by solving them on a (simple) metric chosen at random from T_1, \dots, T_r according to the distribution defined by the α_i .

The usefulness of probabilistic metrics comes from the fact that a sum of metrics is much more powerful than each individual metric. For example, it is not difficult to

show that there are metrics (e.g., cycles [RR98, Gup01]) that cannot be embedded in tree metrics with $o(n)$ distortion. In contrast, we have the following result:

THEOREM 8.4.1 *Fakcharoenphol, Rao, and Talwar [FRT03]*

Let (X, D) be any n -point metric space. For every $k > 1$, there exist a natural number r , k -HST metrics T_1, T_2, \dots, T_r on X , and coefficients $\alpha_1, \dots, \alpha_r > 0$ summing to 1 such that each T_i dominates D , and the (identity) embedding of (X, D) into (X, \overline{D}) , where $\overline{D} = \sum_{i=1}^r \alpha_i T_i$, has distortion $O((k/\log k) \cdot \log n)$.

The first result of this type was obtained by Alon et al [AKPW95]. Their embedding has distortion $2^{O(\sqrt{\log n \log \log n})}$, and uses convex combinations of the metrics induced by spanning trees of M . A few years later Bartal [Bar96] improved the distortion bound considerably, to $O(\log^2 n)$ and later even to $O(\log n \log \log n)$ [Bar98]. The bound on the distortion in the theorem above is optimal up to a constant factor for every fixed k , since any convex combination of tree metrics embeds isometrically into ℓ_1 .

The constructions in [Bar96, Bar98, FRT03] generate trees with Steiner nodes (i.e., nodes that do not belong to X). However, one can get rid of such nodes in *any* tree while increasing the distortion by at most 8 [Gup01].

An interesting extra feature of the construction of Alon et al. mentioned above is that if the metric D is given as the shortest-path metric of a (weighted) graph G on the vertex set X , then all the T_i are spanning trees of this G . None of the constructions in [Bar96, Bar98, FRT03] share this property.

The embedding algorithms in Bartal's papers [Bar96, Bar98] are randomized and run in polynomial time. A deterministic algorithm for the same problem was given in [CCG⁺98]. The latter algorithm constructs a distribution over $O(n \log n)$ trees (the number of trees in Bartal's construction was exponential in n).

8.4.2 RAMSEY-TYPE THEOREMS

Many Ramsey-type questions can be asked in connection with low-distortion embeddings of metric spaces. For example, given classes \mathcal{X} and \mathcal{Y} of finite metric spaces, one can ask whether for every n -point space $Y \in \mathcal{Y}$ there is an m -point $X \in \mathcal{X}$ such that X can be α -embedded in Y , for given n, m, α .

Important results were obtained in [BBM01], and later greatly improved and extended in [BLMN03], for \mathcal{X} the class of all k -HST and \mathcal{Y} the class of all finite metric spaces; they were used for a lower bound in a significant algorithmic problem (metrical task systems). Let us quote some of the numerous results of Bartal et al.:

THEOREM 8.4.2 *Bartal, Linial, Mendel, and Naor [BLMN03]*

Let $R_{\text{UM}}(n, \alpha)$ denote the largest m such that for every n -point metric space Y there exists an m -point 1-HST (i.e., ultrametric) that α -embeds in Y , and let $R_2(n, \alpha)$ be defined similarly with “ultrametric” replaced with “Euclidean metric.”

- (i) There are positive constants C, C_1, c such that for every $\alpha > 2$ and all n ,

$$n^{1-C_1(\log \alpha)/\alpha} \leq R_{\text{UM}}(n, \alpha) \leq R_2(n, \alpha) \leq C n^{1-c/\alpha}.$$

- (ii) (Sharp threshold at distortion 2) For every $\alpha > 2$, there exists $c(\alpha) > 0$ such that $R_2(n, \alpha) \geq R_{\text{UM}}(n, \alpha) \geq n^{c(\alpha)}$ for all n , while for every $\alpha \in (1, 2)$, we

have $c'(\alpha) \log n \leq R_{\text{UM}}(n, \alpha) \leq R_2(n, \alpha) \leq 2 \log n + C'(\alpha)$ for all n , with suitable positive $c'(\alpha)$ and $C'(\alpha)$.

For embedding a k -HST in a given space, one can use the fact that every ultrametric is k -equivalent to a k -HST. For an earlier result similar to the second part of (ii), showing that the largest Euclidean subspace $(1+\varepsilon)$ -embeddable in a general n -point metric space has size $\Theta(\log n)$ for all sufficiently small fixed $\varepsilon > 0$, see [BFM86].

8.4.3 APPROXIMATION BY SPARSE GRAPHS

GLOSSARY

t -spanner: A subgraph H of a graph G (possibly with weighted edges) is a t -spanner of G if $D_H(u, v) \leq t \cdot D_G(u, v)$ for every $u, v \in V(G)$.

Sparse spanners are useful as a more economic representation of a given graph (note that if H is a t -spanner of G , then the identity map $V(G) \rightarrow V(H)$ is a t -embedding).

THEOREM 8.4.3 Althöfer et al. [ADD⁺93]

For every integer $t \geq 2$, every n -vertex graph G has a t -spanner with at most $m(t, n)$ edges, where $m(g, n) = O(n^{1+1/\lfloor g/2 \rfloor})$ is the maximum possible number of edges of an n -vertex graph of girth $g + 1$.

The proof is extremely simple: Start with empty H , consider the edges of G one by one from the shortest to the longest, and insert each edge into the current H unless it creates a cycle with at most t edges. It is also immediately seen that the bound $m(t, n)$ is the best possible in the worst case.

Rabinovich and Raz [RR98] proved that there are (unweighted) n -vertex graphs G that cannot be t -embedded in graphs (possibly weighted) with fewer than $m(\Omega(t), n)$ edges (for t sufficiently large and n sufficiently large in terms of t). Their main tool is the following lemma, proved by elementary topological considerations: If H is a simple unweighted connected n -vertex graph of girth g and G is a (possibly weighted) graph on at least n vertices with $\chi(G) < \chi(H)$, then H cannot be c -embedded in G for $c < g/4 - 3/2$; here $\chi(G)$ denotes the **Euler characteristic** of a graph G , which, for G connected, equals $|E(G)| - |V(G)| + 1$.

8.5 ALGORITHMIC APPLICATIONS OF EMBEDDINGS

In this section we give a brief overview of the scenarios in which embeddings have been used in the design of algorithms and for determining computational complexity. For a more detailed survey, see [Ind01].

The most typical scenario is as follows. Suppose we have a problem defined over a set of points in a metric space M . If the metric space is “complex” enough, the problem is likely to be NP-hard. To solve the problem, we embed the metric in a “simple” metric M' , and solve the problem there. This gives an approximation

algorithm for the original problem, whose approximation factor depends on the distortion of the embedding.

The implementation of this general paradigm depends on “complex” and “simple” metrics M and M' . The most frequent scenarios are as follows:

1. *General metrics → tree metrics.* This approach uses the theorems of [Bar98, FRT03], which enable the embedding of an arbitrary finite metric space, in a “probabilistic” way, in tree metrics, with low distortion. It is not difficult to see that if the goal of the original problem is to minimize a linear function of the interpoint distances, then the properties guaranteed by the above embedding are sufficient to show that given a c -approximation algorithm for HST’s (or trees, resp.), one can construct an $O(c \log n \log \log n)$ -approximation (or $O(c \log n)$ -approximation, resp.) algorithm for the original metric. Since the random choice of a tree does not depend on the function to be optimized, this approach works even if the optimization function is not known in advance. Thus, this approach has been very successful for both *offline* and *online* problems. In particular, it led to a polylog(n)-competitive algorithm [BBBT97] for metrical task systems, resolving a long-standing conjecture. In the latter paper, the embedding in HST’s (as opposed to general trees) is crucial to obtain the result.
2. *General metric → low-dimensional normed spaces.* In this case we use Bourgain’s or Matoušek’s theorem to obtain a low-dimensional approximate representation of a metric. Since the host metric is low-dimensional, each point can be represented using a small number of bits. This has interesting consequences for approximate proximity-preserving labeling [Pel99, GPPR01].
3. *Specific metrics → normed spaces.* This approach uses the results of Section 8.3.2, which provide embeddings of certain metrics (e.g., Hausdorff or Levenshtein metrics) in normed spaces. This enables the use of algorithmic tools designed for normed spaces (see, e.g., Chapter 39 of this Handbook) for problems defined over more complex metrics.
4. *High-dimensional spaces → low dimensional spaces.* Here, we use dimensionality reduction techniques, notably the Johnson-Lindenstrauss theorem. In this way, we reduce the dimension of the original space to $O(\log n)$, which yields significant savings in the running time and/or space. The improvement is particularly impressive if an algorithm for the original problem uses space/time *exponential* in the dimension (see, e.g., Chapter 39).

We note, however, that for most applications, the embedding properties listed in the statement of Theorem 8.2.3 are not sufficient. Instead, one must often use additional properties of the embedding, such as:

- The embedding is chosen at random, independently of the input point set. This property is crucial in situations where not all points are known in advance (e.g., for the nearest neighbor problem).
- The mapping is linear. This property is used, e.g., for dimensionality reduction theorems for hyperplanes (i.e., when the input set can consist of points, lines, planes etc.) [Mag02], and for low-space computation as described below.

- The coefficients of the mapping matrix are chosen independently of each other (this property holds for *some* but not *all* proofs of dimensionality reduction theorems). This property is useful, e.g., if we want to obtain deterministic versions of dimensionality reduction theorems [Ind00, Siv02, EIO02], which have applications to the derandomization of approximation algorithms based on semidefinite programming.
5. “*Complex*” normed spaces → “*simple*” normed spaces. The “complexity” of a normed space clearly depends on the problem we want to solve. For example, if we want to find the *diameter* of a set of points, it is very helpful if the interpoint distances are induced by the l_∞^d norm. In this case, the diameter of the point set is equal to the maximum diameter of all one-dimensional point sets, obtained by projecting the (d -dimensional) points onto one of the coordinates. This approach gives an $O(nd)$ time for computing the diameter in l_∞^d . However, from Section 8.1 we know that the space l_1^d can be isometrically embedded in $l_\infty^{2^{d-1}}$. Thus, we obtain a linear-time (assuming constant dimension) algorithm for computing the diameter in the l_1 norm. Other embedding results described in Section 8.2 have similar algorithmic applications as well.

A second type of result involves using the embeddings in the “reverse” directions, in order to derive *lower* bounds. Specifically, in order to show a hardness result for a metric M' , it suffices to show that a given problem is hard (to approximate) in a metric M that can be embedded in M' . This approach has been used to prove the following results:

- In [Tre01, GI03], it was shown that certain geometric problems (e.g., TSP) are hard to approximate even in $\Theta(\log n)$ dimensions. This was achieved by embedding $(1, 2)$ - B metrics (known to be the “hard” cases) in $l_p^{O(\log n)}$.
- In [BBM01], it was shown that certain online problems (metrical task systems) do not have $\Omega(\log n / \log^{O(1)} \log n)$ -competitive algorithms. This was achieved by showing that “large” HST metrics can be embedded in arbitrary finite metrics, and proving a lower bound for HST metrics.

Finally, embeddings can be used for problems that, at first sight, do not seem to be “metric” in nature. Notable examples of such an application are approximation algorithms for graph problems, such as the algorithm of [LLR95] for the sparsest cut problem and for graph bandwidth [Fei00]. In particular, the former problem can be phrased as finding a cut metric minimizing a certain objective function. Although the problem is NP-hard, its relaxation that requires finding just a metric (minimizing the same objective function) can be solved in polynomial time via linear programming. The algorithm proceeds by embedding the solution metric in l_1 (with low distortion) and decomposing it into a convex combination of cut metrics. It can be shown that that one of those cut metrics provides an approximate solution to the sparsest cut problem.

Another area whose relation to embeddings is not *a priori* apparent is low-space computing. A prototypical example of such a problem is a data structure that maintains a d -dimensional vector x (under increments/decrements of x ’s coordinates). When queried, the data structure reports an approximate value of $\|x\|_p$. In particular, the case $p = 0$ corresponds to maintaining an approximate number of nonzero coordinates. Alternatively, one could request a succinct (e.g., piecewise

constant with few pieces) approximation of x , viewed as a function from $\{1, \dots, d\}$ into the reals. Such problems are motivated by database applications.

In order to obtain low-storage algorithms solving such problems, we can apply dimensionality reduction techniques to reduce the dimension, while approximately preserving important properties of x (e.g., its norm, or its best succinct approximation). In this way, we only need to store the image Ax of x . Since the update operations on x are linear, they can be easily transformed into operations on Ax . One also has to ensure that there is no need to store the description of A explicitly; this is done by showing that a “pseudorandom” matrix A is good enough [AMS99, Ind00].

TABLE 8.5.1 A summary of approximate embeddings.

FROM	TO	DISTORTION	REFERENCE
any constant-degree expander k -reg. graph, $k \geq 3$, girth g	ℓ_p , $1 \leq p < \infty$ ℓ_p , $p < \infty$ fixed ℓ_2 $\ell_\infty^{O(bn^{1/b} \log n)}$ $\Omega(n^{1/b})$ -dim'l. normed space	$O(\log n)$ $\Omega(\log n)$ $\Omega(\sqrt{g})$ $2b-1$, $b=1, 2, \dots$ $2b-1$, $b=1, 2, \dots$ (Erdős's conj.!)	[Bou85] [LLR95] [LMN02] [Mat96] [Mat96] [Mat90] [Mat90]
any some	ℓ_1^1 ℓ_p^d , d fixed	$\Theta(n)$ $O(n^{2/d} \log^{3/2} n)$, $\Omega(n^{1/\lfloor(d+1)/2\rfloor})$	[Mat90] [Mat90]
ℓ_2 metric	$\ell_2^{O(\log n/\varepsilon^2)}$	$1 + \varepsilon$	[JL84]
ℓ_1 metric	$\ell_1^{n^\alpha}$, $0 < \alpha < 1$	$\Omega(\alpha^{-1/2})$	[BC03]
planar or forbidden minor series-parallel	ℓ_2 ℓ_2	$O(\sqrt{\log n})$ $\Omega(\sqrt{\log n})$	[Rao99] [NR02]
planar outerplanar or series-parallel	$\ell_\infty^{O(\log^2 n)}$ ℓ_1	$O(1)$ $O(1)$	implicit in [Rao99] [GNRS99] (folklore)
tree	ℓ_1	1	
tree	$\ell_\infty^{O(\log n)}$	1	[LLR95]
tree	ℓ_2	$\Theta((\log \log n)^{1/2})$	[Bou86, Mat99]
tree	ℓ_2^d	$O(n^{1/(d-1)})$	[Gup00]
tree, unit edges	ℓ_2^2	$\Theta(\sqrt{n})$	[BMMV02]
Hausdorff metric over (X, D)	$\ell_\infty^{ X }$	1	[FCI99]
Hausd. over s -subsets of (X, D)	$\ell_\infty^{s^{O(1)} X ^\alpha \log \Delta}$	$c(\alpha)$	[FCI99]
Hausd. over s -subsets of ℓ_p^k	$\ell_\infty^{s^2(1/\varepsilon)^{O(k)} \log \Delta}$	$1 + \varepsilon$	[FCI99]
EMD over (X, D)	ℓ_1	$O(\log X)$	[Cha02, FRT03]
Levenshtein metric	ℓ_1	$\geq 3/2$	[ADG ⁺ 03]
block-edit metric over Σ^d	ℓ_1	$O(\log d \cdot \log^* d)$	[MS00, CM02]
(1,2)-B metric	$\ell_\infty^{O(B \log n)}$	1	[GI03]; for ℓ_p cf. [Tre01]
any	convex comb. of dom. trees (HSTs)	$O(\log n)$	[FRT03]
any	convex comb. of spanning trees	$2^{O(\sqrt{\log n \log \log n})}$	[AKPW95]

8.6 OPEN PROBLEMS AND WORK IN PROGRESS

The time of writing of this chapter (2002) seems to be a period of particularly rapid development in the area of low-distortion embeddings of metric spaces. Many significant results have recently been achieved, and some of them are still unpublished (or not yet even written). We have tried to mention at least some of them, but it is clear that some parts of the chapter will become obsolete very soon.

Instead of stating open problems here, we refer to a list recently compiled by the second author [Mat02b]. It is available on the Web, and it might occasionally be updated to reflect new developments.

8.7 SOURCES AND RELATED MATERIAL

Discrete metric spaces have been studied from many different points of view, and the area is quite wide and diverse. The low-distortion embeddings treated in this chapter constitute only one particular (although very significant) direction. For recent results in some other directions the reader may consult [Cam00, DDL98, DD96], for instance. For more detailed overviews of the topics surveyed here, with many more references, the reader is referred to Chapter 15 in [Mat02a] (including proofs of basic results) and [Ind01] (with emphasis on algorithmic applications), as well as to [Lin02]. Approximate embeddings of normed spaces are treated, e.g., in [MS86]. A recent general reference for isometric embeddings, especially embeddings in ℓ_1 , is [DL97].

RELATED CHAPTERS

Chapter 39: Nearest neighbors in high-dimensional spaces

REFERENCES

- [Ach01] D. Achlioptas. Database-friendly random projections. In *Proc. 20th Annu. ACM SIGACT-SIGMOD-SIGART Sympos. Princip. Database Syst.*, pages 274–281, 2001.
- [ADD⁺93] I. Althöfer, G. Das, D.P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9:81–100, 1993.
- [ADG⁺03] A. Andoni, M. Deza, A. Gupta, P. Indyk, and S. Raskhodnikova. Lower bounds for embedding of edit distance into normed spaces. In *Proc. 14th Annu. ACM-SIAM Sympos. Discrete Algor.*, 2003.
- [AKPW95] N. Alon, R.M. Karp, D. Peleg, and D. West. A graph-theoretic game and its application to the k -server problem. *SIAM J. Comput.*, 24:78–100, 1995.
- [AMS99] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58:137–147, 1999.
- [Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic appli-

- cations. In *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 184–193, 1996.
- [Bar98] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 161–168, 1998.
- [BBBT97] Y. Bartal, A. Blum, C. Burch, and A. Tomkins. A polylog(n)-competitive algorithm for metrical task systems. In *Proc. 29th Annu. ACM Sympos. Theory Comput.*, pages 711–719, 1997.
- [BBM01] Y. Bartal, B. Bollobás, and M. Mendel. Ramsey-type theorems for metric spaces with applications to online problems. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 396–405, 2001.
- [BC96] H.-J. Bandelt and V. Chepoi. Embedding metric spaces in the rectilinear plane: a six-point criterion. *Discrete Comput. Geom.*, 15:107–117, 1996.
- [BC03] B. Brinkman and M. Charikar. On the impossibility of dimension reduction in ℓ_1 . In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, 2003.
- [BD92] H.-J. Bandelt and A. Dress. A canonical decomposition theory for metrics on a finite set. *Adv. Math.*, 92:47–105, 1992.
- [BFM86] J. Bourgain, T. Figiel, and V. Milman. On Hilbertian subsets of finite metric spaces. *Israel J. Math.*, 55:147–152, 1986.
- [BLMN03] Y. Bartal, N. Linial, M. Mendel, and A. Naor. On metric Ramsey-type phenomena. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, 2003.
- [BMMV02] R. Babilon, J. Matoušek, J. Maxová, and P. Valtr. Low-distortion embeddings of trees. In *Proc. Graph Drawing 2001*. Springer-Verlag, Berlin, 2002.
- [Bou85] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel J. Math.*, 52:46–52, 1985.
- [Bou86] J. Bourgain. The metrical interpretation of superreflexivity in Banach spaces. *Israel J. Math.*, 56:222–230, 1986.
- [Cam00] P. Cameron, editor. *Discrete Metric Spaces*. Selected papers from the 3rd International Conference held in Marseille, September 15–18, 1998. *European J. Combin.*, 21(6), 2000.
- [CCG⁺98] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S.A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 379–388, 1998.
- [Cha02] M. Charikar. Similarity estimation techniques from rounding. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 380–388, 2002.
- [CM02] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algor.*, pages 667–676, 2002.
- [CMS01] G. Cormode, M. Muthukrishnan, and C. Sahinalp. Permutation editing and matching via embeddings. In *Proc. 28th Internat. Colloq. Automata Lang. Program. (ICALP)*, pages 481–492, 2001.
- [DD96] W. Deuber and M. Deza, editors. *Discrete Metric Spaces*. Papers from the conference held in Bielefeld, November 18–22, 1994. *European J. Combin.*, 17 (2–3), 1996.
- [DDL98] W. Deuber, M. Deza, and B. Leclerc, editors. *Discrete Metric Spaces*. Papers from the International Conference held at the Université Claude Bernard, Villeurbanne, September 17–20, 1996. *Discrete Math.*, 192 (1–3), 1998.

- [Djo73] D.Z. Djokovic. Distance preserving subgraphs of hypercubes. *J. Combin. Theory Ser. B*, 14:263–267, 1973.
- [DL97] M.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Volume 15 of *Algor. Combin.* Springer-Verlag, Berlin, 1997.
- [DV01] J. Dunagan and S. Vempala. On Euclidean embeddings and bandwidth minimization. *Proc. 5th Workshop on Randomization and Approximation*, pages 229–240, 2001.
- [EIO02] L. Engebretsen, P. Indyk, and R. O'Donnell. Derandomized dimensionality reduction with applications. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algor.*, pages 705–712, 2002.
- [Erd64] P. Erdős. Extremal problems in graph theory. *Theory of Graphs and Its Applications (Proc. Sympos. Smolenice, 1963)*, pages 29–36, 1964.
- [FCI99] M. Farach-Colton and P. Indyk. Approximate nearest neighbor algorithms for Hausdorff metrics via embeddings. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 171–179, 1999.
- [Fei00] U. Feige. Approximating the bandwidth via volume respecting embeddings. *J. Comput. System Sci.*, 60:510–539, 2000.
- [FRT03] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, 2003.
- [GI03] V. Guruswami and P. Indyk. Embeddings and non-approximability of geometric problems. In *Proc. 14th Annu. ACM-SIAM Sympos. Discrete Algor.*, 2003.
- [GNRS99] A. Gupta, I. Newman, Yu. Rabinovich, and A. Sinclair. Cuts, trees and ℓ_1 -embeddings of graphs. In *Proc. 40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 399–409, 1999.
- [GPPR01] C. Gavoille, D. Peleg, S. Perennes, and R. Raz. Distance labeling in graphs. *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algor.*, pages 210–219, 2001.
- [Gup00] A. Gupta. Embedding tree metrics into low dimensional Euclidean spaces. *Discrete Comput. Geom.*, 24:105–116, 2000.
- [Gup01] A. Gupta. Steiner nodes in trees don't (really) help. In *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algor.*, pages 220–227, 2001.
- [IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 604–613, 1998.
- [Ind00] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 189–197, 2000.
- [Ind01] P. Indyk. Algorithmic applications of low-distortion embeddings. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 10–33, 2001.
- [JL84] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemp. Math.*, 26:189–206, 1984.
- [Lin02] N. Linial. Finite metric spaces—combinatorics, geometry and algorithms. In volume III of *Proc. Internat. Congress Math.*, Beijing, 2002, pages 573–586.
- [LLR95] N. Linial, E. London, and Yu. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.
- [LMN02] N. Linial, A. Magen, and A. Naor. Euclidean embeddings of regular graphs—the girth lower bound. *Geom. Funct. Anal.*, 12:380–394, 2002.

- [Mag02] A. Magen. Dimensionality reductions that preserve volumes and distance to affine spaces, and their algorithmic applications. In *Proc. 6th RANDOM*, pages 239–253, 2002.
- [Mat90] J. Matoušek. Bi-Lipschitz embeddings into low-dimensional Euclidean spaces. *Comment. Math. Univ. Carolin.*, 31:589–600, 1990.
- [Mat96] J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel J. Math.*, 93:333–344, 1996.
- [Mat99] J. Matoušek. On embedding trees into uniformly convex Banach spaces. *Israel J. Math.*, 114:221–237, 1999.
- [Mat02a] J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, New York, 2002.
- [Mat02b] J. Matoušek, editor. Open problems, Workshop on Discrete Metric Spaces and Their Algorithmic Applications, Haifa, March 3–7, 2002. KAM Series (Tech. Report), Department of Applied Mathematics, Charles University, Prague, 2002. Available at <http://kam.mff.cuni.cz/~matousek/haifaop.ps>.
- [MS86] V.D. Milman and G. Schechtman. *Asymptotic Theory of Finite Dimensional Normed Spaces*. Volume 1200 of *Lecture Notes in Math.* Springer-Verlag, Berlin, 1986.
- [MS00] S. Muthukrishnan and C. Sahinalp. Approximate nearest neighbors and sequence comparison with block operations. In *Proc. 32nd Annu. ACM Sympos. Theory Comput.*, pages 416–424, 2000.
- [NR02] I. Newman and Yu. Rabinovich. A lower bound on the distortion of embedding planar metrics into Euclidean space. *Discrete Comput. Geom.*, 29:77–81, 2003.
- [Pel99] D. Peleg. Proximity-preserving labeling schemes and their applications. *Proc. 25th Workshop on Graph-Theoretic Aspects of Comput. Sci.*, volume 1665 of *Lecture Notes in Comput. Sci.*, Springer-Verlag, New York, pages 30–41, 1999.
- [Rao99] S. Rao. Small distortion and volume respecting embeddings for planar and Euclidean metrics. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 300–306, 1999.
- [RR98] Yu. Rabinovich and R. Raz. Lower bounds on the distortion of embedding finite metric spaces in graphs. *Discrete Comput. Geom.*, 19:79–94, 1998.
- [Sch38] I.J. Schoenberg. Metric spaces and positive definite functions. *Trans. Amer. Math. Soc.*, 44:522–53, 1938.
- [Shp93] S.V. Shpectorov. On scale embeddings of graphs into hypercubes. *European J. Combin.*, 14:117–130, 1993.
- [Siv02] D. Sivakumar. Algorithmic derandomization from complexity theory. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 619–626, 2002.
- [Tre01] L. Trevisan. When Hamming meets Euclid: The approximability of geometric TSP and MST. *SIAM J. Comput.*, 30:475–485, 2001.
- [TZ01] M. Thorup and U. Zwick. Approximate distance oracles. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 183–192, 2001.

9 GEOMETRY AND TOPOLOGY OF POLYGONAL LINKAGES

Robert Connelly and Erik D. Demaine

INTRODUCTION

There is a long and involved history of linkages starting at least in the nineteenth century with the advent of very complicated and intricate machinery. Some of the practical problems involved led to interesting, nontrivial geometric problems, and even recently there has been progress on some very basic questions. We will attempt to point the reader to some of the results that we know in this direction.

There are several points of view and groups of people working on various aspects of the theory of linkages, but they seem to be disjointed, with each group unaware of other groups that are in related or even overlapping fields. Despite that, we will also try to point out connections when we can.

9.1 MATHEMATICAL THEORY OF LINKAGES

The underlying principles and definitions are mathematical and in particular geometric. Despite the long history of kinematics, even of theoretical kinematics (see, e.g., Bottema and Roth [BR79a]), only since the 1970s does there seem to be any systematic attempt to explore the mathematical and geometric foundations of a theory of linkages.

We begin with some definitions, some of which follow those in rigidity theory described in [Chapter 60](#). The rough, intuitive notions are as follows. A *linkage* is a combinatorial structure plus edge lengths, and we often distinguish three special types of linkages: arcs, cycles, and trees. A *configuration* realizes a linkage in Euclidean space, a *reconfiguration* (or *flex*) is a continuum of such configurations, and the *configuration space* embodies all reconfigurations. The configuration space can be considered as either allowing or disallowing bars to intersect each other.

GLOSSARY

Bar linkage or **linkage**: A graph $G = (V, E)$ and an assignment $\ell : E \rightarrow \mathbb{R}^+$ of positive real *lengths* to edges.

Vertex or **joint**: A vertex of a linkage.

Bar or **link**: An edge e of a linkage, which has a specified fixed length $\ell(e)$.

FIGURE 9.1.1

Different types of linkages, according to whether the underlying graph is a path, cycle, or tree, or whether the graph is arbitrary.

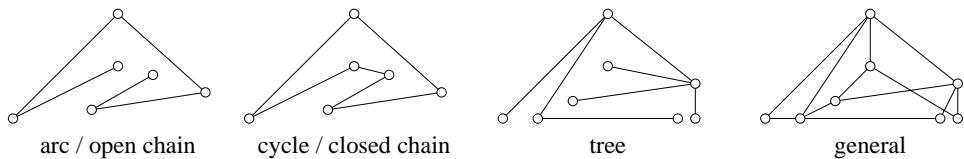
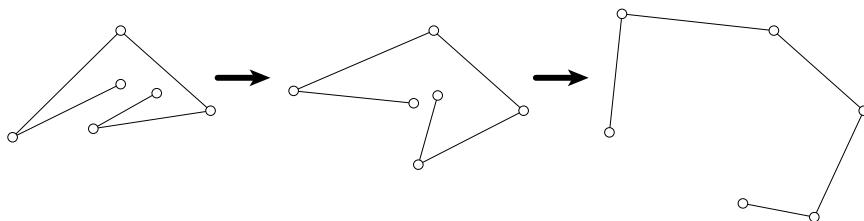


FIGURE 9.1.2

Snapshots of a reconfiguration of a polygonal arc.



Polygonal arc: A linkage whose underlying graph is a single path. (Also called an *open chain* or a *ruler*.)

Polygonal cycle: A linkage whose underlying graph is a single cycle. (Also called a *closed chain* or a *polygon*.)

Polygonal tree: A linkage whose underlying graph is a single tree.

Configuration of a linkage in d -space: A mapping $p : V \rightarrow \mathbb{R}^d$ specifying a point $p(v) \in \mathbb{R}^d$ for each vertex v of the linkage, such that each bar $\{v, w\} \in E$ has the desired length $\ell(e)$, i.e., $|p(v) - p(w)| = \ell(e)$.

A configuration can be viewed as a point p in $\mathbb{R}^{d|V|}$ by arbitrarily ordering the vertices in V and assigning the coordinates of the i th vertex ($0 \leq i < |V|$) to coordinates $id + 1, id + 2, \dots, id + d$ of p .

Framework or bar framework: A linkage together with a configuration.

Reconfiguration or motion or flex of a linkage: A continuous function $f : [0, 1] \rightarrow \mathbb{R}^{d|V|}$ specifying a configuration of the linkage for every moment in time between 0 and 1.

Configuration space or moduli space of a linkage: The set \mathcal{M} of all configurations (treated as points in $\mathbb{R}^{d|V|}$) of the linkage.

Self-intersecting configuration: A configuration in which two bars intersect but are not incident in the underlying graph of the linkage.

Reconfiguration avoiding self-intersection: A reconfiguration f in which no configuration $f(t)$ self-intersects.

Configuration space of a linkage, disallowing self-intersection: The subset \mathcal{F} of the configuration space \mathcal{M} in which no configuration self-intersects. (Also called the *free space* of the linkage.)

Paths in the configuration space of a linkage capture the key notion of reconfiguration (either allowing or disallowing self-intersection as appropriate). Many important questions about linkages can be most easily phrased in terms of the configuration space. For example, we are often interested in whether the configuration space is connected (every configuration can be reconfigured into every other configuration), or in the topology of the configuration space.

9.2 CONFIGURATION SPACES OF ARCS AND CYCLES WITH POSSIBLE INTERSECTIONS

One fundamental problem is to compute the topology of the configuration space of planar polygonal cycles (polygons), allowing possible self-intersections. There is a long list of results in increasing generality for computing information about the algebraic topological invariants of this configuration space. One approach is through Morse Theory, which reveals some of the basic information, in particular, the connectivity and some of the easier invariants such as the Euler characteristic.

CONNECTIVITY

The following is an early result possibly first due to [Hau91], but rediscovered by [Jag92], and then rediscovered again or generalized considerably by many others, in particular, [Kam99, KT99, MS00, KM95, LW95].

THEOREM 9.2.1 *Connectivity for planar polygons* [Hau91]

Let $s_1 \leq s_2 \leq \dots \leq s_n$ be the cyclic sequence of bar lengths in a polygon, and let $s = s_1 + s_2 + \dots + s_n$. Then

- i) The configuration space is nonempty if and only if $s_n \leq s/2$.
- ii) The configuration space, modulo orientation-preserving congruences, is connected if and only if $s_{n-2} + s_{n-1} \leq s/2$. If the space is not connected, there are exactly two connected components, where each configuration in one component is the reflection of a configuration in the other component.

The configuration space is a smooth manifold if and only if there is some configuration p with all its vertices on a line, which in turn is determined by the edge lengths as described above. Also, the configuration space remains congruent no matter how we permute the cyclic sequence of bar lengths. When the linkage is not allowed to self-intersect, it is common to consider the configuration space modulo all congruences of the plane (including reflections); but when self-intersections are allowed, and condition ii) above is satisfied, it is possible to move the linkage from any configuration to its mirror image.

For polygons in dimensions higher than two, the situation is simpler:

THEOREM 9.2.2 *Connectivity for nonplanar polygons* [LW95]

The configuration space of a polygon in d -dimensional space, for $d > 2$, is always connected.

HOMOLOGY, COHOMOLOGY, AND HOMOTOPY

After connectivity, there remains the calculation of the higher homology groups, cohomology groups, and the homotopy type of the configuration space. Here is one special case as an example:

THEOREM 9.2.3 *Configuration space of equilateral polygons* [KT99]

Let \mathcal{M} be the configuration space of a polygon with n equal bar lengths, modulo congruences of the plane. The homology of \mathcal{M} is a torsion-free module given explicitly in [KT99]. When n is odd, \mathcal{M} is a smooth manifold; and when $n = 5$, \mathcal{M} is the compact, orientable two-dimensional manifold of genus 4 (originally shown in [Hav91], as well as in [Jag92]).

See also especially [KM95] for some of the basic techniques. For calculating the configuration space of graphs other than a polygon, see in particular the article [TW84], where a particular linkage, with some pinned vertices, has a configuration space that is an orientable two-dimensional manifold of genus 6.

Another case that has been considered is an equilateral polygon in 3-space with angles between incident edges fixed. This fixed-angle model arises in chemistry [CH88] and in particular in protein folding (see [Section 9.7](#)). Alternatively, a fixed angle can be simulated by adding bars between vertices of distance two along the polygon. The configuration space behaves similarly to the planar case:

THEOREM 9.2.4 *Fixed-angle equilateral 3D polygons* [CJ]

Let \mathcal{M} be the configuration space of an equilateral polygon with $n \geq 6$ equal bar lengths and fixed equal angles, modulo congruences of \mathbb{R}^3 . Suppose further that every turn angle is within an additive ϵ of $2\pi/n$ for ϵ sufficiently small (i.e., configurations are forced nearly planar). Then \mathcal{M} has at most two components. When n is odd, \mathcal{M} is a smooth manifold of dimension $n-6$. When n is even, \mathcal{M} is singular.

When $n = 6$, the underlying graph is the graph of an octahedron, and there are cases when it is rigid and cases when it is not. This linkage corresponds to cyclohexane in chemistry, and its flexibility was studied by [Bri96] and [Con78].

The restriction of the polygon configurations being almost planar leads to the following problem:

PROBLEM 9.2.5 *General equilateral equi-angular 3D polygons* [Cri92]

How many components does \mathcal{M} have in the theorem above if ϵ is allowed to be large?

9.3 CONFIGURATION SPACES WITHOUT SELF-INTERSECTIONS

When the linkage is not permitted to self-intersect, the main question that has been studied is when it can be locked. Three main classes of linkages have been studied in this context: arcs, cycles, and trees. When the linkage is planar and has cycles, we assume that the clockwise/counterclockwise orientation is given and fixed, for otherwise the linkage is trivially locked: no cycle can be “flipped over” in the plane without self-intersection.

GLOSSARY

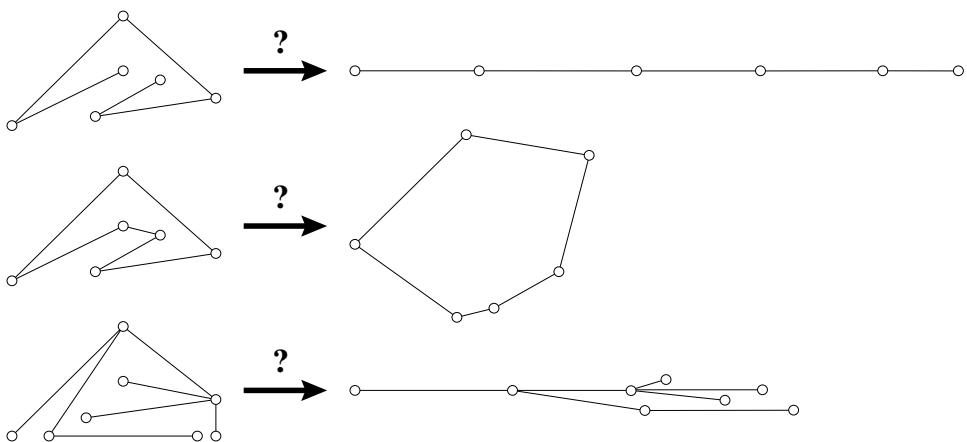
Locked linkage: A linkage whose configuration space has multiple connected components when self-intersections are disallowed.

Lockable class of linkages: There is a locked linkage in the class.

Unlockable class of linkages: No linkage in the class is locked.

FIGURE 9.3.1

The problems of arc straightening, cycle convexifying, and tree flattening.



Straightening an arc: A motion bringing a polygonal arc from a given configuration to its **straight configuration** in which every joint angle is π .

Convexifying a cycle: A motion bringing a polygonal arc from a given configuration to a **convex configuration** in which every joint angle is at most π .

Flattening a tree: A motion bringing a polygonal tree from a given configuration to a **flat configuration** in which every joint angle is either 0 , π , or 2π , and every bar points “away” from a designated root node.

WHICH LINKAGES ARE LOCKED?

Which of the main classes of linkages can be locked is summarized in [Table 9.3.1](#). In short, the existence of locked arcs and locked unknotted cycles is equivalent to the existence of knots in that dimension: this happens just in 3D. However, this equivalence is by no means obvious, especially in 2D, as evidenced by the existence of knotted trees in 2D.

One main approach for determining whether a linkage is locked is to consider the equivalent problem of finding a motion from any configuration to a **canonical configuration**. Because linkage motions are reversible and concatenable, if every configuration can be canonicalized, then every configuration can be brought to any other configuration, routing through the canonical configuration. Conversely, if

TABLE 9.3.1 Summary of what types of linkages can be locked.

	ARCS AND CYCLES	TREES
2D	Not lockable [CDR03, Str00, CDIO02]	Lockable [BDD ⁺ 02, CDR02]
3D	Lockable [CJ98, BDD ⁺ 01, Tou01]	Lockable [arcs are a special case]
4D ⁺	Not lockable [CO01]	Not lockable [CO01]

some configuration cannot be canonicalized, then we know a pair of configurations that cannot reach each other, and therefore the linkage is locked.

This idea leads to the notions of straightening arcs, convexifying cycles, and flattening trees, as defined above. There is only one straight configuration of an arc, but there are multiple convex configurations of cycles and flat configurations of trees; fortunately, it is fairly easy to reconfigure between any pair of convex configurations of a cycle [ADE⁺01] or between any pair of flat configurations of a tree [BDD⁺02].

LOCKED LINKAGES

The first results along these lines were negative (see [Figure 9.3.2](#)): polygonal arcs in 3D and unknotted polygonal cycles in 3D can be locked [CJ98], and planar polygonal trees can be locked [BDD⁺02]. Since these results, other examples of unknotted but locked 3D polygonal cycles [BDD⁺01, Tou01] and locked 2D polygonal trees [CDR02] have been discovered.

More generally and recently, Alt, Knauer, Rote, and Whitesides [AKRW03] constructed a large family of locked 2D trees and 3D arcs in which it is PSPACE-hard to determine whether one configuration can reach another configuration via a continuous motion that avoids self-intersection. Their construction combines several gadgets, many of which resemble the examples in [Figure 9.3.2](#), as well as the “interlocked” linkages of [DLOS03, DLOS02]. However, this work leaves open a closely related problem, deciding whether *every* pair of configurations can reach each other:

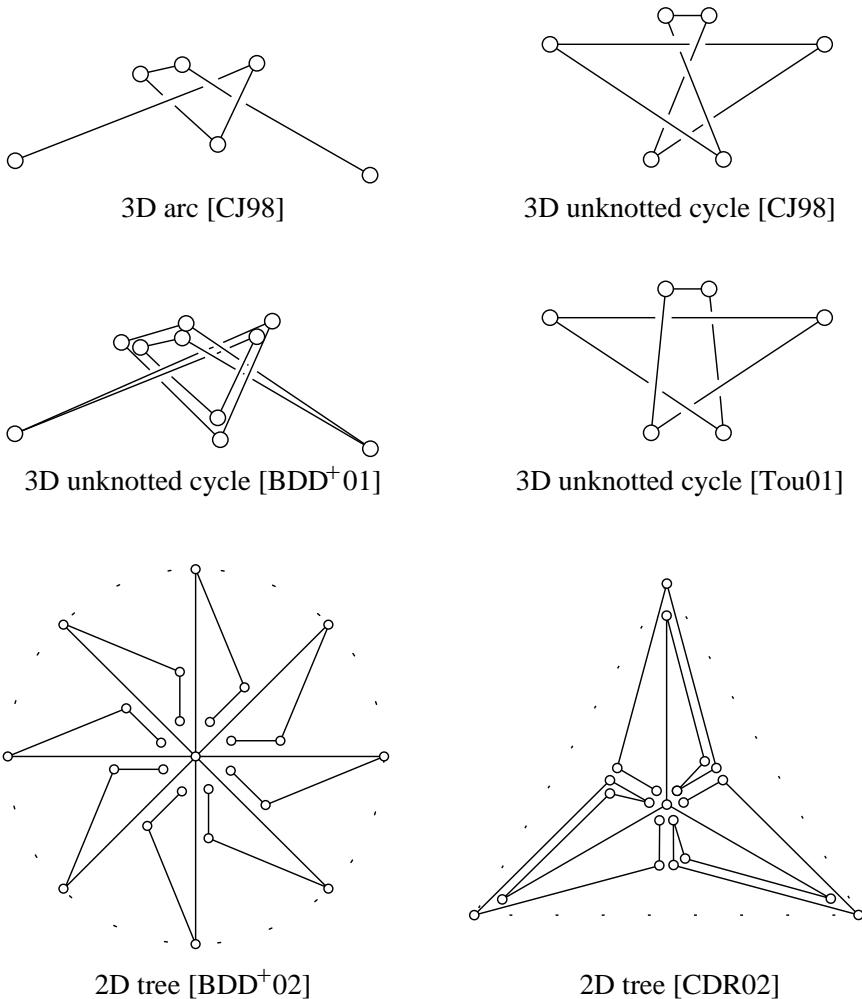
PROBLEM 9.3.1 *Complexity of testing if a linkage is locked* [BDD⁺01]

What is the complexity of deciding whether a linkage is locked? Particular cases of interest are 3D arcs, unknotted 3D cycles, and 2D trees.

UNLOCKED LINKAGES

Unlockability was first established in 4D and higher [CO01], where one-dimensional arcs, cycles, and trees have so much freedom that they can never lock. Intuitively, the barriers (self-intersecting configurations) that might prevent, e.g., straightening the vertex between the first two bars of an arc have dimension at least 2 lower than the configuration space of that vertex, and hence all barriers can be avoided. Thus, the only problem with straightening an arc vertex-by-vertex is that the configuration that results from straightening one extreme vertex might have self-intersections; in this case, the linkage can be perturbed to remove the problem. Convexifying cy-

FIGURE 9.3.2
Known examples of locked linkages.



cles in 4D and higher is more difficult, but follows a similar idea.

The last cell of Table 9.3.1 to be filled was that 2D arcs and cycles never lock [CDR03]. Indeed, the following more general theorem holds:

THEOREM 9.3.2 *Straightening 2D arcs and convexifying 2D cycles* [CDR03]

*Given a disjoint collection of polygonal arcs and polygonal cycles in the plane, there is a motion that avoids self-intersection and, after finite time, straightens every outermost arc and convexifies every outermost cycle. (An arc or cycle is **outermost** if it is not contained within another cycle.)*

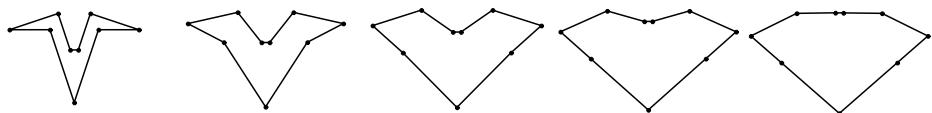
In this theorem, arcs and cycles contained within other cycles may not straighten or convexify—they simply “come along for the ride”—but this is the best we could

hope for in general.

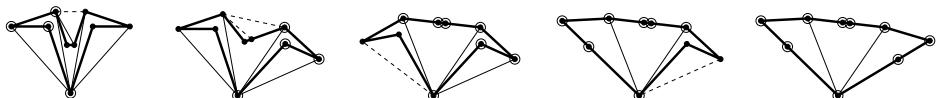
There are now three methods for solving this problem. See Figure 9.3.3 for a visual comparison on a simple example. The first method is based on flow through an ordinary differential equation defined implicitly by a convex optimization problem [CDR03]. The second method is more combinatorial and is based on algebraic motions defined by single-degree-of-freedom mechanisms given by pseudotriangulations [Str00]. The third method is based on energy minimization via gradient descent [CDIO02].

FIGURE 9.3.3

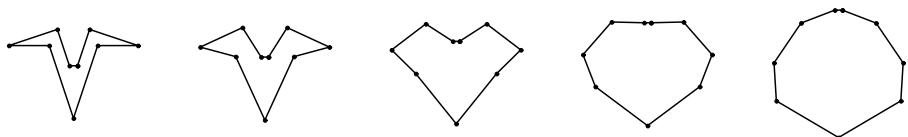
Convexifying a common polygon via all three convexification methods.



(a) Via convex programming [CDR03].



(b) Via pseudotriangulations [Str00]. Pinned vertices are circled.



(c) Via energy minimization [CDIO02].

The first two motions have the additional property of being *expansive*—the distance between every pair of vertices never decreases over time—while the third motion only relies on the existence of such a motion. The first and last motions, being flow-based, preserve any initial symmetries of the linkage. Characterizing by continuity, the three motions are respectively piecewise- C^1 , piecewise- C^∞ , and C^∞ . Only the last motion has a corresponding finite-time algorithm to compute a motion that is *piecewise-linear* through configuration space, i.e., the motion can be decomposed into steps where each angle in each step changes at a constant rate. This algorithm is also easy to implement.

SPECIAL CLASSES OF LINKAGES

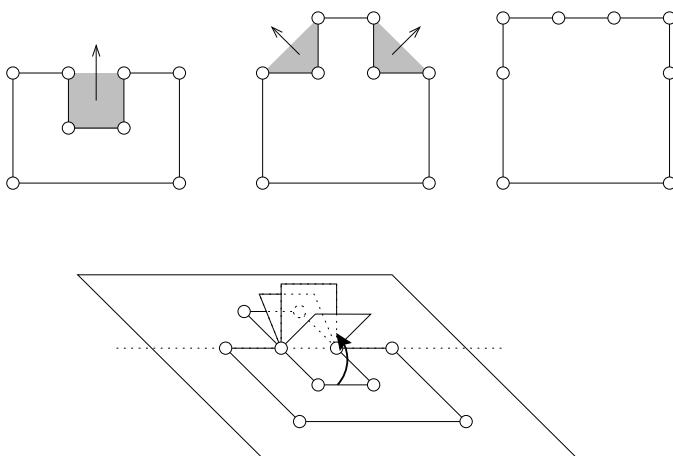
In addition to these results for general classes of linkages, various special classes have been shown to have different properties. Polygonal arcs in 3D that lie on

the surface of a convex polyhedron, or having a non-self-intersecting orthogonal projection, are never locked [BDD⁺01]. Polygonal cycles in 3D having a non-self-intersecting orthogonal projection are also never locked [CKM⁺01].

FLIPS AND FLIPTURNS

One of the first papers essentially about unlocking linkages is by Erdős [Erd35], who asked whether a particular “flipping” algorithm always convexifies a planar polygon by motions through 3D in a finite number of steps. A **flip** rotates by 180° a subchain of the polygon, called a **pocket**, whose endpoints are consecutive vertices along the convex hull of the polygon. Each such flip never causes the polygon to self-intersect.¹ Nagy [Nag39] was the first to prove that a polygon admits only finitely many flips before convexifying. Thus, pocket flipping is one suitable strategy for convexifying a 2D polygon by motions in 3D. This result was subsequently rediscovered several times; see [Tou99, Grü95].

FIGURE 9.3.4
Flipping a polygon until it is convex.



Joss and Shannon (1973) first proved that the number of flips required to convexify a polygon cannot be bounded in terms of the number of vertices, but this work remains unpublished; see [Grü95, Tou99]. However, it may still be possible to bound the number of flips using other metrics:

PROBLEM 9.3.3 *Bounding the number of flips* [M. Overmars, Feb. 1998]

Bound the maximum number of flips a polygon admits in terms of natural measures of geometric closeness such as the sharpest angle, the diameter, and the minimum distance between two nonincident edges.

¹Erdős [Erd35] originally proposed flipping multiple pockets at once, but such an operation can lead to self-intersection; Nagy [Nag39] fixed this problem by proposing flipping only one pocket at once.

A related computational problem is to compute the extreme numbers of flips:

PROBLEM 9.3.4 *Maximizing or minimizing flips* [Dem02]

What is the complexity of minimizing or maximizing the length of a convexifying sequence of flips for a given polygon?

Several variations on flips have also been considered. Grünbaum and Zaks [GZ01] generalized Nagy’s results to polygons with self-intersections; still they can be convexified by finitely many flips. Wegner [Weg93] introduced the notion of **deflations**, which are the exact reverse of flips, and Fevens et al. [FHM⁺01] showed that some polygons admit infinitely many deflations.

Flipturns are similar to flips, except that the pocket is temporarily severed from the rest of the linkage and rotated 180° in the plane around the midpoint of the hull edge. Such an operation is not a valid linkage motion, but it has the advantage that the number of flipturns that a polygon admits before convexification is $O(n^2)$ [ACD⁰², ABC⁰⁰]. This bound is tight up to a constant factor [Bie00], and there is extensive work on finding the precise constants [ACD⁰²], though some gaps remain to be closed. Also, related to Problem 9.3.4, it is known that maximizing the length of a convexifying flipturn sequence is weakly NP-hard [ACD⁰²]. Minimizing the number of flipturns leads to the following interesting problem:

PROBLEM 9.3.5 *Number of required flipturns* [Bie00]

Is there a polygon that requires $\Omega(n^2)$ flipturns to convexify, or can all polygons be convexified by $o(n^2)$ carefully chosen flipturns?

The best known lower bound is $\Omega(n)$.

INTERLOCKED LINKAGES

Combinations of polygonal arcs and cycles in 3D that can or cannot be locked (or, more accurately, “interlocked”) are studied in [DLOS03, DLOS02]. More precisely, this work studies the shortest (fewest-bar) 3D arcs and cycles that can interlock with each other. For example, three 3-arcs (arcs with three bars each) can interlock, as can a 3-arc and a 4-arc, or a 3-cycle and a 4-arc, or a 3-arc and a 4-cycle. However, two 3-arcs and arbitrarily many 2-arcs never interlock, nor can a 3-cycle and a 3-arc. Also considered in [DLOS02] is the case that some of the pieces have restricted motion, e.g., all angles are fixed, or only rigid motions are allowed.

9.4 UNIVERSALITY RESULTS

TRACING CURVES

The classic motivation of building linkages is to design a planar linkage in which one of the vertices traces a portion of a desired curve given by some polynomial function. In particular, Watt posed the problem of finding a linkage with some vertices pinned so that one vertex would trace out a line (segment). Watt’s problem, at first thought

to be impossible, was finally solved by Peaucellier in [Pea73], as well as by Lipkin in [Lip71]. See also [Kem77] and [Har74].

Later, Kempe [Kem76] described a linkage that would trace out a portion of any algebraic curve in the plane. However, his description is very brief and it leaves unspecified what portion of the algebraic curve is actually traced out, and whether there are other, possibly unwanted components or pieces of other algebraic curves that can also be traced out. This question also arises for the linkages that trace a line segment.

GLOSSARY

Real algebraic set: A subset of \mathbb{R}^N given by a finite number of polynomial equations with real coefficients.

Real semialgebraic set: A subset of \mathbb{R}^N given by a finite number of polynomial equations and inequalities with real coefficients.

It is important to realize the distinction between an algebraic set and a semialgebraic set. For example, a circle (excluding its interior) is an algebraic set, while a (closed) line segment is a semialgebraic set but not an algebraic set. The linear projection of an algebraic set is always a semialgebraic set, but it may not be an algebraic set. The configuration space of a linkage is an algebraic set, but the locus of possible positions of one of its vertices is only guaranteed to be a semialgebraic set, because it represents the projection onto the coordinates corresponding to one of the vertices of the linkage.

ARBITRARY CONFIGURATION SPACES

One of the more precise results related to Kempe's result is the following:

THEOREM 9.4.1 *Creating linkage configuration spaces* [KM95]

Let M be any compact smooth manifold. Then there is a planar linkage whose configuration space is diffeomorphic to a disjoint union of some number of copies of M .

This result was also claimed by Thurston, but there does not seem to be a written proof by him. As a consequence of this result, we obtain the following precise version of what Kempe was trying to claim. This consequence is proved by King [Kin99] using the techniques of Kapovich-Millson [KM02] and Thurston.

THEOREM 9.4.2 *Tracing out an algebraic curve* [Kin99]

Let X be any set in the plane that is the polynomial image of a closed interval. Then there is a linkage in the plane with some pinned vertices such that one of the vertices traces out X exactly.

See [JS99, BM56] for other discussions of how to create linkages to trace out at least a portion of a given algebraic curve. King [Kin] also generalizes this result to higher dimensions and to the semialgebraic sets arising from projecting the

configuration space down to consider some subset of the vertices. See also [KM02] for connections to universality theorems concerning configuration spaces of lines in the plane, for example, as in the work of [Mnë88]. Finally, the complexity results of [HJW85] described in Section 9.5 build off a universality construction similar to those mentioned above.

9.5 COMPUTATIONAL COMPLEXITY

There are a variety of algorithmic questions that can be asked about a given linkage. Most of these questions are computationally difficult to answer, either NP-hard or PSPACE-hard. Nonetheless, given the importance of these problems, there is work on developing (exponential-time) algorithms.

GLOSSARY

Ruler folding problem: Given a polygonal arc (i.e., a sequence of bar lengths) and a desired length L , is there a configuration of the arc (ruler) in which the bars lie along a common line segment of length L ? If so, find such a configuration. (The problem can also be phrased as reconfiguration, provided the linkage is permitted to self-intersect.)

Reachability problem: Given a configuration of a linkage, a distinguished vertex, and a point in the plane, is it possible to reconfigure the linkage so that the distinguished vertex touches the given point? If so, find such a reconfiguration. In this problem, the linkage has one or more vertices pinned to particular locations in the plane.

Reconfiguration problem: Given two configurations of a linkage, is it possible to reconfigure one into the other? If so, find such a reconfiguration.

Locked decision problem: Given a linkage, is it locked?

HARDNESS RESULTS

One of the simplest complexity results is about the ruler folding problem, obtained via a reduction from set partition:

THEOREM 9.5.1 *Complexity of ruler folding* [HJW85]

The ruler folding problem is NP-complete.

Building on this result, the same authors establish

THEOREM 9.5.2 *Complexity of arc reachability* [HJW85]

The reachability problem is NP-hard for a planar polygonal arc in the presence of four line-segment obstacles and permitting the arc to self-intersect.

For general linkages instead of arcs, stronger complexity results exist:

THEOREM 9.5.3 *Complexity of reachability* [HJW84]

The reachability problem is PSPACE-hard for a planar linkage without obstacles and permitting the linkage to self-intersect.

On the other hand, a similar result holds for a polygonal arc among obstacles:

THEOREM 9.5.4 *Complexity of arc reachability among obstacles* [JP85]

The reachability problem is PSPACE-hard for a planar polygonal arc in the presence of polygonal obstacles and permitting the arc to self-intersect.

Finally, when the linkage is not permitted to self-intersect, and there are no obstacles, hardness is known in cases when the linkage can be locked; see [Section 9.3](#).

THEOREM 9.5.5 *Complexity of non-self-intersecting arc reconfiguration* [AKRW03]

The reconfiguration problem is PSPACE-hard for a 3D polygonal arc or a 2D polygonal tree when the linkage is not permitted to self-intersect.

ALGORITHMS

Algorithms for linkage reconfiguration problems can be obtained from the general motion-planning results in [Chapter 47](#) (Section 47.1.1). This connection seems to have only recently been made explicit [AKRW03]. To apply the roadmap algorithm of Canny [Can87] (Theorem 47.1.2), we first phrase the algorithmic linkage problems into the motion-planning framework.

The configuration space of a given linkage is the subset of \mathbb{R}^{vc} in which every point satisfies certain bar-length constraints and, if desired, non-intersection constraints between all pairs of bars. Both types of constraints can be phrased using constant-degree polynomial equations and inequalities, e.g., the former by setting the squared length of each bar to the desired value. (There are also embeddings of the configuration space into Euclidean spaces with fewer than vc dimensions, dependent on the number of degrees of freedom in the linkage, but the vc -dimensional parameterization is most naturally semialgebraic.)

Returning to the motion-planning framework, the polynomial equations and inequalities are precisely the obstacle surfaces. The configuration space has dimension $k = vc$, and there are $n \leq b^2$ obstacle surfaces where b is the number of bars, each with degree $d = O(1)$. We can factor out the trivial rigid motions by supposing that one bar of the linkage is pinned, reducing k to $(v - 2)c$. Now running the roadmap algorithm produces a representation of the entire configuration space. By path planning within this space, we can solve the reconfiguration problem. By a simple pass through the representation, we can tell whether the space is connected, solving the locked decision problem. By slicing the space with a polynomial specifying that a particular vertex is located at a particular point in the plane, we can solve the reachability problem.

Plugging $k \leq vc$, $n \leq b^2$, and $d = O(1)$ into the roadmap algorithm with deterministic running time $O(n^k (\log n)^{d^{O(k^4)}})$ and randomized expected running time $O(n^k (\log n)^{d^{O(k^2)}})$, we obtain:

COROLLARY 9.5.6 *Roadmap algorithm applied to linkages [AKRW03]*

The reachability, reconfiguration, and locked decision problems can be solved for an arbitrary linkage with v vertices and b bars in \mathbb{R}^c using $O(b^{2vc}(\log b)2^{O(vc)^4})$ deterministic time or $O(b^{2vc}(\log b)2^{O(vc)^2})$ expected randomized time.

9.6 KINEMATICS

According to Bottema and Roth [BR79b], “kinematics is that branch of mechanics which treats the phenomenon of motion without regard to the cause of the motion. In kinematics there is no reference to mass or force; the concern is only with relative positions and their changes.” Kinematics is a subject with a long history and which has had, at various times, notable influence on and has to some extent has been partially identified with such areas as algebraic geometry, differential geometry, mechanics, singularity theory, and Lie theory. It has often been a subject studied from an engineering point of view, and there are many detailed calculations with respect to particular mechanisms of interest. As a representative example, we consider four-bar mechanisms (Figure 9.6.1):

GLOSSARY

Mechanism: A linkage with one degree of freedom, modulo global translation and rotation.

Four-bar mechanism: A four-bar polygonal cycle; see Figure 9.6.1 for an example. Sometimes called a **three-bar mechanism**.

Frame: We generally fix a frame of reference for a mechanism by pinning one bar, fixing its position in the plane. This bar is called the frame. In Figure 9.6.1, bar AB is pinned.

Coupler: A distinguished bar other than the frame. In Figure 9.6.1, we consider the coupler CD .

Coupler motion: The motion of the entire plane induced by the relative motion of the coupler with respect to the frame.

Coupler curve: The path traced during the coupler motion by any point rigidly attached to the coupler (e.g., via two additional bars). Figure 9.6.1 shows the coupler curve of the midpoint E of the coupler bar CD .

FOUR-BAR MECHANISM

Coupler curves can be surprisingly complex. In the generic case, a coupler curve of a four-bar mechanism is an algebraic curve of degree 6. Substantial effort has been put into cataloging the different shapes of coupler curves that can arise from four-bar and other mechanisms. A sample theorem in this context is the following:

THEOREM 9.6.1 *Multiplicity of coupler curves [Rob75]*

Any coupler curve of a four-bar mechanism can be generated by two other four-bar mechanisms.

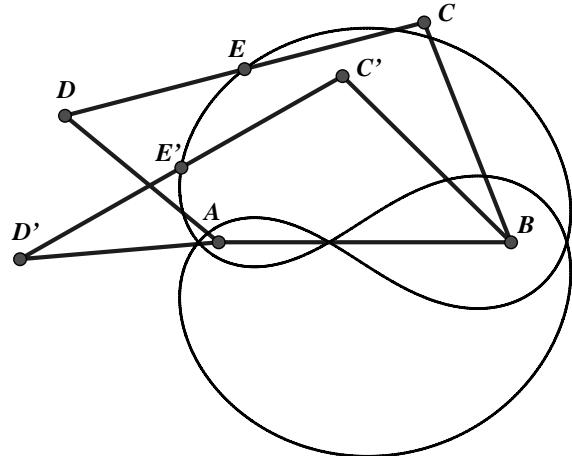


FIGURE 9.6.1

The coupler curve of the midpoint E of the coupler CD as it moves relative to the frame AB in a four-bar mechanism.

GLOSSARY

Infinitesimal motion or first-order flex: The first derivative of a motion at a moment in time, assigning a velocity vector to each point involved in the motion. (See [Chapter 60](#) for a more thorough explanation in the context of rigidity.)

Pole or instantaneous pole: The instantaneous fixed point of a first-order motion of the plane. For a rotation, the pole is the center of rotation. For a translation, the pole is a point at infinity in the projective plane. A combination of rotation and translation can be rewritten as a pure rotation.

Polode: The locus of poles over time during a motion of the plane.

POLES

Some of the central theorems in kinematics treat the instantaneous case. Poles characterize the first-order action of a motion at each moment in time. Together, the polode can be viewed relative to either the fixed plane of the frame (the **fixed polode**) or the moving plane of the coupler (**moving polode**). Apart from degenerate cases, a planar motion can be described by the moving polode rolling along the fixed polode. A basic theorem in the context of poles is the following:

THEOREM 9.6.2 Three-Pole Theorem

For any three motions of the plane, the instantaneous poles of the three mutual relative motions are collinear at any moment in time.

FURTHER READING

For a general introduction to and sampling of the field of kinematics, see [Hun78, BR79b, Sta97, McC90, Pot94, McC00]. For relations to singularity theory, see, e.g., [GHM97]. For examples, analysis, and synthesis of specific mechanisms such as the four-bar mechanism, see [GN86, Mik01, Sta99, Ale95, BS90, Leb67, Con79, Con78]. For some typical examples from an engineering viewpoint, see, e.g., [CP91, Che02, Ler00]. See also Section 59.4 of this Handbook.

9.7 APPLICATIONS

Applications of linkages arise throughout science and engineering. We highlight three modern applications: robotics, manufacturing, and protein folding.

APPLICATIONS IN ENGINEERING

The study of linkages in fact originated in the context of mechanical engineering, e.g., for the purpose of converting circular motion into linear motion. Today, one of the driving applications for linkages is *robotics*, in particular *robotic arms*.

A robotic arm can be modeled as a linkage, typically a polygonal chain. Some robotic arms have hinges that force the bars to remain coplanar, modeled by 2D chains; other arms have universal joints, modeled by 3D chains; other arms pose additional constraints (such as incident bars being coplanar, without the whole linkage necessarily being coplanar), leading to other models of linkage folding. Some planar robotic arms reserve slightly offset planar planes for the bars, modeled by a planar polygonal chain that permits self-intersection. Most other robotic arms are modeled by disallowing self-intersection.

The reachability problem is largely motivated by robotic arms, where the “hand” at one end of the arm must be placed at a particular location, e.g., to pick up an object, but the rest of the configuration is secondary. In other contexts, the entire configuration of the arm is important, and we need to plan a motion to a target configuration, leading to the reconfiguration problem. The locked decision problem is the first question one might ask about the simplicity/complexity of motion planning for a particular type of linkage. However, all of these problems are typically studied in the context of linkages without obstacles, yet in robotics there are almost always obstacles. Some obstacles, such as a halfplane representing the floor, can often be avoided; but more generally the problems become much more complicated. See [Chapter 47](#).

Another area with linkage applications is *manufacturing*. Given a straight hydraulic tube or piece of wire, a typical goal is to produce a desired folded configuration. In these contexts, we want to bend the wire as little as possible. In particular, a typical constraint is to bend the wire only monotonically: once it is bent one way, it cannot be bent the other way. This constraint forces straight segments of the target shape to remain straight throughout the motion. Thus, the problem can be modeled as straightening a polygonal chain, either in 2D or 3D depending on the application, with additional constraints. For example, the expansive motions described in Section 9.3 fold all joints monotonically; however, their reliance on bending most joints simultaneously may be undesirable. Arkin et al. [AFMS01] consider the restriction in which only a single joint can be rotated at once, together with additional realistic constraints arising in wire bending.

APPLICATIONS IN BIOLOGY

A crude model of a protein backbone is a polygonal chain in 3D, and a similarly crude model of an entire protein is a polygonal tree in 3D. In both cases, the

vertices represent atoms, and the bars represent bonds between atoms (which in reality stay roughly the same length). In proteins, these bar/bond lengths are typically all within a factor of 2 of each other. Two atoms cannot occupy the same space, which can be roughly modeled by disallowing self-intersection. One interesting open problem in this context is the following:

PROBLEM 9.7.1 *Equilateral or near-equilateral locked linkages* [BDD⁺01]

Is there a locked equilateral arc, cycle, or tree in 3D? More generally, what is the smallest value of $\alpha \geq 1$ for which there is a locked arc/cycle/tree in 3D with all edge lengths between 1 and α ?

These crude models may lead to some biological insight, but they do not capture several aspects of real protein folding.

One aspect that can easily be incorporated into linkage folding is that the angles between incident bars is typically fixed. This *fixed-angle constraint* can alternatively be viewed as adding bars between vertices originally at distance two from each other. Soss et al. [Sos01, SEO03, ST00] initiated the study of such fixed-angle linkages in computational geometry, in particular establishing the NP-hardness of deciding reconfigurability or flattenability. Aloupis et al. [ADD⁺02, ADM⁺02] consider when fixed-angle linkages are not locked in the sense that all flat states are reachable from each other by motions avoiding self-intersection.

A more challenging aspect of protein folding is the *thermodynamic hypothesis* [Anf73]: that folding is encouraged to follow energy-minimizing pathways. Indeed, the bars are not strictly binding, nor are they completely fixed in length; they are merely encouraged to do so, and sometimes violate these constraints. Unfortunately, these properties are different to model, and the energy functions defined so far are either incomplete or difficult to manipulate. Also, the implications for linkage-folding problems remain unclear.

One particularly simple energy-based model of protein folding that has received substantial attention in computer science and biology is the HP (Hydrophilic-Hydrophobic) model; see, e.g., [ABD⁺, CD93, Dil90, Hay98]. This model is particularly discrete, modeling a protein as an equilateral chain on a lattice, typically a square or cubic grid, but possibly also a triangular or tetrahedral lattice. The model captures only hydrophobic bonds and forces, clustering to avoid external water. Finding the optimal folding even in this simple model is NP-complete [BL98, CGP⁺98], though there are several constant-factor approximation algorithms [HI96, New02, ABD⁺97]. One interesting open problem is whether designing a protein to fold into a particular shape is easier than finding the shape to which a particular protein folds [ABD⁺]:

PROBLEM 9.7.2 *HP protein design* [ABD⁺]

What is the complexity of deciding whether a given subset of the lattice is an optimal folding of some HP protein, and, if so, finding such a protein? What if it must be the unique optimal folding of the HP protein?

A result related to the second half of this problem is that arbitrarily long HP proteins with unique optimal foldings exist, at least for open and closed chains in a 2D square grid [ABD⁺].

9.8 SOURCES AND RELATED MATERIAL

FURTHER READING

[O'R00, Dem00, Dem02]: Surveys on folding and unfolding problems in general, which includes linkage folding in particular.

RELATED CHAPTERS

- Chapter 32: Computational topology
- Chapter 33: Computational real algebraic geometry
- Chapter 47: Algorithmic motion planning
- Chapter 48: Robotics
- Chapter 49: Computer graphics
- Chapter 55: Manufacturing processes
- Chapter 59: Geometric applications of the Grassmann-Cayley algebra
- Chapter 60: Rigidity and scene analysis
- Chapter 63: Biological applications of computational topology

REFERENCES

- [ABC⁺00] H.-K. Ahn, P. Bose, J. Czyzowicz, N. Hanusse, E. Kranakis, and P. Morin. Flipping your lid. *Geombinatorics*, 10:57–63, 2000.
- [ABD⁺] O. Aichholzer, D. Bremner, E.D. Demaine, H. Meijer, V. Sacristán, and M. Soss. Long proteins with unique optimal foldings in the H-P model. *Comput. Geom. Theory Appl.*, to appear.
- [ABD⁺97] R. Agarwala, S. Batzoglou, V. Dancik, S.E. Decatur, M. Farach, S. Hannenhalli, S. Muthukrishnan, and S. Skiena. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model. *J. Comput. Biol.*, 4:275–296, 1997.
- [ACD⁺02] O. Aichholzer, C. Cortés, E.D. Demaine, V. Dujmović, J. Erickson, H. Meijer, M. Overmars, B. Palop, S. Ramaswami, and G.T. Toussaint. Flipturning polygons. *Discrete Comput. Geom.*, 28:231–253, 2002.
- [ADD⁺02] G. Aloupis, E.D. Demaine, V. Dujmović, J. Erickson, S. Langerman, H. Meijer, I. Streinu, J. O'Rourke, M. Overmars, M. Soss, and G.T. Toussaint. Flat-state connectivity of linkages under dihedral motions. In *Proc. 13th Annu. Internat. Sympos. Algor. Comput. Lecture Notes in Comput. Sci.* 2518, pages 369–380. Springer-Verlag, New York, 2002.
- [ADE⁺01] O. Aichholzer, E.D. Demaine, J. Erickson, F. Hurtado, M. Overmars, M.A. Soss, and G.T. Toussaint. Reconfiguring convex polygons. *Comput. Geom. Theory Appl.*, 20:85–95, 2001.

- [ADM⁺02] G. Aloupis, E.D. Demaine, H. Meijer, J. O'Rourke, I. Streinu, and G.T. Toussaint. Flat-state connectedness of fixed-angle chains: Special acute chains. In *Proc. 14th Annu. Canad. Conf. Comput. Geom.*, 2002, pages 27–30.
- [AFMS01] E.M. Arkin, S.P. Fekete, J.S.B. Mitchell, and S.S. Skiena. On the manufacturability of paperclips and sheet metal structures. In *Proc. 17th Europ. Workshop Comput. Geom.*, 2001, pages 187–190.
- [AKRW03] H. Alt, C. Knauer, G. Rote, and S. Whitesides. The complexity of (un)folding. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, 2003, pages 164–170.
- [Ale95] V.A. Aleksandrov. A new example of a bendable polyhedron. *Sibirsk. Mat. Zh.*, 36:1215–1224, i, 1995.; transl. in *Siberian J. Math.*, 36:1049–1057, 1995.
- [Anf73] C.B. Anfinsen. Studies on the principles that govern the folding of protein chains. In *Les Prix Nobel en 1972*, pages 103–119. Nobel Foundation, Stockholm, 1973.
- [BDD⁺01] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. Locked and unlocked polygonal chains in three dimensions. *Discrete Comput. Geom.*, 26:269–281, 2001; full version at arXiv:cs.CG/9910009.
- [BDD⁺02] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. A note on reconfiguring tree linkages: Trees can lock. *Discrete Appl. Math.*, 117:293–297, 2002.; full version at arXiv:cs.CG/9910024.
- [Bie00] T. Biedl. Polygons needing many flipturns. Tech. Rep. CS-2000-04, Dept. of Comput. Sci., Univ. Waterloo, 2000. <ftp://cs-archive.uwaterloo.ca/cs-archive/CS-2000-04/>.
- [BL98] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (*HP*) model is NP-complete. *J. Comput. Biol.*, 5:27–40, 1998.
- [BM56] W. Blaschke and H.R. Müller. *Ebene Kinematik*. Oldenbourg, Munich, 1956.
- [BR79a] O. Bottema and B. Roth. *Theoretical Kinematics*. North-Holland, Amsterdam, 1979. Reprinted by Dover, 1990.
- [BR79b] O. Bottema and B. Roth. *Theoretical Kinematics*, volume 24 of *North-Holland Ser. Appl. Math. Mech.* North-Holland, Amsterdam, 1979.
- [Bri96] R. Bricard. Sur une question de géométrie relative aux polyèdres. *Nouv. Ann. Math.*, 15:331–334, 1896.
- [BS90] A.V. Bushmelev and I.Kh. Sabitov. Configuration spaces of Bricard octahedra (Russian). *Ukrain. Geom. Sb.*, 33:36–41, ii, 1990; transl. in *J. Soviet Math.*, 53:487–491, 1991.
- [Can87] J.F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, 1987.
- [CD93] H.S. Chan and K.A. Dill. The protein folding problem. *Phys. Today*, 46:24–32, 1993.
- [CDIO02] J.H. Cantarella, E.D. Demaine, H.N. Iben, and J.F. O'Brien. An energy-driven approach to linkage unfolding. Proc. 12th Annu. Fall Workshop Comput. Geom., DIMACS, Piscataway, 2002.
- [CDR02] R. Connelly, E.D. Demaine, and G. Rote. Infinitesimally locked self-touching linkages with applications to locked trees. In J. Calvo, K. Millett, and E. Rawdon, editors, *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in 3-Space*, pages 287–311. Amer. Math. Soc., Providence, 2002.
- [CDR03] R. Connelly, E.D. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete Comput. Geom.*, 30:205–239, 2003.

- [CGP⁺98] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *J. Comput. Biol.*, 5, 1998.
- [CH88] G.M. Crippen and T.F. Havel. *Distance Geometry and Molecular Conformation*, volume 15 of *Chemometrics Series*. Research Studies Press, Chichester, 1988.
- [Che02] C.-H. Chen. Kinemato-geometrical methodology for analyzing curvature and torsion of trajectory curve and its applications. *Mech. Mach. Theory*, 37:35–47, 2002.
- [CJ] R. Connelly and B. Jaggi. Unpublished.
- [CJ98] J. Cantarella and H. Johnston. Nontrivial embeddings of polygonal intervals and unknots in 3-space. *J. Knot Theory Ramifications*, 7:1027–1039, 1998.
- [CKM⁺01] J.A. Calvo, D. Krizanc, P. Morin, M. Soss, and G. Toussaint. Convexifying polygons with simple projections. *Inform. Process. Lett.*, 80:81–86, 2001.
- [CO01] R. Cocan and J. O'Rourke. Polygonal chains cannot lock in 4D. *Discrete Comput. Geom.*, 20:105–129, 2001.
- [Con78] R. Connelly. The rigidity of suspensions. *J. Differential Geom.*, 13:399–408, 1978.
- [Con79] R. Connelly. The rigidity of polyhedral surfaces. *Math. Mag.*, 52:275–283, 1979.
- [CP91] C.R. Calladine and S. Pellegrino. First-order infinitesimal mechanisms. *Internat. J. Solids Structures*, 27:505–515, 1991.
- [Cri92] G.M. Crippen. Exploring the conformation space of cycloalkanes by linearized embedding. *J. Comput. Chem.*, 13:351–361, 1992.
- [Dem00] E.D. Demaine. Folding and unfolding linkages, paper, and polyhedra. In *Proc. 3rd Japan Conf. Discrete Comput. Geom.*, volume 2098 of *Lecture Notes in Comput. Sci.*, pages 113–124. Springer-Verlag, New York, 2001.
- [Dem02] E.D. Demaine. *Folding and Unfolding*. Ph.D. thesis, Dept. of Comput. Sci., Univ. Waterloo, 2002.
- [Dil90] K.A. Dill. Dominant forces in protein folding. *Biochemistry*, 29:7133–7155, 1990.
- [DLOS02] E.D. Demaine, S. Langerman, J. O'Rourke, and J. Snoeyink. Interlocked open linkages with few joints. In *Proc. 18th ACM Sympos. Comput. Geom.*, 2002, pages 189–198.
- [DLOS03] E.D. Demaine, S. Langerman, J. O'Rourke, and J. Snoeyink. Interlocked open and closed linkages with few joints. *Comput. Geom. Theory Appl.*, 26:37–45, 2003.
- [Erd35] P. Erdős. Problem 3763. *Amer. Math. Monthly*, 42:627, 1935.
- [FHM⁺01] T. Fevens, A. Hernandez, A. Mesa, P. Morin, M. Soss, and G. Toussaint. Simple polygons with an infinite sequence of deflations. *Beitr. Algebra Geom.*, 42:307–311, 2001.
- [GHM97] C.G. Gibson, C.A. Hobbs, and W.L. Marar. On versal unfoldings of singularities for general two-dimensional spatial motions. *Acta Appl. Math.*, 47:221–242, 1997.
- [GN86] C.G. Gibson and P.E. Newstead. On the geometry of the planar 4-bar mechanism. *Acta Appl. Math.*, 7:113–135, 1986.
- [Grü95] B. Grünbaum. How to convexify a polygon. *Geombinatorics*, 5:24–30, 1995.
- [GZ01] B. Grünbaum and J. Zaks. Convexification of polygons by flips and by flipturns. *Discrete Math.*, 241:333–342, 2001.
- [Har74] H. Hart. On certain conversions of motion. *Messenger Math.*, IV:82–88, 1874.
- [Hau91] J.-C. Hausmann. Sur la topologie des bras articulés. In *Algebraic Topology Poznań 1989*, volume 1474 of *Lecture Notes in Math.*, pages 146–159. Springer-Verlag, Berlin, 1991.

- [Hav91] T.F. Havel. Some examples of the use of distances as coordinates for Euclidean geometry. In B. Sturmfels and N. White, editors, *Invariant-Theoretic Algorithms in Geometry*, *J. Symbolic Comput.*, 11:579–593, 1991.
- [Hay98] B. Hayes. Prototeins. *Amer. Sci.*, 86:216–221, 1998.
- [HI96] W.E. Hart and S. Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *J. Comput. Biol.*, 3:53–96, 1996.
- [HJW84] J. Hopcroft, D. Joseph, and S. Whitesides. Movement problems for 2-dimensional linkages. *SIAM J. Comput.*, 13:610–629, 1984.
- [HJW85] J. Hopcroft, D. Joseph, and S. Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM J. Comput.*, 14:315–333, 1985.
- [Hun78] K.H. Hunt. *Kinematic Geometry of Mechanisms*. Oxford Engrg. Sci. Ser., Clarendon, Oxford Univ. Press, New York, 1978.
- [Jag92] B. Jaggi. *Punktmengen mit vorgeschriebenen Distanzen und ihre Konfigurationsräume*. Inauguraldissertation, Univ. Bern, 1992.
- [JP85] D.A. Joseph and W.H. Plantings. On the complexity of reachability and motion planning questions. In *Proc. 1st ACM Sympos. Comput. Geom.*, 1985, pages 62–66.
- [JS99] D. Jordan and M. Steiner. Configuration spaces of mechanical linkages. *Discrete Comput. Geom.*, 22:297–315, 1999.
- [Kam99] Y. Kamiyama. Topology of equilateral polygon linkages in the Euclidean plane modulo isometry group. *Osaka J. Math.*, 36:731–745, 1999.
- [Kem76] A.B. Kempe. On a general method of describing plane curves of the n th degree by linkwork. *Proc. London Math. Soc.*, 7:213–216, 1876.
- [Kem77] A.B. Kempe. *How to Draw a Straight Line: A Lecture on Linkages*. Macmillan, London, 1877.
- [Kin] H.C. King. Configuration spaces of linkages in \mathbb{R}^n . arXiv:math.GT/9811138.
- [Kin99] H.C. King. Planar linkages and algebraic sets. *Turkish J. Math.*, 23:33–56, 1999.
- [KM95] M. Kapovich and J. Millson. On the moduli space of polygons in the Euclidean plane. *J. Differential Geom.*, 42:133–164, 1995.
- [KM02] M. Kapovich and J.J. Millson. Universality theorems for configuration spaces of planar linkages. *Topology*, 41:1051–1107, 2002.
- [KT99] Y. Kamiyama and M. Tezuka. Topology and geometry of equilateral polygon linkages in the Euclidean plane. *Quart. J. Math. Oxford Ser. (2)*, 50:463–470, 1999.
- [Leb67] H. Lebesgue. Octaèdres articulés de Bricard. *Enseign. Math. (2)*, 13:175–185, 1967.
- [Ler00] J. Lerbet. Some explicit relations in kinematics of mechanisms. *Mech. Res. Comm.*, 27:621–630, 2000.
- [Lip71] L. Lipkin. Dispositif articulé pour la transformation rigoureuse du mouvement circulaire en mouvement rectiligne. *Rev. Univers. Mines Métall. Liège*, 30:149–150, 1871.
- [LW95] W.J. Lenhart and S.H. Whitesides. Reconfiguring closed polygonal chains in Euclidean d -space. *Discrete Comput. Geom.*, 13:123–140, 1995.
- [McC90] J.M. McCarthy. *An Introduction to Theoretical Kinematics*. MIT Press, Cambridge, 1990.
- [McC00] J.M. McCarthy. *Geometric Design of Linkages*, volume 11 of *Interdisciplinary Appl. Math.* Springer-Verlag, New York, 2000.

- [Mik01] S.N. Mikhalev. Some necessary metric conditions for the flexibility of suspensions (Russian). *Vestnik Moskov. Univ. Ser. I Mat. Mekh.*, 3:15–21, 77, 2001.; transl. in *Moscow Univ. Mat. Bull.*, 56:14–20, 2001.
- [Mnë88] N.E. Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In O.Ya. Viro, editor, *Topology and Geometry—Rohlin Seminar*, volume 1346 of *Lecture Notes in Math.*, pages 527–544. Springer-Verlag, Berlin, 1988.
- [MS00] O. Mermoud and M. Steiner. Visualisation of configuration spaces of polygonal linkages. *J. Geom. Graph.*, 4:147–157, 2000.
- [Nag39] B. Sz.-Nagy. Solution to problem 3763. *Amer. Math. Monthly*, 46:176–177, 1939.
- [New02] A. Newman. A new algorithm for protein folding in the HP model. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algor.*, 2002, pages 876–884.
- [O'R00] J. O'Rourke. Folding and unfolding in computational geometry. In *Revised Papers from the Japan Conf. Discrete Comput. Geom.*, volume 1763 of *Lecture Notes in Comput. Sci.*, pages 258–266. Springer-Verlag, New York, 2000.
- [Pea73] A. Peaucellier. Note sur une question de géometrie de compas. *Nouv. Ann. de Math.*, 2e serie, XII:71–73, 1873.
- [Pot94] H. Pottmann. Kinematische Geometrie. In O. Giering and J. Hoschek, editors, *Geometrie und ihre Anwendungen*, pages 141–175. Hanser, Munich, 1994.
- [Rob75] S. Roberts. On three-bar motion in plane space. *Proc. London Math. Soc.*, 7:14–23, 1875.
- [SEO03] M. Soss, J. Erickson, and M. Overmars. Preprocessing chains for fast dihedral rotations is hard or even impossible. *Comput. Geom. Theory Appl.*, 26:235–246, 2003.
- [Sos01] M. Soss. *Geometric and Computational Aspects of Molecular Reconfiguration*. Ph.D. thesis, School of Computer Science, McGill Univ., Montreal, 2001.
- [ST00] M. Soss and G.T. Toussaint. Geometric and computational aspects of polymer reconfiguration. *J. Math. Chem.*, 27:303–318, 2000.
- [Sta97] H. Stachel. Euclidean line geometry and kinematics in the 3-space. In *Proc. 4th Internat. Congr. Geom. (Thessaloniki, 1996)*, pages 380–391. Giachoudis-Giapoulis, Thessaloniki, 1997.
- [Sta99] H. Stachel. Higher order flexibility of octahedra. In K. Bezdek and R. Connelly, editors, *Discrete Geometry and Rigidity (Budapest, 1999)*, *Period. Math. Hungar.*, 39:225–240, 1999.
- [Str00] I. Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proc. 41st Annu. IEEE Sympos. Found. Comput. Sci.*, 2000, pages 443–453.
- [Tou99] G. Toussaint. The Erdős-Nagy theorem and its ramifications. In *Proc. 11th Canad. Conf. Comput. Geom.*, 1999, pages 9–12. Long version at http://www.cs.ubc.ca/conferences/CCCG/elec_proc/fp19.ps.gz.
- [Tou01] G. Toussaint. A new class of stuck unknots in pol_6 . *Beitr. Algebra Geom.*, 42:1027–1039, 2001.
- [TW84] W. Thurston and J. Weeks. The mathematics of three-dimensional manifolds. *Sci. Amer.*, July 1984, pages 108–120.
- [Weg93] B. Wegner. Partial inflation of closed polygons in the plane. *Beitr. Algebra Geom.*, 34:77–85, 1993.

10 GEOMETRIC GRAPH THEORY

János Pach

INTRODUCTION

In the traditional areas of graph theory (Ramsey theory, extremal graph theory, random graphs, etc.), graphs are regarded as abstract binary relations. The relevant methods are often incapable of providing satisfactory answers to questions arising in geometric applications. Geometric graph theory focuses on combinatorial and geometric properties of graphs drawn in the plane by straight-line edges (or, more generally, by edges represented by simple Jordan arcs). It is a fairly new discipline abounding in open problems, but it has already yielded some striking results that have proved instrumental in the solution of several basic problems in combinatorial and computational geometry (including the k -set problem and metric questions discussed in Sections 1.1 and 1.2, respectively, of this Handbook). This chapter is partitioned into extremal problems (Section 10.1), crossing numbers (Section 10.2), and generalizations (Section 10.3).

10.1 EXTREMAL PROBLEMS

Turán's classical theorem [Tur54] determines the maximum number of edges that an abstract graph with n vertices can have without containing, as a subgraph, a complete graph with k vertices. In the spirit of this result, one can raise the following general question. Given a class \mathcal{H} of so-called *forbidden geometric subgraphs*, what is the maximum number of edges that a geometric graph of n vertices can have without containing a geometric subgraph belonging to \mathcal{H} ? Similarly, Ramsey's theorem [Ram30] for abstract graphs has some natural analogues for geometric graphs. In this section we will be concerned mainly with problems of these two types.

GLOSSARY

Geometric graph: A graph drawn in the plane by (possibly crossing) straight-line segments; i.e., a pair $(V(G), E(G))$, where $V(G)$ is a set of points ('vertices'), no three of which are collinear, and $E(G)$ is a set of segments ('edges') whose endpoints belong to $V(G)$.

Convex geometric graph: A geometric graph whose vertices are in *convex position*; i.e., they form the vertex set of a convex polygon.

Cyclic chromatic number of a convex geometric graph: The minimum number $\chi_c(G)$ of colors needed to color all vertices of G so that each color class consists of consecutive vertices along the boundary of the convex hull of the vertex set.

Convex matching: A convex geometric graph consisting of disjoint edges, each of which belongs to the boundary of the convex hull of its vertex set.

Parallel matching: A convex geometric graph consisting of disjoint edges, the convex hull of whose vertex set contains only two of the vertices on its boundary.

Complete geometric graph: A geometric graph G whose edge set consists of all $\binom{|V(G)|}{2}$ segments between its vertices.

Complete bipartite geometric graph: A geometric graph G with $V(G) = V_1 \cup V_2$, whose edge set consists of all segments between V_1 and V_2 .

Geometric subgraph of G : A geometric graph H , for which $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Crossing: A common interior point of two edges of a geometric graph.

(k, l) -grid: $k + l$ vertex-disjoint edges in a geometric graph such that each of the first k edges crosses all of the last l edges.

Disjoint edges: Edges of a geometric graph that do not cross and do not even share an endpoint.

Parallel edges: Edges of a geometric graph whose supporting lines are parallel or intersect at points not belonging to any of the edges (including their endpoints).

x -monotone curve: A continuous curve that intersects every vertical line in at most one point.

Outerplanar graph: A (planar) graph that can be drawn in the plane without crossing so that all points representing its vertices lie on the outer face of the resulting subdivision of the plane. A **maximal outerplanar graph** is a triangulated cycle.

Hamiltonian path: A path going through all elements of a finite set S . If the elements of S are colored by two colors, and no two adjacent elements of the path have the same color, then it is called an *alternating* path.

Hamiltonian cycle: A cycle going through all elements of a finite set S .

Caterpillar: A tree consisting of a path P and of some extra edges, each of which is adjacent to a vertex of P .

CROSSING-FREE GEOMETRIC GRAPHS

1. Hanani's theorem: Any graph that can be drawn in the plane so that its edges are represented by simple Jordan arcs any two of which either share an endpoint or properly cross an even number of times is planar [Cho34].
2. Fáry's theorem: Every planar graph admits a crossing-free straight-line drawing [Fár48, Tut60, Ste22]. Moreover, every 3-connected planar graph and its dual have simultaneous straight-line drawings in the plane such that only dual pairs of edges cross and every such pair is perpendicular [BS93].
3. Koebe's theorem: The vertices of every planar graph can be represented by nonoverlapping disks in the plane such that two of them are tangent to each other if and only if the corresponding two vertices are adjacent [Koe36, Thu78]. This immediately implies Fáry's theorem.
4. Pach-Tóth theorem: Any graph that can be drawn in the plane so that its edges are represented by x -monotone curves with the property that any two of them either share an endpoint or properly cross an even number of times admits a crossing-free straight-line drawing, in which the x -coordinates of the vertices remain the same [PT03].

5. Grid drawings of planar graphs: Every planar graph of n vertices admits a straight-line drawing such that the vertices are represented by points belonging to an $(n-1) \times (n-1)$ grid [dFPP90, Sch90]. Furthermore, such a drawing can be found in $O(n)$ time.
6. Straight-line drawings of outerplanar graphs: For any outerplanar graph H with n vertices and for any set P of n points in the plane in general position, there is a crossing-free geometric graph G with $V(G) = P$, whose underlying graph is isomorphic to H [GMPP91]. For any rooted tree T and for any set P of $|V(T)|$ points in the plane in general position with a specified element $p \in P$, there is a crossing-free straight-line drawing of T such that every vertex of T is represented by an element of P and the root is represented by p [IPTT94]. This theorem generalizes to any *pair* of rooted trees, T_1 and T_2 : for any set P of $n = |V(T_1)| + |V(T_2)|$ points in general position in the plane, there is a crossing-free mapping of $T_1 \cup T_2$ that takes the roots to arbitrarily prespecified elements of P . Such a mapping can be found in $O(n^2 \log n)$ time [KK00]. The analogous statement for *triples* of trees is false.
7. Alternating paths: Given n red points and n blue points in general position in the plane, separated by a straight line, they always admit a noncrossing alternating Hamiltonian path [KK03].

TURÁN-TYPE PROBLEMS

By Euler's Polyhedral Formula, if a geometric graph G with $n \geq 3$ vertices has no 2 crossing edges, it cannot have more than $3n - 6$ edges. It was shown in [AAP⁺97] that under the weaker condition that no 3 edges are *pairwise crossing*, the number of edges of G is still $O(n)$. It is not known whether this statement remains true even if we assume only that no 4 edges are pairwise crossing. As for the analogous problem when the forbidden configuration consists of k pairwise *disjoint edges*, the answer is linear for every k [PT94]. In particular, for $k = 2$, the number of edges of G cannot exceed the number of vertices [HP34]. The best lower and upper bounds known for the number of edges of a geometric graph with n vertices, containing no *forbidden* geometric subgraph of a certain type, are summarized in [Table 10.1.1](#). The letter k always stands for a *fixed* positive integer parameter and n tends to infinity. Wherever k does not appear in the asymptotic bounds, it is hidden in the constants involved in the O - and Ω -notations.

Better results are known for *convex* geometric graphs, i.e., when the vertices are in convex position. The relevant bounds are listed in [Table 10.1.2](#). For any convex geometric graph G , let $\chi_c(G)$ denote its *cyclic chromatic number*. Furthermore, let $\text{ex}(n, K_k)$ stand for the maximum number of edges of a graph with n vertices that does not have a complete subgraph with k vertices. By Turán's theorem [Tur54] mentioned above, $\text{ex}(n, K_k) = \frac{k-2}{k-1} \binom{n}{2} + O(n)$ is equal to the number of edges of a complete $(k-1)$ -partite graph with n vertices whose vertex classes are of size $\lfloor n/(k-1) \rfloor$ or $\lceil n/(k-1) \rceil$. Two disjoint self-intersecting paths of length 3, $xyvz$ and $x'y'v'z'$, in a convex geometric graph are said to be of the *same orientation* if the cyclic order of their vertices is $x, v, x', v', y', z', y, z$ ($\bowtie \bowtie$). They are said to have *opposite orientations* if the cyclic order of their vertices is $x, v, v', x', z', y', y, z$ (*type 1*: $\bowtie \bowtie$) or $v, x, x', v', y', z', z, y$ (*type 2*: $\bowtie \bowtie$).

TABLE 10.1.1 Maximum number of edges of a geometric graph of n vertices containing no forbidden subconfigurations of a certain type.

FORBIDDEN CONFIGURATION	LOWER BOUND	UPPER BOUND	SOURCE
2 crossing edges	$3n - 6$	$3n - 6$	Euler
3 pairwise crossing edges	$\Omega(n)$	$O(n)$	[AAP ⁺ 97]
$k > 3$ pairwise crossing edges	$\Omega(n)$	$O(n \log n)$	[Val98]
an edge crossing 2 others	$4n - 9$	$4n - 9$	[PT97]
an edge crossing 3 others	$5n - 12$	$5n - 10$	[PT97]
an edge crossing 4 others	$5.5n + \Omega(1)$	$5.5n + O(1)$	[PRTT04]
an edge crossing k others	$\Omega(\sqrt{kn})$	$O(\sqrt{kn})$	[PT97]
2 crossing edges crossing k others	$\Omega(n)$	$O(n)$	Pach-Radoičić-Tóth
(k, l) -grid	$\Omega(n)$	$O(n)$	[PPST]
self-intersecting path of length 3	$\Omega(n \log n)$	$O(n \log n)$	[PPTT02]
self-intersecting path of length 5	$\Omega(n \log \log n)$	$O(n \log n / \log \log n)$	Tardos, [PPTT02]
self-intersecting cycle of length 4	$\Omega(n^{3/2})$	$O(n^{8/5})$	[PR03]
2 disjoint edges	n	n	[HP34]
noncrossing path of length k	$\Omega(kn)$	$O(k^2 n)$	[Tót00]
k pairwise parallel edges	$\Omega(n)$	$O(n)$	[Val98]

FIGURE 10.1.1

Geometric graph with $n = 20$ vertices and $5n - 12 = 88$ edges, none of which crosses 3 others.

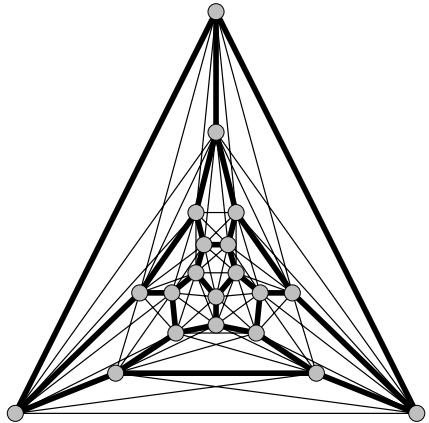


FIGURE 10.1.2

Convex geometric graph with $n = 13$ vertices and $6n - \binom{7}{2} = 57$ edges, no 4 of which are pairwise crossing [CP92].

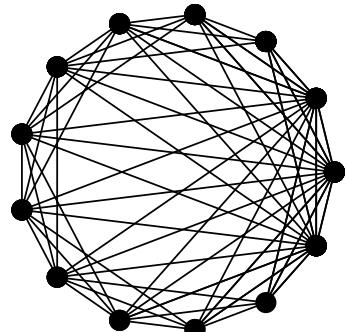


TABLE 10.1.2 Maximum number of edges of a *convex* geometric graph of n vertices containing no forbidden subconfigurations of a certain type.

FORBIDDEN CONFIGURATION	LOWER BOUND	UPPER BOUND	SOURCE
2 crossing edges	$2n - 3$	$2n - 3$	Euler
self-intersecting path of length 3	$2n - 3$	$2n - 3$	Perles
k self-intersecting paths of length 3 with the same orientation	$\Omega(n)$	$O(n)$	[BKV03]
2 self-intersecting paths of length 3 with opposite orientations of type 1	$\Omega(n \log n)$	$O(n \log n)$	[BKV03]
2 self-intersecting paths of length 3 with opposite orientations of type 2	$\Omega(n \log n)$	$O(n \log n)$	[BKV03]
2 adjacent edges crossing a 3rd	$\lfloor 5n/2 - 4 \rfloor$	$\lfloor 5n/2 - 4 \rfloor$	Perles-Pinhasi, [BKV03]
k pairwise crossing edges	$2(k-1)n - \binom{2k-1}{2}$	$2(k-1)n - \binom{2k-1}{2}$	[CP92]
noncrossing outerplanar graph of k vertices, having a Hamiltonian cycle	$\text{ex}(n, K_k)$	$\text{ex}(n, K_k)$	Pach [PA95], Perles
convex geometric subgraph G	$\text{ex}(n, K_{\chi_c(G)})$	$\text{ex}(n, K_{\chi_c(G)}) + o(n^2)$	[BKV03]
convex matching of k disjoint edges	$\text{ex}(n, K_k) + n - k + 1$	$\text{ex}(n, K_k) + n - k + 1$	[KP96]
parallel matching of k disjoint edges	$(k-1)n$	$(k-1)n$	[Kup84]
noncrossing caterpillar C of k vertices	$\lfloor (k-2)n/2 \rfloor$	$\lfloor (k-2)n/2 \rfloor$	Perles [BKV03]

RAMSEY-TYPE PROBLEMS

In classical Ramsey theory, one wants to find large monochromatic subgraphs in a complete graph whose edges are colored with several colors [GRS90]. Most questions of this type can be generalized to complete geometric graphs, where the monochromatic subgraphs are required to satisfy certain geometric conditions.

1. Karolyi-Pach-Toth theorem [KPT97]: If the edges of a finite complete geometric graph are colored by two colors, there exists a noncrossing spanning tree, all of whose edges are of the same color. (This statement was conjectured by Bialostocki and Dierker [BV]. The analogous assertion for abstract graphs follows from the fact that any graph or its complement is connected.)
2. Geometric Ramsey numbers: Let $\mathcal{G}_1, \dots, \mathcal{G}_k$ be not necessarily different classes of geometric graphs. Let $R(\mathcal{G}_1, \dots, \mathcal{G}_k)$ denote the smallest positive number R with the property that any complete geometric graph of R vertices whose edges are colored with k colors ($1, \dots, k$, say) contains, for some i , an i -colored subgraph belonging to \mathcal{G}_i . If $\mathcal{G}_1 = \dots = \mathcal{G}_k = \mathcal{G}$, we write $R(\mathcal{G}; k)$ instead of $R(\mathcal{G}_1, \dots, \mathcal{G}_k)$. If $k = 2$, for the sake of simplicity, let $R(\mathcal{G})$ stand for $R(\mathcal{G}; 2)$. Some known results on the numbers $R(\mathcal{G}_1, \mathcal{G}_2)$ are listed in Table 10.1.3. In line 3 of the table, we have a better result if we restrict our attention to *convex* geometric graphs: For any 2-coloring of the edges of a complete convex geometric graph with $2k - 1$ vertices, there exists a noncrossing monochromatic path of length $k \geq 2$, and this result cannot be improved. The bounds in line 4 also hold when $\mathcal{G}_1 = \mathcal{G}_2$ consists of all noncrossing cycles of length k , triangulated from one of their vertices. The geometric Ramsey numbers of convex geometric graphs, when $\mathcal{G}_1 = \mathcal{G}_2$ consists of all isomorphic copies of a given convex geometric graph with at most 4 vertices, can be found in [BH96].

TABLE 10.1.3 Geometric Ramsey numbers $R(\mathcal{G}_1, \mathcal{G}_2)$ from [KPT97] and [KPTV98].

\mathcal{G}_1	\mathcal{G}_2	LOWER BOUND	UPPER BOUND
all noncrossing trees of k vertices	all noncrossing trees of k vertices	k	k
k disjoint edges	l disjoint edges	$k + l + \max\{k, l\} - 1$	$k + l + \max\{k, l\} - 1$
noncrossing paths of length k	noncrossing paths of length k	$\Omega(k)$	$O(k^{3/2})$
noncrossing cycles of length k	noncrossing cycles of length k	$(k - 1)^2$	$2(k - 1)(k - 2) + 2$

3. Pairwise disjoint copies: For any positive integer k , let $k\mathcal{G}$ denote the class of all geometric graphs that can be obtained by taking the union of k pairwise disjoint members of \mathcal{G} . If k is a power of 2 then

$$R(k\mathcal{G}) \leq (R(\mathcal{G}) + 1)k - 1.$$

In particular, if $\mathcal{G} = \mathcal{T}$ is the class of triangles, we have $R(\mathcal{T}) = 6$. Thus, the above bound yields that

$$R(k\mathcal{T}) \leq 7k - 1,$$

provided that k is a power of 2. This result cannot be improved [KPTV98].

Furthermore, for any $k > 0$, we have

$$R(k\mathcal{G}) \leq \left\lceil \frac{3(R(\mathcal{G}) + 1)}{2} \right\rceil k - \left\lceil \frac{R(\mathcal{G}) + 1}{2} \right\rceil.$$

For the corresponding quantities for convex geometric graphs, we have

$$R_c(k\mathcal{G}) \leq (R_c(\mathcal{G}) + 1)k - 1.$$

4. Constructive vertex- and edge-Ramsey numbers: Given a class of geometric graphs \mathcal{G} , let $R_v(\mathcal{G})$ denote the smallest number R such that there exists a (complete) geometric graph of R vertices that, for any 2-coloring of its edges, has a monochromatic subgraph belonging to \mathcal{G} . Similarly, let $R_e(\mathcal{G})$ denote the minimum number of edges of a geometric graph with this property. $R_v(\mathcal{G})$ and $R_e(\mathcal{G})$ are called the **vertex-** and **edge-Ramsey number** of \mathcal{G} , respectively. Clearly, we have

$$R_v(\mathcal{G}) \leq R(\mathcal{G}), \quad R_e(\mathcal{G}) \leq \binom{R(\mathcal{G})}{2}.$$

(For abstract graphs, similar notions are discussed in [EFRS78, Bec83].)

For \mathcal{P}_k , the class of noncrossing paths of length k , we have $R_v(\mathcal{P}_k) = O(k^{3/2})$ and $R_e(\mathcal{P}_k) = O(k^2)$.

OPEN PROBLEMS

- What is the smallest number $u = u(n)$ such that there exists a “universal” set U of u points in the plane with the property that every planar graph of n vertices admits a noncrossing straight-line drawing on a suitable subset of U [dFPP90]? It follows from the existence of a small grid drawing (see above) that $u(n) \leq n^2$. From below we have only $u(n) > 1.01n$.

2. Can the vertices of every planar graph G be represented by straight-line segments in the plane so that two segments intersect if and only if the corresponding vertices are adjacent? The answer is known to be in the affirmative if the chromatic number of G is 2 [dFdMP94] or 3 (de Fraysseix-de Mendez).
 3. (Erdős, Kaneko-Kano) What is the largest number $A = A(n)$ such that any set of n red and n blue points in the plane admits a noncrossing alternating path of length A ? It is known that $A(n) \leq (4/3 + o(1))n$.
 4. Is it true that, for any fixed k , the maximum number of edges of a geometric graph with n vertices that does not have k pairwise crossing edges is $O(n)$?
 5. (Aronov et al.) Is it true that any complete geometric graph with n vertices has at least $\Omega(n)$ pairwise crossing edges? It was shown in [AEG⁺94] that one can always find $\sqrt{n/12}$ pairwise crossing edges. On the other hand, any complete geometric graph with n vertices has a noncrossing Hamiltonian path, hence $\lfloor n/2 \rfloor$ pairwise disjoint edges.
 6. (Larman-Matoušek-Pach-Töröcsik) What is the smallest positive number $r = r(n)$ such that any family of r closed segments in general position in the plane has n members that are either pairwise disjoint or pairwise crossing? It is known [LMPT94, KPT97] that $n^{\log 5/\log 2} \approx n^{2.322} \leq r(n) \leq n^5$.
-

10.2 CROSSING NUMBERS

The investigation of crossing numbers started during WWII with Turán's Brick Factory Problem [Tur77]: how should one redesign the routes of railroad tracks between several kilns and storage places in a brick factory so as to minimize the number of crossings? In the early 1980s, it turned out that the chip area required for the realization (VLSI layout) of an electrical circuit is closely related to the crossing number of the underlying graph [Lei83]. This discovery gave an impetus to research in the subject. More recently, it has been realized that general bounds on crossing numbers can be used to solve a large variety of problems in discrete and computational geometry.

GLOSSARY

Drawing of a graph: A representation of the graph in the plane such that its vertices are represented by distinct points and its edges by simple continuous arcs connecting the corresponding point pairs. In a drawing (a) no edge passes through any vertex other than its endpoints, (b) no two edges touch each other (i.e., if two edges have a common interior point, then at this point they properly cross each other), and (c) no three edges cross at the same point.

Crossing: A common interior point of two edges in a graph drawing. Two edges may have several crossings.

Crossing number of a graph: The smallest number of crossings in any drawing of G , denoted by $\text{CR}(G)$. Clearly, $\text{CR}(G) = 0$ if and only if G is planar.

Rectilinear crossing number: The minimum number of crossings in a drawing of G in which every edge is represented by a straight-line segment. It is denoted by $\text{LIN-CR}(G)$.

Pairwise crossing number: The minimum number of crossing pairs of edges over all drawings of G , denoted by $\text{PAIR-CR}(G)$. (Here the edges can be represented by arbitrary continuous curves, so that two edges may cross more than once, but every pair of edges can contribute at most one to $\text{PAIR-CR}(G)$.)

Odd crossing number: The minimum number of those pairs of edges that cross an odd number of times, over all drawings of G . It is denoted by $\text{ODD-CR}(G)$.

Biplanar crossing number: The minimum of $\text{CR}(G_1) + \text{CR}(G_2)$ over all partitions of the graph into two edge-disjoint subgraphs G_1 and G_2 .

Bisection width: The minimum number $b(G)$ of edges whose removal splits the graph G into two roughly equal subgraphs. More precisely, $b(G)$ is the minimum number of edges running between V_1 and V_2 over all partitions of the vertex set of G into two disjoint parts $V_1 \cup V_2$ such that $|V_1|, |V_2| \geq |V(G)|/3$.

Cut width: The minimum number $c(G)$ such that there is a drawing of G in which no two vertices have the same x -coordinate and every vertical line crosses at most $c(G)$ edges.

Path width: The minimum number $p(G)$ such that there is a sequence of at most $(p(G) + 1)$ -element sets $V_1, V_2, \dots, V_r \subseteq V(G)$ with the property that both endpoints of every edge belong to some V_i and, if a vertex occurs in V_i and V_k ($i < k$), then it also belongs to every V_j , $i < j < k$.

GENERAL ESTIMATES

Garey and Johnson [GJ83] showed that the determination of the crossing number is an *NP-complete* problem. Analogous results hold for the rectilinear crossing number [Bie91], for the pair crossing number [SSS02], and for the odd crossing number [PT00b]. The exact determination of crossing numbers of relatively small graphs of a simple structure (such as complete or complete bipartite graphs) is a hopelessly difficult task, but there are several useful bounds. There is an algorithm [EGS03] for computing a drawing of a bounded-degree graph with n vertices, for which n plus the number of crossings is $O(\log^3 n)$ times the optimum.

1. For a simple graph G with $n \geq 3$ vertices and e edges, $\text{CR}(G) \geq e - 3n + 6$. From this inequality, a simple probabilistic argument shows that $\text{CR}(G) \geq ce^3/n^2$, for a suitable positive constant c . This important bound, due to Ajtai-Chvátal-Newborn-Szemerédi [ACNS82] and, independently, to Leighton [Lei83], is often referred to as the **crossing lemma**. We know that $0.03 \leq c \leq 0.09$ [PT97, PRTT04]. The lower bound follows from line 6 in [Table 10.1.1](#). Similar statements hold for $\text{PAIR-CR}(G)$ and $\text{ODD-CR}(G)$ [PT00b].
2. Crossing lemma for multigraphs [Szé97]: Let G be a **multigraph** with n vertices and e edges, i.e., the same pair of vertices can be connected by more than one edge. Let m denote the maximum multiplicity of an edge. Then

$$\text{CR}(G) \geq c \frac{e^3}{mn^2} - m^2 n,$$

where c denotes the same constant as in the previous paragraph.

3. Midrange crossing constant: Let $\kappa(n, e)$ denote the minimum crossing number of a graph G with n vertices and at least e edges. That is,

$$\kappa(n, e) = \min_{\substack{n(G) = n \\ e(G) \geq e}} \text{CR}(G).$$

It follows from the crossing lemma that, for $e \geq 4n$, $\kappa(n, e)n^2/e^3$ is bounded from below and from above by two positive constants. Erdős and Guy [EG73] conjectured that if $e \gg n$ then $\lim \kappa(n, e)n^2/e^3$ exists. (We use the notation $f(n) \gg g(n)$ to mean that $\lim_{n \rightarrow \infty} f(n)/g(n) = \infty$.) This was partially settled in [PST00]: if $n \ll e \ll n^2$, then

$$\lim_{n \rightarrow \infty} \kappa(n, e) \frac{n^2}{e^3} = C > 0$$

exists. Moreover, the same result is true with the same constant C , for drawings on every other orientable surface.

4. Graphs with monotone properties: A graph property \mathcal{P} is said to be **monotone** if (i) for any graph G satisfying \mathcal{P} , every subgraph of G also satisfies \mathcal{P} ; and (ii) if G_1 and G_2 satisfy \mathcal{P} , then their disjoint union also satisfies \mathcal{P} . For any monotone property \mathcal{P} , let $\text{ex}(n, \mathcal{P})$ denote the maximum number of edges that a graph of n vertices can have if it satisfies \mathcal{P} . In the special case when \mathcal{P} is the property that the graph does not contain a subgraph isomorphic to a fixed forbidden subgraph H , we write $\text{ex}(n, H)$ for $\text{ex}(n, \mathcal{P})$.

Let \mathcal{P} be a monotone graph property with $\text{ex}(n, \mathcal{P}) = O(n^{1+\alpha})$ for some $\alpha > 0$. In [PST00], it was proved that there exist two constants $c, c' > 0$ such that the crossing number of any graph G with property \mathcal{P} that has n vertices and $e \geq cn \log^2 n$ edges satisfies

$$\text{CR}(G) \geq c' \frac{e^{2+1/\alpha}}{n^{1+1/\alpha}}.$$

This bound is asymptotically tight, up to a constant factor. In particular, if $e > 4n$ and G has no cycle of length at most $2r$, then the crossing number of G satisfies

$$\text{CR}(G) \geq c_r \frac{e^{r+2}}{n^{r+1}},$$

where $c_r > 0$ is a suitable constant. For $r = 2, 3$, and 5 , these bounds are asymptotically tight, up to a constant factor. If G does not contain a complete bipartite subgraph $K_{r,s}$ with r and s vertices in its classes, $s \geq r$, then we have

$$\text{CR}(G) \geq c_{r,s} \frac{e^{3+1/(r-1)}}{n^{2+1/(r-1)}},$$

where $c_{r,s} > 0$ is a suitable constant. These bounds are tight up to a constant factor if $r = 2, 3$, or if r is arbitrary and $s > (r-1)!$.

5. Crossing number vs. bisection width $b(G)$: For any vertex $v \in V(G)$, let $d(v)$ denote the degree of v in G . It was shown in [PSS96] and [SV94] that

$$\text{CR}(G) + \frac{1}{16} \sum_{v \in V(G)} d^2(v) \geq \frac{1}{40} b^2(G).$$

A similar statement holds with a worse constant for the cut width $c(G)$ of G [DV02]. This, in turn, implies that the same is true for $p(G)$, the path width of G , as we have $p(G) \leq c(G)$ for every G [Kin92].

6. Relations between different crossing numbers: Clearly, we have

$$\text{ODD-CR}(G) \leq \text{PAIR-CR}(G) \leq \text{CR}(G) \leq \text{LIN-CR}(G).$$

It was shown [BD93] that there are graphs with crossing number 4 whose rectilinear crossing numbers are arbitrarily large. On the other hand, we cannot rule out the possibility that

$$\text{ODD-CR}(G) = \text{PAIR-CR}(G) = \text{CR}(G)$$

for every graph G . It was established in [PT00b] that

$$\text{CR}(G) \leq 2(\text{ODD-CR}(G))^2.$$

Recently, Kolman and Matoušek found a slightly better upper bound on $\text{CR}(G)$, in terms of $\text{PAIR-CR}(G)$.

7. Crossing numbers of random graphs: Let $G = G(n, p)$ be a *random graph* with n vertices, whose edges are chosen independently with probability $p = p(n)$. Let e denote the *expected number* of edges of G , i.e., $e = p \cdot \binom{n}{2}$. It is not hard to see that if $e > 10n$, then almost surely $b(G) \geq e/10$. It therefore follows from the above relation between the crossing number and the bisection width that almost surely we have $\text{LIN-CR}(G) \geq \text{CR}(G) \geq e^2/4000$. Evidently, the order of magnitude of this bound cannot be improved. A similar inequality was proved in [ST02] for the pairwise crossing number, under the stronger condition that $e > n^{1+\epsilon}$ for some $\epsilon > 0$.
8. Biplanar crossing number vs. crossing number: It is known [SSV] that the biplanar crossing number of every graph is at most $3/8$ times its crossing number. The best value of the constant may be as small as $7/24$.
9. Harary-Kainen-Schwenk conjecture [HKS73]: For every $n \geq m \geq 3$ and cycles C_n and C_m , $\text{CR}(C_n \times C_m)$ is equal to $n(m-2)$. This was proved in [GS] for every m and for all sufficiently large n . For the crossing number of the skeleton of the n -dimensional hypercube Q_n , we have $1/20 + o(1) \leq \text{CR}(Q_n)/4^n \leq 163/1024$ [FdF00, SV93].

OPEN PROBLEMS

1. Is it true that $\text{ODD-CR}(G) = \text{PAIR-CR}(G) = \text{CR}(G)$ for every graph G ?
2. Zarankiewicz's conjecture [Guy69]: The crossing number of the complete bipartite graph $K_{n,m}$ with n and m vertices in its classes satisfies

$$\text{CR}(K_{n,m}) = \left\lfloor \frac{m}{2} \right\rfloor \cdot \left\lfloor \frac{m-1}{2} \right\rfloor \cdot \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{n-1}{2} \right\rfloor.$$

Kleitman [Kle70] verified this conjecture in the special case when $\min\{m, n\} \leq 6$ and Woodall [Woo93] for $m = 7, n \leq 10$.

It is also conjectured that the crossing number of the complete graph K_n satisfies

$$\text{CR}(K_n) = \frac{1}{4} \left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{n-1}{2} \right\rfloor \cdot \left\lfloor \frac{n-2}{2} \right\rfloor \cdot \left\lfloor \frac{n-3}{2} \right\rfloor.$$

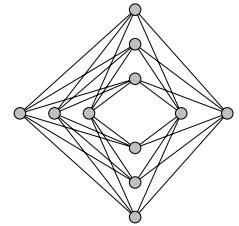


FIGURE 10.2.1

Complete bipartite graph $K_{5,6}$ with 24 crossings.

3. Rectilinear crossing numbers of complete graphs: Determine the value

$$\kappa = \lim_{n \rightarrow \infty} \frac{\text{LIN-CR}(K_n)}{\binom{n}{4}}.$$

The best known bounds $3/8 = 0.375 < \kappa \leq 0.381$ are due to Lovász-Vesztergombi-Wagner-Welzl and Ábrego-Fernández and to Aichholzer et al. [AAK01], resp. The known exact values of $\text{LIN-CR}(G)$ are listed in Table 10.2.1 [BDG01].

TABLE 10.2.1

n	$\text{LIN-CR}(K_n)$
4	0
5	1
6	3
7	9
8	19
9	36
10	62
11	102
12	153

4. Let $G = G(n, p)$ be a *random* graph with n vertices, whose edges are chosen independently with probability $p = p(n)$. Let $e = p \cdot \binom{n}{2}$. Is it true that the pairwise crossing number, the odd crossing number, and the biplanar crossing number are bounded from below by a constant times e^2 , provided that $e \gg n$?

10.3 GENERALIZATIONS

The concept of geometric graph can be generalized in two natural directions. Instead of straight-line drawings, we can consider curvilinear drawings. If we put them at the focus of our investigations and we wish to emphasize that they are objects of independent interest rather than planar representations of abstract graphs, we call these drawings *topological graphs*. In this sense, the results in the previous section about crossing numbers belong to the theory of topological graphs. Instead of systems of segments induced by a planar point set, we can also consider systems of simplices in the plane or in higher-dimensional spaces. Such a system is called a *geometric hypergraph*.

GLOSSARY

Topological graph: A graph drawn in the plane so that its vertices are distinct points and its edges are simple continuous arcs connecting the corresponding vertices. In a topological graph (a) no edge passes through any vertex other than its endpoints, (b) any two edges have only a finite number of interior points in common, at which they properly cross each other, and (c) no three edges cross at the same point. (Same as *drawing of a graph*.)

Weakly isomorphic topological graphs: Two topological graphs, G and H , such that there is an incidence-preserving one-to-one correspondence between $(V(G), E(G))$ and $(V(H), E(H))$ in which two edges of G intersect if and only if the corresponding edges of H do.

Thrackle: A topological graph in which any two nonadjacent edges cross precisely once and no two adjacent edges cross.

Generalized thrackle: A topological graph in which any two nonadjacent edges cross an odd number of times and any two adjacent edges cross an even number of times (not counting their common endpoint).

d -dimensional geometric r -hypergraph H_r^d : A pair (V, E) , where V is a set of points in general position in d -space, and E is a set of *closed* $(r-1)$ -dimensional simplices induced by some r -tuples of V . The sets V and E are called the *vertex set* and *(hyper)edge set* of H_r^d , respectively. Clearly, a geometric graph is a 2-dimensional geometric 2-hypergraph.

Forbidden geometric hypergraphs: A class \mathcal{F} of geometric hypergraphs not permitted to be contained in the geometric hypergraphs under consideration. Given a class \mathcal{F} of forbidden geometric hypergraphs, $\text{ex}_r^d(\mathcal{F}, n)$ denotes the maximum number of edges that a d -dimensional geometric r -hypergraph H_r^d of n vertices can have without containing a geometric subhypergraph belonging to \mathcal{F} .

Nontrivial intersection: k simplices are said to have a nontrivial intersection if their relative interiors have a point in common.

Crossing of k simplices: A common point of the relative interiors of k simplices, all of whose vertices are *distinct*. The simplices are called *crossing simplices* if such a point exists. A set of simplices may be *pairwise crossing* but not necessarily crossing. If we want to emphasize that they *all* cross, we say that they cross in the *strong sense* or, in brief, that they *strongly cross*.

TOPOLOGICAL GRAPHS

The fairly extensive literature on topological graphs focuses on very few special questions, and there is no standard terminology. Most of the methods developed for the study of geometric graphs break down for topological graphs, unless we make some further structural assumptions. For example, many arguments go through for x -monotone drawings such that any two edges cross at most once. Sometimes it is sufficient to assume the latter condition.

1. An Erdős-Szekeres type theorem: A classical theorem of Erdős and Szekeres states that every complete geometric graph with n vertices has a complete geometric subgraph, weakly isomorphic to a convex complete graph C_m with $m \geq c \log n$ vertices. For complete *topological* graphs with n vertices, any two

of whose edges cross at most once, one can prove the existence of a complete topological subgraph with $m \geq c \log^{1/8} n$ vertices that is weakly isomorphic either to a convex complete graph C_m or to a so-called *twisted* complete graph T_m , as depicted in Figure 10.3.1 [PT01].

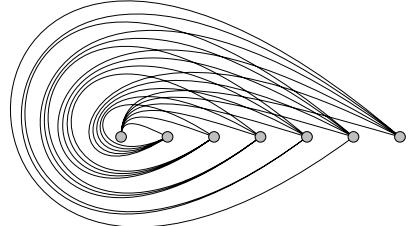


FIGURE 10.3.1

The twisted drawing T_m discovered by Harborth and Mengersen [HM92].

2. Every topological complete graph with n vertices, any two of whose edges cross at most once, has a noncrossing subgraph isomorphic to any given tree T with at most $c \log^{1/6} n$ vertices. In particular, it contains a noncrossing path with at least $c \log^{1/6} n$ vertices [PT01].
3. Number of topological complete graphs: Let $\overline{\Phi}(n)$, $\Phi(n)$, and $\Phi_d(n)$ denote the number of different (i.e., pairwise weakly nonisomorphic) geometric complete graphs, topological complete graphs, and topological complete graphs in which every pair of edges cross at most d times, resp. We have $\log \overline{\Phi}(n) = \Theta(n \log n)$, $\log \Phi(n) = \Theta(n^4)$, $\Omega(n^2) \leq \log \Phi_1(n) \leq O(n^2 \log n)$, and $\Omega(n^2 \log n) \leq \log \Phi_d(n) \leq o(n^4)$ for every $d \geq 2$ (Pach-Tóth).
4. Reducing the number of crossings [PT02, SŠ01]: Given an abstract graph $G = (V, E)$ and a set of pairs of edges $P \subseteq \binom{E}{2}$, we say that a topological graph K is a **weak realization** of G if no pair of edges not belonging to P cross each other. If G has a weak realization, then it also has a weak realization in which every edge crosses at most $2^{|E|}$ other edges. There is an almost matching lower bound for this quantity [KM91].
5. Every cycle of length different from 4 can be drawn as a thrackle [Woo71]. A bipartite graph can be drawn in the plane as a generalized thrackle if and only if it is planar [LPS97]. Every generalized thrackle with $n > 2$ vertices has at most $2n - 2$ edges, and this bound is sharp [CN00].

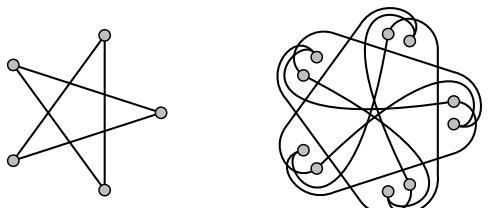


FIGURE 10.3.2

Cycles C_5 and C_{10} drawn as thrackles.

GEOMETRIC HYPERGRAPHS

If we want to generalize the results in the first two sections to higher dimensional geometric hypergraphs, we face some unexpected difficulties. Even if we restrict our attention to systems of triangles induced by 3-dimensional point sets in general

position, it is not completely clear how a “crossing” should be defined. If two segments cross, they do not share an endpoint. Should this remain true for triangles? In this subsection, we describe some scattered results in this direction, but it will require further research to identify the key notions and problems.

- Let \mathcal{D}_k^r denote the class of all geometric r -hypergraphs consisting of k pairwise disjoint edges (closed $(r-1)$ -dimensional simplices). Let \mathcal{I}_k^r (respectively, \mathcal{SI}_k^r) denote the class of all geometric r -hypergraphs consisting of k simplices, any two of which have a nontrivial intersection (respectively, all of which are strongly intersecting). Similarly, let \mathcal{C}_k^r (respectively, \mathcal{SC}_k^r) denote the class of all geometric r -hypergraphs consisting of k pairwise crossing (respectively, strongly crossing) edges. In Table 10.3.1, we summarize the known estimates on $\text{ex}_r^d(\mathcal{F}, n)$, the maximum number of hyperedges (or, simply, edges) that a d -dimensional geometric r -hypergraph of n vertices can have without containing any forbidden subconfiguration belonging to \mathcal{F} . We assume $d \geq 3$. In the first line of the table, the lower bound is conjectured to be tight. The upper bounds in the second line are tight for $d = 2, 3$.

TABLE 10.3.1 Estimates on $\text{ex}_r^d(\mathcal{F}, n)$, the maximum number of edges of a d -dimensional geometric r -hypergraph of n vertices containing no forbidden subconfigurations belonging to \mathcal{F} .

r	\mathcal{F}	LOWER BOUND	UPPER BOUND	SOURCE
d	\mathcal{D}_k^d	$\Omega(n^{d-1})$	$n^{d-(1/k)^{d-1}}$	[AA89]
d	\mathcal{I}_k^d ($k = 2, 3$)	?	$O(n^{d-1})$	[DP98]
d	\mathcal{I}_k^d ($k > 3$)	?	$O(n^{d-1} \log n)$	[Val98]
d	\mathcal{C}_2^d	$\Omega(n^{d-1})$	$O(n^{d-1})$	[DP98]
d	\mathcal{C}_k^d ($k > 2$)	?	$O(n^{d-(1/d)^{k-2}})$	[DP98]
$d+1$	\mathcal{I}_k^{d+1}	$\Omega(n^{\lceil d/2 \rceil})$	$O(n^{\lceil d/2 \rceil})$	[BF87, DP98]
$d+1$	\mathcal{SI}_k^{d+1}	$\Omega(n^{\lceil d/2 \rceil})$	$O(n^{\lceil d/2 \rceil})$	[BF87, DP98]
$d+1$	\mathcal{C}_2^{d+1}	$\Omega(n^d)$	$O(n^d)$	[DP98]

- Akiyama-Alon theorem [AA89]: Let $V = V_1 \cup \dots \cup V_d$ ($|V_1| = \dots = |V_d| = n$) be a dn -element set in general position in d -space, and let E consist of all $(d-1)$ -dimensional simplices having exactly one vertex in each V_i . Then E contains n disjoint simplices. This result can be applied to deduce the upper bound in the first line of Table 10.3.1.
- Assume that, for suitable constants c_1 and $0 \leq \delta \leq 1$, we have $\text{ex}_r^d(\mathcal{SC}_k^r, n) < c_1 \binom{n}{r}/n^\delta$ and $e \geq (c_1 + 1)\binom{n}{r}/n^\delta$. Then there exists $c_2 > 0$ such that the minimum number of strongly crossing k -tuples of edges in a d -dimensional r -hypergraph with n vertices and e edges is at least

$$c_2 \binom{n}{kr} e^\gamma / \binom{n}{r}^\gamma,$$

where $\gamma = 1 + (k-1)r/\delta$. This result can be used to deduce the upper bound in line 5 of Table 10.3.1.

4. A Ramsey-type result [DP98]: Let us 2-color all $(d-1)$ -dimensional simplices induced by $(d+1)n-1$ points in general position in \mathbb{R}^d . Then one can always find n disjoint simplices of the same color. This result cannot be improved.
5. Convex geometric hypergraphs in the plane [Bra04]: If we choose triangles from points in convex position in the plane, then the concept of isomorphism is much clearer than in the higher-dimensional cases. Thus two triangles without a common vertex can occur in three mutual positions, and we have $\text{ex}(n, \text{△}) = \Theta(n^3)$, $\text{ex}(n, \text{△△}) = \Theta(n^2)$, $\text{ex}(n, \text{△△△}) = \Theta(n^2)$. Similarly, two triangles with one common vertex can occur again in three positions and we have $\text{ex}(n, \text{△△}) = \Theta(n^3)$, $\text{ex}(n, \text{△△△}) = \Theta(n^2)$, $\text{ex}(n, \text{△△△△}) = \Theta(n^2)$, which is surprising, since the underlying hypergraph has a linear Turán function. Finally, two triangles with two common vertices have two possible positions, and we have $\text{ex}(n, \text{△△△}) = \Theta(n^3)$, $\text{ex}(n, \text{△△△△}) = \Theta(n^2)$. Larger sets of forbidden convex geometric subhypergraphs occur as the combinatorial core of several combinatorial geometry problems.

OPEN PROBLEMS

1. (Ringel, Harborth) For any k , determine or estimate the smallest integer $n = n(k)$ for which there is a complete topological graph with n vertices, every pair of whose edges intersect at most once (including possibly at their common endpoints), and every edge of which crosses at least k others. It is known that $n(1) = 8$, $7 \leq n(2) \leq 11$, $7 \leq n(3) \leq 14$, $7 \leq n(4) \leq 16$, and $n(k) \leq 4k/3 + O(\sqrt{k})$ [HT94]. Does $n(k) = o(k)$ hold?
2. (Harborth) Is it true that each vertex of a complete topological graph with n vertices, every pair of whose edges cross at most once (including possibly at their common endpoints), is a vertex of at least *two* empty triangles? (A triangle bounded by all edges connecting three vertices is said to be *empty*, if there is no point in its interior or exterior.) It is known [Har98] that every complete topological graph with the above property has at least two empty triangles.
3. (Conway) Is it true that the number of edges of a thrackle can never exceed its number of vertices? It is known that every thrackle with n vertices has at most $1.5(n - 1)$ edges [CN00].
4. (Kalai) What is the maximum number $\mu(n)$ of hyperedges that a 3-dimensional geometric 3-hypergraph of n vertices can have, if any pair of its hyperedges either are disjoint or share at most one vertex? Is it true that $\mu(n) = o(n^2)$? Károlyi and Solymosi [KS02] showed that $\mu(n) = \Omega(n^{3/2})$.

10.4 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[PA95]: Monograph devoted to combinatorial geometry. [Chapter 14](#) is dedicated to geometric graphs.

[Pac99] The most extensive survey on geometric graph theory.

[Pac91, DP98]: The first surveys of results in geometric graph theory and geometric hypergraph theory, respectively.

[PT00a, Pac00, Szé, SSSV97]: Surveys on open problems and on crossing numbers.

[BETT99]: Monograph on graph drawing algorithms.

[BMP04]: Survey of representative results and open problems in discrete geometry, originally started by the Moser brothers.

[Grü72]: Monograph containing many results and conjectures on configurations and arrangements of points and arcs.

RELATED CHAPTERS

[Chapter 1: Finite point configurations](#)

[Chapter 5: Pseudoline arrangements](#)

[Chapter 11: Euclidean Ramsey theory](#)

[Chapter 24: Arrangements](#)

[Chapter 52: Graph drawing](#)

REFERENCES

- [AA89] J. Akiyama and N. Alon. Disjoint simplices and geometric hypergraphs. In G.S. Bloom, R. Graham, and J. Malkevitch, editors, *Combinatorial Mathematics*, volume 555 of *Ann. New York Acad. Sci.*, pages 1–3. New York Acad. Sci., 1989.
- [AAK01] O. Aichholzer, F. Aurenhammer, and H. Krasser. Enumerating order types for small point sets with applications. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 2001, pages 11–18.
- [AAP⁺97] P.K. Agarwal, B. Aronov, J. Pach, R. Pollack, and M. Sharir. Quasi-planar graphs have a linear number of edges. *Combinatorica*, 17:1–9, 1997.
- [ACNS82] M. Ajtai, V. Chvátal, M. Newborn, and E. Szemerédi. Crossing-free subgraphs. *Ann. Discrete Math.*, 12:9–12, 1982.
- [AEG⁺94] B. Aronov, P. Erdős, W. Goddard, D.J. Kleitman, M. Klugerman, J. Pach, and L.J. Schulman. Crossing families. *Combinatorica*, 14:127–134, 1994.
- [BD93] D. Bienstock and N. Dean. Bounds for rectilinear crossing numbers. *J. Graph Theory*, 17:333–348, 1993.
- [BDG01] A. Brodsky, S. Durocher, and E. Gethner. The rectilinear crossing number of K_{10} is 62. *Electron. J. Combin.*, 8:#R23, 2001.
- [BV] A. Bialostocki and W. Voxman. Either a graph or its complement is connected: A continuing saga. Manuscript.
- [Bec83] J. Beck. On size Ramsey number of paths, trees, and circuits I. *J. Graph Theory*, 7:115–129, 1983.
- [BETT99] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, Upper Saddle River, 1999.

- [BF87] I. Bárány and Z. Füredi. Empty simplices in Euclidean space. *Canad. Math. Bull.*, 30:436–445, 1987.
- [BH96] A. Bialostocki and H. Harborth. Ramsey colorings for diagonals of convex polygons. *Abh. Braunschweig. Wiss. Ges.*, 47:159–163, 1996.
- [Bie91] D. Bienstock. Some provably hard crossing number problems. *Discrete Comput. Geom.*, 6:443–459, 1991.
- [BKV03] P. Brass, G. Károlyi, and P. Valtr. A Turán-type extremal theory of convex geometric graphs. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 275–300. Springer-Verlag, Berlin, 2003.
- [BMP04] P. Brass, W.O.J. Moser, and J. Pach. *Research Problems in Discrete Geometry*. To appear, 2004.
- [Bra04] P. Brass. Turán-type extremal problems for convex geometric hypergraphs. In J. Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemp. Math.* Amer. Math. Soc., Providence, 2004.
- [BS93] G.R. Brightwell and E.R. Scheinerman. Representations of planar graphs. *SIAM J. Discrete Math.*, 6:214–229, 1993.
- [Cho34] Ch. Chojnacki. Über wesentlich unplättbare Kurven im drei-dimensionalen Raume. *Fund. Math.*, 23:135–142, 1934.
- [CN00] G. Cairns and Y. Nikalayevsky. Bounds for generalized thrackles. *Discrete Comput. Geom.*, 23:191–206, 2000.
- [CP92] V. Capoyleas and J. Pach. A Turán-type theorem on chords of a convex polygon. *J. Combin. Theory Ser. B*, 56:9–15, 1992.
- [dFdMP94] H. de Fraysseix, P. de Mendez, and J. Pach. Representation of planar graphs by segments. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, volume 63 of *Colloq. Math. Soc. János Bolyai*, pages 109–117. North-Holland, Amsterdam, 1994.
- [dFPP90] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.
- [DP98] T.K. Dey and J. Pach. Extremal problems for geometric hypergraphs. *Discrete Comput. Geom.*, 19:473–484, 1998.
- [DV02] H. Djidjev and I. Vrt'o. An improved lower bound on crossing numbers. In *Graph Drawing*, volume 2265 of *Lecture Notes in Comput. Sci.*, pages 96–101. Springer-Verlag, Berlin, 2002.
- [EFRS78] P. Erdős, R.J. Faudree, C.C. Rousseau, and R.H. Schelp. The size Ramsey number. *Period. Math. Hungar.*, 9:145–161, 1978.
- [EG73] P. Erdős and R.K. Guy. Crossing number problems. *Amer. Math. Monthly*, 80:52–58, 1973.
- [EGS03] G. Even, S. Guha, and B. Schieber. Improved approximations of crossings in graph drawings and VLSI layout areas. *SIAM J. Comput.*, 32:231–252, 2003.
- [Fár48] I. Fáry. On straight line representation of planar graphs. *Acta Univ. Szeged. Sect. Sci. Math.*, 11:229–233, 1948.
- [FdF00] L. Faria and C.M.H. de Figueiredo. On Eggleton and Guy conjectured upper bounds for the crossing number of the n -cube. *Math. Slovaca*, 50:271–287, 2000.
- [GJ83] M.R. Garey and D.S. Johnson. Crossing number is NP-complete. *SIAM J. Alg. Discrete Meth.*, 4:312–316, 1983.

- [GMPP91] P. Gritzmann, B. Mohar, J. Pach, and R. Pollack. Embedding a planar triangulation with vertices at specified points (solution to problem E3341). *Amer. Math. Monthly*, 98:165–166, 1991.
- [GRS90] R.L. Graham, B.L. Rothschild, and J.H. Spencer. *Ramsey Theory*, 2nd ed. Wiley, New York, 1990.
- [Grü72] B. Grünbaum. *Arrangements and Spreads*. Volume 10 of *CBMS Regional Conf. Ser. in Math.* Amer. Math. Soc., Providence, 1972.
- [GS] L. Y. Glebsky and G. Salazar. The crossing number of $C_m \times C_n$ is as conjectured for $n \geq m(m+1)$. *J. Graph Theory*, to appear.
- [Guy69] R.K. Guy. The decline and fall of Zarankiewicz's theorem. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 63–69. Academic Press, New York, 1969.
- [Har98] H. Harborth. Empty triangles in drawings of the complete graph. *Discrete Math.*, 191:109–111, 1998.
- [HKS73] F. Harary, P.C. Kainen, and A.J. Schwenk. Toroidal graphs with arbitrarily high crossing numbers. *Nanta Math.*, 6:58–67, 1973.
- [HM92] H. Harborth and I. Mengersen. Drawings of the complete graph with maximum number of crossings. In *Proc. 23rd Southeast. Internat. Conf. Combin. Graph Theory Comput., Congr. Numer.*, 88:225–228, 1992.
- [HP34] H. Hopf and E. Pannwitz. Aufg. Nr. 167. *Jahresb. Deutsch. Math.-Ver.*, 43:114, 1934.
- [HT94] H. Harborth and C. Thürmann. Minimum number of edges with at most s crossings in drawings of the complete graph. In *Proc. 25th Southeast. Internat. Conf. Combin. Graph Theory Comput., Congr. Numer.*, 102:83–90, 1994.
- [IPTT94] Y. Ikebe, M. Perles, A. Tamura, and S. Tokunaga. The rooted tree embedding problem into points in the plane. *Discrete Comput. Geom.*, 11:51–63, 1994.
- [Kin92] N. Kinnersley. The vertex separation number of a graph equals its path-width. *Inform. Process. Lett.*, 42:345–350, 1992.
- [KK00] A. Kaneko and M. Kano. Straight line embeddings of rooted star forests in the plane. *Discrete Appl. Math.*, 101:167–175, 2000.
- [KK03] A. Kaneko and M. Kano. Discrete geometry on red and blue points in the plane—a survey. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 551–570. Springer-Verlag, Berlin, 2003.
- [Kle70] D.J. Kleitman. The crossing number of $k_{5,n}$. *J. Combin. Theory*, 9:315–323, 1970.
- [KM91] J. Kratochvíl and J. Matoušek. String graphs requiring exponential representations. *J. Combin. Theory Ser. B*, 53:1–4, 1991.
- [Koe36] P. Koebe. Kontaktprobleme der konformen Abbildung. *Ber. Verh. Sächs. Akad. Wiss. Leipzig Math.-Phys. Klasse*, 88:141–164, 1936.
- [KP96] Y. Kupitz and M.A. Perles. Extremal theory for convex matchings in convex geometric graphs. *Discrete Comput. Geom.*, 15:195–220, 1996.
- [KPT97] G. Károlyi, J. Pach, and G. Tóth. Ramsey-type results for geometric graphs I. *Discrete Comput. Geom.*, 18:247–255, 1997.
- [KPTV98] G. Károlyi, J. Pach, G. Tóth, and P. Valtr. Ramsey-type results for geometric graphs II. *Discrete Comput. Geom.*, 20:375–388, 1998.
- [KS02] G. Károlyi and J. Solymosi. Almost disjoint triangles in 3-space. *Discrete Comput. Geom.*, 28:577–583, 2002.

- [Kup84] Y. Kupitz. On pairs of disjoint segments in convex position in the plane. *Ann. Discrete Math.*, 20:203–208, 1984.
- [Lei83] T. Leighton. *Complexity Issues in VLSI, Foundations of Computing Series*. MIT Press, Cambridge, 1983.
- [LMPT94] D. Larman, J. Matoušek, J. Pach, and J. Töröcsik. A Ramsey-type result for planar convex sets. *Bull. London Math. Soc.*, 26:132–136, 1994.
- [LPS97] L. Lovász, J. Pach, and M. Szegedy. On Conway’s thrackle conjecture. *Discrete Comput. Geom.*, 18:369–376, 1997.
- [PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. Wiley, New York, 1995.
- [Pac91] J. Pach. Notes on geometric graph theory. In J. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 273–285. Amer. Math. Soc., Providence, 1991.
- [Pac99] J. Pach. Geometric graph theory. In J.D. Lamb and D.A. Preece, editors, *Surveys in Combinatorics, 1999*, volume 267 of *London Math. Soc. Lecture Note Ser.*, pages 167–200. Cambridge University Press, 1999.
- [Pac00] J. Pach. Crossing numbers. In J. Akiyama, M. Kano, and M. Urabe, editors, *Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Comput. Sci.*, pages 267–273. Springer-Verlag, Berlin, 2000.
- [PPST] J. Pach, R. Pinchasi, M. Sharir, and G. Tóth. Topological graphs with no large grids. *Graphs Combin.*, to appear.
- [PPTT02] J. Pach, R. Pinchasi, G. Tardos, and G. Tóth. Geometric graphs with no self-intersecting path of length three. In M.T. Goodrich and S.G. Kobourov, editors, *Graph Drawing*, volume 2528 of *Lecture Notes in Comput. Sci.*, pages 295–311. Springer-Verlag, Berlin, 2002.
- [PR03] R. Pinchasi and R. Radoičić. Topological graphs with no self-intersecting cycle of length 4. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 98–103, 2003.
- [PRTT04] J. Pach, R. Radoičić, G. Tardos, and G. Tóth. Graphs drawn with at most 3 crossings per edge. In J. Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemp. Math.* Amer. Math. Soc., Providence, 2004.
- [PSS96] J. Pach, F. Shahrokhi, and M. Szegedy. Applications of crossing number. *Algorithmica*, 16:111–117, 1996.
- [PST00] J. Pach, J. Spencer, and G. Tóth. New bounds on crossing numbers. *Discrete Comput. Geom.*, 24:623–644, 2000.
- [PT94] J. Pach and J. Töröcsik. Some geometric applications of Dilworth’s theorem. *Discrete Comput. Geom.*, 12:1–7, 1994.
- [PT97] J. Pach and G. Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17:427–439, 1997.
- [PT00a] J. Pach and G. Tóth. Thirteen problems on crossing numbers. *Geombinatorics*, 9:194–207, 2000.
- [PT00b] J. Pach and G. Tóth. Which crossing number is it anyway? *J. Combin. Theory Ser. B*, 80:225–246, 2000.
- [PT01] J. Pach and G. Tóth. Unavoidable configurations in complete topological graphs. In J. Marks, editor, *Graph Drawing*, volume 1984 of *Lecture Notes in Comput. Sci.*, pages 328–337. Springer-Verlag, Berlin, 2001.
- [PT02] J. Pach and G. Tóth. Recognizing string graphs is decidable. *Discrete Comput. Geom.*, 28:593–606, 2002.

- [PT03] J. Pach and G. Tóth. Monotone drawings of planar graphs. In P. Bose and P. Morin, editors, *Algorithms and Computation*, volume 2518 of *Lecture Notes in Comput. Sci.*, pages 647–653. Springer-Verlag, Berlin, 2003.
- [Ram30] F. Ramsey. On a problem of formal logic. *Proc. London Math. Soc.*, 30:264–286, 1930.
- [Sch90] W. Schnyder. Embedding planar graphs on the grid. In *Proc. 1st Annu. ACM-SIAM Sympos. Discrete Algor.*, pages 138–148, 1990.
- [SŠ01] M. Schaefer and D. Štefankovič. Decidability of string graphs. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 241–246, 2001.
- [SSŠ02] M. Schaefer, E. Sedgwick, and D. Štefankovič. Recognizing string graphs in NP. In *Proc. 34th Annu ACM Sympos. Theory Comput.*, pages 1–6, 2002.
- [SSSV97] F. Shahrokhi, O. Sýkora, L.A. Székely, and I. Vrt'o. Crossing numbers: bounds and applications. In I. Bárány and K. Böröczky, editors, *Intuitive Geometry*, volume 6 of *Bolyai Soc. Math. Stud.*, pages 179–206. J. Bolyai Math. Soc., Budapest, 1997.
- [SSV] O. Sýkora, L.A. Székely, and I. Vrt'o. Crossing numbers and biplanar crossing numbers: using the probabilistic method. To appear.
- [ST02] J. Spencer and G. Tóth. Crossing numbers of random graphs. *Random Structures Algorithms*, 21:347–358, 2002.
- [Ste22] E. Steinitz. Polyeder und Raumteilungen, part 3AB12. In *Enzykl. Math. Wiss. 3 (Geometrie)*, pages 1–139. 1922.
- [SV93] O. Sýkora and I. Vrt'o. On the crossing number of the hypercube and the cube connected cycles. *BIT*, 33:232–237, 1993.
- [SV94] O. Sýkora and I. Vrt'o. On VLSI layouts of the star graph and related networks. *Integration, the VLSI journal*, 17:83–93, 1994.
- [Szé] L.A. Székely. A successful concept for measuring non-planarity of graphs: the crossing number. *Discrete Math.*, to appear.
- [Szé97] L.A. Székely. Crossing numbers and hard Erdős problems in discrete geometry. *Combin. Probab. Comput.*, 6:353–358, 1997.
- [Thu78] W.P. Thurston. *The Geometry and Topology of 3-manifolds*. Lecture notes, Princeton Univ., 1978.
- [Tót00] G. Tóth. Note on geometric graphs. *J. Combin. Theory Ser. A*, 89:126–132, 2000.
- [Tur54] P. Turán. On the theory of graphs. *Colloq. Math.*, 3:19–30, 1954.
- [Tur77] P. Turán. A note of welcome. *J. Graph Theory*, 1:7–9, 1977.
- [Tut60] W.T. Tutte. Convex representations of graphs. *Proc. London Math. Soc.*, 10:304–320, 1960.
- [Val98] P. Valtr. On geometric graphs with no k pairwise parallel edges. *Discrete Comput. Geom.*, 19:461–469, 1998.
- [Woo71] D.R. Woodall. Thrackles and deadlock. In D.J.A. Welsh, editor, *Combinatorial Mathematics and Its Applications*, pages 335–348. Academic Press, London, 1971.
- [Woo93] D.R. Woodall. Cyclic-order graphs and Zarankiewicz's crossing-number conjecture. *J. Graph Theory*, 17:657–671, 1993.

11 EUCLIDEAN RAMSEY THEORY

R.L. Graham

INTRODUCTION

Ramsey theory typically deals with problems of the following type. We are given a set S , a family \mathcal{F} of subsets of S , and a positive integer r . We would like to decide whether or not for every partition of $S = C_1 \cup \dots \cup C_r$ into r subsets, it is always true that some C_i contains some $F \in \mathcal{F}$. If so, we abbreviate this by writing $S \xrightarrow{r} \mathcal{F}$ (and we say S is r -Ramsey). If not, we write $S \not\xrightarrow{r} \mathcal{F}$. (For a comprehensive treatment of Ramsey theory, see [GRS90].)

In Euclidean Ramsey theory, S is usually taken to be the set of points in some Euclidean space \mathbb{E}^N , and the sets in \mathcal{F} are determined by various geometric considerations. The case most studied is the one in which $\mathcal{F} = \text{Cong}(X)$ consists of all *congruent* copies of a fixed finite configuration $X \subset S = \mathbb{E}^N$. In other words, $\text{Cong}(X) = \{gX \mid g \in SO(N)\}$, where $SO(N)$ denotes the special orthogonal group acting on \mathbb{E}^N .

Further, we say that X is *Ramsey* if, for all r , $\mathbb{E}^N \xrightarrow{r} \text{Cong}(X)$ holds provided N is sufficiently large (depending on X and r). This we indicate by writing $\mathbb{E}^N \longrightarrow X$.

Another important case we will discuss (in Section 11.4) is that in which $\mathcal{F} = \text{Hom}(X)$ consists of all *homothetic* copies $aX + \bar{t}$ of X , where a is a positive real and $\bar{t} \in \mathbb{E}^N$. Thus, in this case \mathcal{F} is just the set of all images of X under the group of positive homotheties acting on \mathbb{E}^N .

It is easy to see that any Ramsey (or r -Ramsey) set must be finite. A standard compactness argument shows that if $\mathbb{E}^N \xrightarrow{r} X$ then there is always a *finite* set $Y \subseteq \mathbb{E}^N$ such that $Y \xrightarrow{r} X$. Also, if X is Ramsey (or r -Ramsey) then so is any homothetic copy $aX + \bar{t}$ of X .

GLOSSARY

$\mathbb{E}^N \xrightarrow{r} \text{Cong}(X)$: For any partition $\mathbb{E}^N = C_1 \cup \dots \cup C_r$, some C_i contains a set congruent to X . We say that X is **r -Ramsey**. When $\text{Cong}(X)$ is understood we will usually write $\mathbb{E}^N \xrightarrow{r} X$.

$\mathbb{E}^N \longrightarrow X$: For every r , $\mathbb{E}^N \xrightarrow{r} \text{Cong}(X)$ holds, provided N is sufficiently large. We say in this case that X is **Ramsey**.

11.1 r -RAMSEY SETS

In this section we focus on low-dimensional r -Ramsey results. We begin by stating three conjectures.

CONJECTURE 11.1.1

For any nonequilateral triangle T (i.e., the set of 3 vertices of T),

$$\mathbb{E}^2 \xrightarrow{2} T.$$

CONJECTURE 11.1.2 (stronger)

For any partition $\mathbb{E}^2 = C_1 \cup C_2$, every triangle occurs (up to congruence) in C_1 , or else the same holds for C_2 , with the possible exception of a single equilateral triangle.

The partition $\mathbb{E}^2 = C_1 \cup C_2$ with

$$\begin{aligned} C_1 &= \{(x, y) \mid -\infty < x < \infty, 2m \leq y < 2m + 1, m = 0, \pm 1, \pm 2, \dots\} \\ C_2 &= \mathbb{E}^2 \setminus C_1 \end{aligned}$$

into alternating half-open strips of width 1 prevents the equilateral triangle of side $\sqrt{3}$ from occurring in a single C_i . In fact, it is conjectured that except for some freedom in assigning the boundary points (x, m) , m an integer, this is the only way of avoiding *any* triangle.

CONJECTURE 11.1.3

For any triangle T ,

$$\mathbb{E}^2 \xrightarrow[3]{\quad} T.$$

In the positive direction, we have [EGM⁺75b]:

THEOREM 11.1.4

(a) $\mathbb{E}^2 \xrightarrow{2} T$ if T is a triangle satisfying:

- (i) T has a ratio between two sides equal to $2 \sin \theta / 2$ with $\theta = 30^\circ, 72^\circ, 90^\circ$, or 120°
- (ii) T has a $30^\circ, 90^\circ$, or 150° angle [Sha76]
- (iii) T has angles $(\alpha, 2\alpha, 180^\circ - 3\alpha)$ with $0 < \alpha < 60^\circ$
- (iv) T has angles $(180^\circ - \alpha, 180^\circ - 2\alpha, 3\alpha - 180^\circ)$ with $60^\circ < \alpha < 90^\circ$
- (v) T is the degenerate triangle $(a, 2a, 3a)$
- (vi) T has sides (a, b, c) satisfying

$$a^6 - 2a^4b^2 + a^2b^4 - 3a^2b^2c^2 + b^2c^2 = 0$$

or

$$a^4c^2 + b^4a^2 + c^4b^2 - 5a^2b^2c^2 = 0$$

- (vii) T has sides (a, b, c) satisfying

$$c^2 = a^2 + 2b^2 \text{ with } a < 2b \quad [\text{Sha76}]$$

- (viii) T has sides (a, b, c) satisfying

$$a^2 + c^2 = 4b^2 \text{ with } 3b^2 < 2a^2 < 5b^2 \quad [\text{Sha76}]$$

- (ix) T has sides equal in length to the sides and circumradius of an isosceles triangle;
- (b) $\mathbb{E}^3 \xrightarrow{2} T$ for any nondegenerate triangle T
- (c) $\mathbb{E}^3 \xrightarrow{3} T$ for any nondegenerate right triangle T [BT96]
- (d) $\mathbb{E}^3 \xrightarrow{12} T$, a triangle with angles $(30^\circ, 60^\circ, 90^\circ)$ [Bón93]
- (e) $\mathbb{E}^2 \xrightarrow{2} Q^2$ (4 points forming a square)
- (f) $\mathbb{E}^4 \xrightarrow{2} Q^2$ [Can96a]
- (g) $\mathbb{E}^5 \xrightarrow{2} R^2$, any rectangle [Tót96]
- (h) $\mathbb{E}^n \xrightarrow{4} \circ \xrightarrow{1} \circ \xrightarrow{1} \circ$ for any n (a degenerate $(1, 1, 2)$ triangle)
- (i) $\mathbb{E}^n \xrightarrow{16} \circ \xrightarrow{a} \circ \xrightarrow{b} \circ$ for any n (a degenerate $(a, b, a+b)$ triangle).

It is not known whether the 4 in (h) or the 16 in (i) can be replaced by smaller values. Other results of this type can be found in [EGM⁺73], [EGM⁺75a], [EGM⁺75b], [Sha76], [CFG91].

The 2-point set X_2 consisting of two points a unit distance apart is the simplest set about which such questions can be asked, and has a particularly interesting history (see [Soi91] for details). It is clear that

$$\mathbb{E}^1 \xrightarrow{2} X_2 \quad \text{and} \quad \mathbb{E}^2 \xrightarrow{2} X_2.$$

To see that $\mathbb{E}^2 \xrightarrow{3} X_2$, consider the 7-point Moser graph shown in Figure 11.1.1. All edges have length 1. On the other hand, $\mathbb{E}^2 \not\xrightarrow{7} X_2$, which can be seen by an appropriate periodic 7-coloring (= partition into 7 parts) of a tiling of \mathbb{E}^2 by regular hexagons of diameter 0.9 (see Figure 1.3.1).

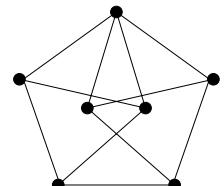


FIGURE 11.1.1
The Moser graph.

Definition: The *chromatic number* of \mathbb{E}^n , denoted by $\chi(\mathbb{E}^n)$, is the least m such that $\mathbb{E}^n \xrightarrow{m} X_2$.

By the above remarks,

$$4 \leq \chi(\mathbb{E}^2) \leq 7.$$

These bounds have remained unchanged for over 50 years.

Some evidence that $\chi(\mathbb{E}^2) \geq 5$ (in the author's opinion) is given by the following result of O'Donnell:

THEOREM 11.1.5 [O'D00a], [O'D00b]

For any $g > 0$, there is 4-chromatic unit distance graph in \mathbb{E}^2 with girth greater than g .

Note that the Moser graph has girth 3.

PROBLEM 11.1.6

Determine the exact value of $\chi(\mathbb{E}^2)$.

The best bounds currently known for \mathbb{E}^n are:

$$(6/5 + o(1))^n < \chi(\mathbb{E}^n) < (3 + o(1))^n$$

(see [FW81], [CFG91]).

A "near miss" for showing $\chi(\mathbb{E}^2) < 7$ was found by Soifer [Soi92]. He shows that there exists a partition $\mathbb{E}^2 = C_1 \cup \dots \cup C_7$ where C_i contains no pair of points at distance 1 for $1 \leq i \leq 6$, while C_7 has no pair at distance $1/\sqrt{5}$.

The best bounds known for $\chi(\mathbb{E}^3)$ are:

$$6 \leq \chi(\mathbb{E}^3) \leq 15.$$

The lower bound is due to Nechushtan [Nech02] and the upper bound is due to R. Radoicic and G. Tóth [RT03] (improving earlier results of Székely/Wormald [SW89] and Bóna/Tóth [BT96]).

See [Section 1.3](#) of this Handbook for more details.

11.2 RAMSEY SETS

Recall that X is Ramsey (written $\mathbb{E}^N \rightarrow X$) if, for all r , if $\mathbb{E}^N = C_1 \cup \dots \cup C_r$ then some C_i must contain a congruent copy of X , provided only that $N \geq N_0(X, r)$.

GLOSSARY

Spherical: X is spherical if it lies on the surface of some sphere.

Rectangular: X is rectangular if it is a subset of the vertices of a rectangular parallelepiped.

Simplex: X is a simplex if it spans $\mathbb{E}^{|X|-1}$.

THEOREM 11.2.1 [EGM⁺73]

If X and Y are Ramsey then so is $X \times Y$.

Thus, since any 2-point set is Ramsey (for any r , consider the unit simplex S_{2r+1} in \mathbb{E}^{2r} scaled appropriately), then so is any rectangular parallelepiped. This implies:

THEOREM 11.2.2

Any rectangular set is Ramsey.

Frankl and Rödl strengthen this significantly in the following way.

Definition: A set $A \subset \mathbb{E}^n$ is called ***super-Ramsey*** if there exist positive constants c and ϵ and subsets $X = X(N) \subset \mathbb{E}^N$ for every $N \geq N_0(X)$ such that:

- (i) $|X| < c^n$;
- (ii) $|Y| < |X|/(1 + \epsilon)^n$ holds for all subsets $Y \subset X$ containing no congruent copy of A .

THEOREM 11.2.3 [FR90]

- (i) All two-element sets are super-Ramsey.
- (ii) If A and B are super-Ramsey then so is $A \times B$.

COROLLARY 11.2.4

If X is rectangular then X is super-Ramsey.

In the other direction we have:

THEOREM 11.2.5

Any Ramsey set is spherical.

The simplest nonspherical set is the degenerate $(1, 1, 2)$ triangle.

Concerning simplices, we have the result of Frankl and Rödl:

THEOREM 11.2.6 [FR90]

Every simplex is Ramsey.

In fact, they show that for any simplex X , there is a constant $c = c(X)$ such that for all r ,

$$\mathbb{E}^{c \log r} \xrightarrow{r} X,$$

which follows from their result:

THEOREM 11.2.7

Every simplex is super-Ramsey.

It was an open problem for more than 20 years as to whether the set of vertices of a regular pentagon was Ramsey. This was finally settled by Kříž [Kří91] who proved the following two fundamental results:

THEOREM 11.2.8 [Kří91]

Suppose $X \subseteq \mathbb{E}^N$ has a transitive solvable group of isometries. Then X is Ramsey.

COROLLARY 11.2.9

Any set of vertices of a regular polygon is Ramsey.

THEOREM 11.2.10 [Kří91]

Suppose $X \subseteq \mathbb{E}^N$ has a transitive group of isometries that has a solvable subgroup with at most two orbits. Then X is Ramsey.

COROLLARY 11.2.11

The vertex sets of the Platonic solids are Ramsey.

CONJECTURE 11.2.12

Any 4-point subset of a circle is Ramsey.

Kříž [Kři92] has shown this holds if a pair of opposite sides of the 4-point set are parallel (i.e., form a trapezoid).

Certainly, the outstanding open problem in Euclidean Ramsey theory is to determine the Ramsey sets. The author (bravely?) makes the following:

CONJECTURE 11.2.13 (\$1000)

Any spherical set is Ramsey.

If true then this would imply that the Ramsey sets are exactly the spherical sets.

11.3 SPHERE-RAMSEY SETS

Since spherical sets play a special role in Euclidean Ramsey theory, it is natural that the following concept arises.

GLOSSARY

$S^N(\rho)$: A sphere in \mathbb{E}^N with radius ρ .

Sphere-Ramsey: X is sphere-Ramsey if, for all r , there exist $N = N(X, r)$ and $\rho = \rho(X, r)$ such that

$$S^N(\rho) \xrightarrow{r} X.$$

In this case we write $S^N(\rho) \longrightarrow X$.

For a spherical set X , let $\rho(X)$ denote its circumradius, i.e., the radius of the smallest sphere containing X as a subset.

Remark. If X and Y are sphere-Ramsey then so is $X \times Y$.

THEOREM 11.3.1 [Gra83]

If X is rectangular then X is sphere-Ramsey.

In [Gra83], it was conjectured that in fact if X is rectangular and $\rho(X) = 1$ then $S^N(1 + \epsilon) \longrightarrow X$ should hold. This was proved by Frankl and Rödl [FR90] in a much stronger “super-Ramsey” form.

Concerning simplices, Matoušek and Rödl proved the following spherical analogue of simplices being Ramsey:

THEOREM 11.3.2 [MR95]

For any simplex X with $\rho(X) = 1$, any r , and any $\epsilon > 0$, there exists $N = N(X, r, \epsilon)$ such that

$$S^N(1 + \epsilon) \xrightarrow{r} X.$$

The proof uses an interesting mix of techniques from combinatorics, linear algebra, and Banach space theory.

The following results show that the “blowup factor” of $1 + \epsilon$ is really needed.

THEOREM 11.3.3 [Gra83]

Let $X = \{x_1, \dots, x_m\} \subset \mathbb{E}^N$ such that:

(i) for some nonempty $I \subseteq \{1, 2, \dots, m\}$, there exist nonzero a_i , $i \in I$, with

$$\sum_{i \in I} a_i x_i = 0 \in \mathbb{E}^N$$

(ii) for all nonempty $J \subseteq I$,

$$\sum_{j \in J} a_j \neq 0.$$

Then X is not sphere-Ramsey.

This implies that $X \subset S^N(1)$ is not sphere-Ramsey if the convex hull of X contains the center of $S^N(1)$.

Definition: A simplex $X \subset \mathbb{E}^N$ is called *exceptional* if there is a subset $A \subseteq X$, $|A| \geq 2$, such that the affine hull of A translated to the origin has a nontrivial intersection with the linear span of the points of $X \setminus A$ regarded as vectors.

THEOREM 11.3.4 [MR95]

If X is a simplex with $\rho(X) = 1$ and $S^N(1) \rightarrow X$ then X must be exceptional.

It is not known whether it is true for exceptional X that $S^N(1) \rightarrow X$. The simplest nontrivial case is for the set of three points $\{a, b, c\}$ lying on some great circle of $S^N(1)$ (with center o) so that the line joining a and b is parallel to the line joining o and c .

We close with a fundamental conjecture:

CONJECTURE 11.3.5

If X is Ramsey, then X is sphere-Ramsey.

11.4 EDGE-RAMSEY SETS

In this variant (introduced in [EGM⁺75b]), we color all the line segments $[a, b]$ in \mathbb{E}^n rather than coloring the points. Analogously to our earlier definition, we will say that a configuration E of line segments is *edge-Ramsey* if for any r , there is an $N = N(r)$ such any r -coloring of the line segments in \mathbb{E}^N contains a monochromatic congruent copy of E (up to some Euclidean motion). The main results known for edge-Ramsey configurations are the following:

THEOREM 11.4.1 [EGM⁺75b]

If E is edge-Ramsey then all edges of E must have the same length.

THEOREM 11.4.2 [Gra83]

If E is edge-Ramsey then the endpoints of the edges of E must lie on two spheres.

THEOREM 11.4.3 [Gra83]

If the endpoints of E do not lie on a sphere and the graph formed by E is not bipartite then E is not edge-Ramsey.

It is clear that the edge set of an n -dimensional simplex is edge-Ramsey. Less obvious (but equally true) are the following.

THEOREM 11.4.4 [Can96b]

The edge set of an n -cube is edge-Ramsey.

THEOREM 11.4.5 [Can96b]

The edge set of an n -dimensional cross polytope is edge-Ramsey.

This set, a generalization of the octahedron, has as its edges all $2n(n - 1)$ line segments of the form $[(0, 0, \dots, \pm 1, \dots, 0), (0, 0, \dots, 0, \pm 1, \dots, 0)]$ where the two ± 1 's occur in different positions.

THEOREM 11.4.6 [Can96b]

The edge set of a regular n -gon is not edge-Ramsey if $n = 5$ or $n \geq 7$.

Since regular n -gons are edge-Ramsey for $n = 2, 3$, and 4 , the only undecided value is $n = 6$.

PROBLEM 11.4.7 Is the edge set of a regular hexagon edge-Ramsey?

The situation is not as simple as one might hope since as pointed out by Cantwell [Can96b]:

(i) If AB is a line segment with C as its midpoint, then the set E_1 consisting of the line segments AC and CB is not edge-Ramsey, even though its graph is bipartite and A, B, C lie on two spheres.

(ii) There exist nonspherical sets that are edge-Ramsey.

PROBLEM 11.4.8 Characterize edge-Ramsey configurations.

It is not clear at this point what a reasonable conjecture might be. For more results on these topics, see [Can96b] or [Gra83].

11.5 HOMOTHETIC RAMSEY SETS AND DENSITY THEOREMS

In this section we will survey various results of the type $\mathbb{E}^N \xrightarrow{r} \text{Hom}(X)$, the set of positive homothetic images $aX + \bar{t}$ of a given set X . Thus, we are allowed to dilate and translate X but we cannot rotate it. The classic result of this type is van der Waerden's theorem, which asserts the following:

THEOREM 11.5.1 [vdW27]

If $X = \{1, 2, \dots, m\}$ then $\mathbb{E} \xrightarrow{r} \text{Hom}(X)$.

(Note that $\text{Hom}(X)$ is just the set of m -term arithmetic progressions.)

By the compactness theorem mentioned in the Introduction there exists, for each m , a minimum value $W(m)$ such that

$$\{1, 2, \dots, W(m)\} \xrightarrow{2} \text{Hom}(X).$$

The determination or even estimation of $W(m)$ seems to be extremely difficult. The known values are:

m	1	2	3	4	5
$W(m)$	1	3	9	35	178

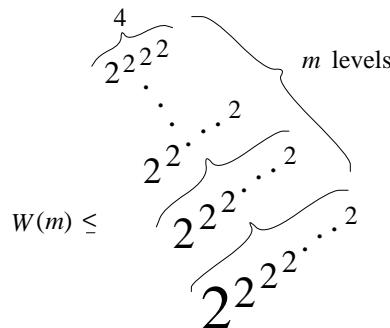
The best general result from below (due to Berlekamp—see [GRS90]) is

$$W(p+1) \geq p \cdot 2^p, \quad p \text{ prime.}$$

The best upper bound known follows from a spectacular result of Gowers [Gow01]:

$$W(m) < 2^{2^{2^{2^{m+9}}}}$$

This settled a long-standing \$1000 conjecture of the author. This result is a corollary of Gowers's new quantitative form of Szemerédi's theorem mentioned in the next section. It improves on the earlier bound of Shelah: [She88]:



The following conjecture of the author has been open for more than 30 years:

CONJECTURE 11.5.2 (\$1000)

For all m ,

$$W(m) \leq 2^{m^2}$$

The generalization to \mathbb{E}^N is due independently to Gallai and Witt (see [GRS90]).

THEOREM 11.5.3

For any finite set $X \subset \mathbb{E}^n$,

$$\mathbb{E}^N \longrightarrow \text{Hom}(X).$$

We remark here that a number of results in (Euclidean) Ramsey theory have stronger so-called *density* versions. As an example, we state the well-known theorem of Szemerédi.

GLOSSARY

\mathbb{N} : The set of natural numbers $\{1, 2, 3, \dots\}$.

$\overline{\delta}(A)$: The **upper density** of a set $A \subseteq \mathbb{N}$ is defined by:

$$\overline{\delta}(A) = \limsup_{n \rightarrow \infty} \frac{|A \cap \{1, 2, \dots, n\}|}{n}.$$

THEOREM 11.5.4 (Szemerédi [Sze75])

If $A \subseteq \mathbb{N}$ has $\overline{\delta}(A) > 0$ then A contains arbitrarily long arithmetic progressions.

That is, $A \cap \text{Hom}\{1, 2, \dots, m\} \neq \emptyset$ for all m . This clearly implies van der Waerden's theorem since $\mathbb{N} = C_1 \cup \dots \cup C_r \Rightarrow \max_i \overline{\delta}(C_i) \geq 1/r$.

Furstenberg [Fur77] has given a quite different proof of Szemerédi's theorem, using tools from ergodic theory and topological dynamics. This approach has proved to be very powerful, allowing Furstenberg, Katznelson, and others to prove density versions of the Hales-Jewett theorem (see [FK91]), the Gallai-Witt theorem, and many others. Recently, Gowers has given a strong quantitative version of Szemerédi's theorem:

THEOREM 11.5.5 [Gow01]

For every $k > 0$, any subset of $1, 2, \dots, N$ of size at least $N(\log \log N)^{-c(k)}$ contains a k -term arithmetic progression, where $c(k) = 2^{-2^{k+9}}$.

There are other ways of expressing the fact that A is relatively dense in \mathbb{N} besides the condition that $\overline{\delta}(A) > 0$. One would expect that these could also be used as a basis for a density version of van der Waerden or Gallai-Witt. Very little is currently known in this direction, however. We conclude this section with several conjectures of this type.

CONJECTURE 11.5.6 (Erdős)

If $A \subseteq \mathbb{N}$ satisfies $\sum_{a \in A} 1/a = \infty$ then A contains arbitrarily long arithmetic progressions.

CONJECTURE 11.5.7 (Graham)

If $A \subseteq \mathbb{N} \times \mathbb{N}$ with $\sum_{(x,y) \in A} 1/(x^2 + y^2) = \infty$ then A contains the 4 vertices of an axes-parallel square.

More generally, I expect that A will always contain a homothetic image of $\{1, 2, \dots, m\} \times \{1, 2, \dots, m\}$ for all m .

Finally, we mention a direction in which the group $SO(n)$ is enlarged to allow dilatations as well.

Definition: For a set $W \subseteq \mathbb{E}^k$, define the **upper density** $\overline{\delta}(W)$ of W by

$$\overline{\delta}(W) := \limsup_{R \rightarrow \infty} \frac{m(B(o, R) \cap W)}{m(B(o, R))},$$

where $B(o, R)$ denotes the k -ball $\left\{ (x_1, \dots, x_k) \in \mathbb{E}^k \mid \sum_{i=1}^k x_i^2 \leq R^2 \right\}$ centered at the origin, and m denotes Lebesgue measure.

THEOREM 11.5.8 (Bourgain [Bou86])

Let $X \subseteq \mathbb{E}^k$ be a simplex. If $W \subseteq \mathbb{E}^k$ with $\bar{\delta}(W) > 0$ then there exists t_0 such that for all $t > t_0$, W contains a congruent copy of tX .

Some restrictions on X are necessary as the following result shows.

THEOREM 11.5.9 (Graham [Gra94])

Let $X \subseteq \mathbb{E}^k$ be nonspherical. Then for any N there exist a set $W \subseteq \mathbb{E}^N$ with $\bar{\delta}(W) > 0$ and a set $T \subseteq \mathbb{R}$ with $\underline{\delta}(T) > 0$ such that W contains no congruent copy of tX for any $t \in T$.

Here $\underline{\delta}$ denotes **lower density**, defined similarly to $\bar{\delta}$ but with \liminf replacing \limsup .

It is clear that much remains to be done here.

11.6 VARIATIONS

There are quite a few variants of the preceding topics that have received attention in the literature (e.g., see [Sch93]). We mention some of the more interesting ones.

ASYMMETRIC RAMSEY THEOREMS

Typical results of this type assert that for given sets X_1 and X_2 (for example), for every partition of $\mathbb{E}^N = C_1 \cup C_2$, either C_1 contains a congruent copy of X_1 , or C_2 contains a congruent copy of X_2 . We can denote this by

$$\mathbb{E}^N \xrightarrow{2} (X_1, X_2).$$

Here is a sampling of results of this type (more of which can be found in [EGM⁺73], [EGM⁺75a], [EGM⁺75b]).

- (i) $\mathbb{E}^2 \xrightarrow{2} (T_2, T_3)$ where T_i is any subset of \mathbb{E}^2 with i points, $i = 2, 3$.
- (ii) $\mathbb{E}^2 \xrightarrow{2} (P_2, P_4)$ where P_2 is a set of two points at a distance 1, and P_4 is a set of four collinear points with distance 1 between consecutive points.
- (iii) $\mathbb{E}^3 \xrightarrow{2} (T, Q^2)$ where T is an isosceles right triangle and Q^2 is a square.
- (iv) $\mathbb{E}^2 \xrightarrow{2} (P_2, T_4)$ where P_2 is as in (ii) and T_4 is any set of four points [Juh79].
- (v) There is a set T_8 of 8 points such that

$$\mathbb{E}^2 \xrightarrow{2} (P_2, T_8) \quad [\text{CT94}].$$

This strengthens an earlier result of Juhász [Juh79], which proved this for a certain set of 12 points.

POLYCHROMATIC RAMSEY THEOREMS

Here, instead of asking for a copy of the target set X in a single C_i , we require only that it be contained in the union of a small number of C_i , say at most m of the C_i .

Let us indicate this by writing $\mathbb{E}^N \xrightarrow{m} X$.

- (i) If $\mathbb{E}^N \xrightarrow{m} X$ then X must be embeddable on the union of m concentric spheres [EGM⁺ 73].
- (ii) Suppose X_i is finite and $\mathbb{E}^N \xrightarrow{m_i} X_i$, $1 \leq i \leq t$. Then

$$\mathbb{E}^N \xrightarrow[m_1 m_2 \cdots m_t]{} X_1 \times X_2 \times \cdots \times X_t \quad [\text{ERS83}].$$

- (iii) If X_6 is the 6-point set formed by taking the four vertices of a square together with the midpoints of two adjacent sides then $\mathbb{E}^2 \not\xrightarrow{2} X_6$ but $\mathbb{E}^2 \xrightarrow{2} X_6$.
- (iv) If X is the set of vertices of a regular simplex in \mathbb{E}^N together with the trisection points of each of its edges then

$$\mathbb{E}^2 \not\xrightarrow{2} X_6 \quad \text{but} \quad \mathbb{E}^2 \xrightarrow{3} X_6.$$

It is not known if $\mathbb{E}^2 \xrightarrow{2} X_6$. Many other results of this type can be found in [ERS83].

PARTITIONS OF \mathbb{E}^n WITH ARBITRARILY MANY PARTS

Since $\mathbb{E}^2 \not\xrightarrow{7} P_2$, where P_2 is a set of two points with unit distance, one might ask whether there is any nontrivial result of the type $\mathbb{E}^2 \xrightarrow{m} \mathcal{F}$ when m is allowed to go to infinity. Of course, if \mathcal{F} is sufficiently large, then there certainly are. There are some interesting geometric examples for which \mathcal{F} is not too large.

THEOREM 11.6.1 [Gra80a]

For any partition of \mathbb{E}^n into finitely many parts, some part contains, for all $\alpha > 0$ and all sets of lines L_1, \dots, L_n that span \mathbb{E}^n , a simplex having volume α and edges through one vertex parallel to the L_i .

Many other theorems of this type are possible (see [Gra80a]).

PARTITIONS WITH INFINITELY MANY PARTS

Results of this type tend to have a strong set-theoretic flavor. For example: $\mathbb{E}^2 \not\xrightarrow{\aleph_0} T_3$ where T_3 is an equilateral triangle [Ced69]. In other words, \mathbb{E}^2 can be partitioned into countably many parts so that no part contains the vertices of an equilateral triangle. In fact, this was recently strengthened by Schmerl [Sch94b]

who showed that for all N ,

$$\mathbb{E}^N \xrightarrow{\aleph_0} T_3.$$

In fact, this result holds for *any* fixed triangle T in place of T_3 [Sch94b]. Schmerl also has shown [Sch94a] that there is a partition of \mathbb{E}^N into countably many parts such that no part contains the vertices of *any* isosceles triangle.

Another result of this type is this:

THEOREM 11.6.2 [Kun]

Assuming the Continuum Hypothesis, it is possible to partition \mathbb{E}^2 into countably many parts, none of which contains the vertices of a triangle with rational area.

We also note the interesting result of Erdős and Komjath:

THEOREM 11.6.3 [EK90]

The existence of a partition of \mathbb{E}^2 into countably many sets, none of which contains the vertices of a right triangle is equivalent to the Continuum Hypothesis.

The reader can consult Komjath [Kom97] for more results of this type.

COMPLEXITY ISSUES

S. Burr [Bur82] has shown that the algorithmic question of deciding if a given set $X \subset \mathbb{N} \times \mathbb{N}$ can be partitioned $X = C_1 \cup C_2 \cup C_3$ so that $x, y \in C_i \Rightarrow \text{distance}(x, y) \geq 6$, $i = 1, 2, 3$, is NP-complete. (Also, he shows that a certain infinite version of this is undecidable.)

Finally, we make a few remarks about the celebrated problem of Esther Klein (who became Mrs. Szekeres), which, in some sense, initiated this whole area (see [Sze73] for a charming history).

THEOREM 11.6.4 [ES35]

There is a minimum function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that any set of $f(n)$ points in \mathbb{E}^2 in general position contains the vertices of a convex n -gon.

This result of Erdős and George Szekeres actually spawned an independent genesis of Ramsey theory.

The best bounds currently known for $f(n)$ are:

$$2^{n-2} + 1 \leq f(n) \leq \binom{2n-5}{n-3} + 2.$$

The lower bound appears in [ES35], while the upper, improved by G. Tóth and P. Valtr from the original $\binom{2n-4}{n-2+1}$, appears in [TV98].

CONJECTURE 11.6.5

Prove (or disprove) that $f(n) = 2^{n-2} + 1$, $n \geq 3$.

(See [Chapter 1](#) of this Handbook for more details.)

11.7 SOURCES AND RELATED MATERIAL

SURVEYS

The principal surveys for results in Euclidean Ramsey theory are [GRS90], [Gra80b], [Gra85], and [Gra94]. The first of these is a monograph on Ramsey theory in general, with a section devoted to Euclidean Ramsey theory, while the last three are specifically about the topics discussed in the present chapter.

RELATED CHAPTERS

Chapter 1: Finite point configurations

Chapter 13: Geometric discrepancy theory and uniform distribution

REFERENCES

- [Bón93] M. Bóna. A Euclidean Ramsey theorem. *Discrete Math.*, 122:349–352, 1993.
- [BT96] M. Bóna and G. Tóth. A Ramsey-type problem on right-angled triangles in space. *Discrete Math.*, 150:61–67, 1996
- [Bou86] J. Bourgain. A Szemerédi type theorem for sets of positive density in \mathbb{R}^k . *Israel J. Math.*, 54:307–316, 1986.
- [Bur82] S.A. Burr. An NP-complete problem in Euclidean Ramsey theory. In *Proc. 13th Southeastern Conf. on Combinatorics, Graph Theory and Computing*, volume 35, pages 131–138, 1982.
- [Can96a] K. Cantwell. Finite Euclidean Ramsey theory. *J. Combin. Theory Ser. A*, 73:273–285, 1996.
- [Can96b] K. Cantwell. Edge-Ramsey theory. *Discrete Comput. Geom.*, 15:341–352, 1996.
- [Ced69] J. Ceder. Finite subsets and countable decompositions of Euclidean spaces. *Rev. Roumaine Math. Pures Appl.*, 14:1247–1251, 1969.
- [CFG91] H.T. Croft, K.J. Falconer, and R.K. Guy. *Unsolved Problems in Geometry*. Springer-Verlag, New York, 1991.
- [CT94] G. Csizmadia and G. Tóth. Note on a Ramsey-type problem in geometry. *J. Combin. Theory Ser. A*, 65:302–306, 1994.
- [EGM⁺73] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J. Spencer, and E.G. Straus. Euclidean Ramsey theorems. *J. Combin. Theory Ser. A*, 14:341–63, 1973.
- [EGM⁺75a] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J. Spencer, and E.G. Straus. Euclidean Ramsey theorems II. In A. Hajnal, R. Rado, and V. Sós, editors, *Infinite and Finite Sets I*, pages 529–557. North-Holland, Amsterdam, 1975.
- [EGM⁺75b] P. Erdős, R.L. Graham, P. Montgomery, B.L. Rothschild, J. Spencer, and E.G. Straus. Euclidean Ramsey theorems III. In A. Hajnal, R. Rado, and V. Sós, editors, *Infinite and Finite Sets II*, pages 559–583. North-Holland, Amsterdam, 1975.
- [EK90] P. Erdős and P. Komjáth. Countable decompositions of \mathbb{R}^2 and \mathbb{R}^3 . *Discrete Comput. Geom.* 5:325–331, 1990.

- [ERS83] P. Erdős, B. Rothschild, and E.G. Straus. Polychromatic Euclidean Ramsey theorems. *J. Geom.*, 20:28–35, 1983.
- [ES35] P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Math.*, 2:463–470, 1935.
- [FK91] H. Furstenberg and Y. Katznelson. A density version of the Hales-Jewett theorem. *J. Anal. Math.*, 57:64–119, 1991.
- [FR90] P. Frankl and V. Rödl. A partition property of simplices in Euclidean space. *J. Amer. Math. Soc.*, 3:1–7, 1990.
- [Fur77] H. Furstenberg. Ergodic behavior of diagonal measures and a theorem of Szemerédi on arithmetic progressions. *J. d'Anal. Math.*, 31:204–256, 1977.
- [FW81] P. Frankl and R.M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1:357–368, 1981.
- [Gow01] T. Gowers. A new proof of Szemerédi's theorem. *Geom. Funct. Anal.*, 11:465–588, 2001.
- [Gra80a] R.L. Graham. On partitions of \mathbb{E}^n . *J. Combin. Theory Ser. A*, 28:89–97, 1980.
- [Gra80b] R.L. Graham. Topics in Euclidean Ramsey theory. In J. Nešetřil and V. Rödl, editors, *Mathematics of Ramsey Theory*. Springer-Verlag, Heidelberg, 1980.
- [Gra83] R.L. Graham. Euclidean Ramsey theorems on the n -sphere. *J. Graph Theory*, 7:105–114, 1983.
- [Gra85] R.L. Graham. Old and new Euclidean Ramsey theorems. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440, Ann. New York Acad. Sci., pages 20–30. New York, 1985.
- [Gra94] R.L. Graham. Recent trends in Euclidean Ramsey theory. *Discrete Math.*, 136:119–127, 1994.
- [GRS90] R.L. Graham, B.L. Rothschild, and J. Spencer. *Ramsey Theory*, 2nd edition. Wiley, New York, 1990.
- [Juh79] R. Juhász. Ramsey type theorems in the plane. *J. Combin. Theory Ser. A*, 27:152–160, 1979.
- [Kom97] P. Komjáth. Set theory: geometric and real. *The mathematics of Paul Erdős, II*, volume 14 of *Algorithms Combin.*, pages 461–466. Springer-Verlag, Berlin, 1997.
- [Kři91] I. Kříž. Permutation groups in Euclidean Ramsey theory. *Proc. Amer. Math. Soc.*, 112:899–907, 1991.
- [Kři92] I. Kříž. All trapezoids are Ramsey. *Discrete Math.*, 108:59–62, 1992.
- [Kun] K. Kunen. Personal communication.
- [MR95] J. Matoušek and V. Rödl. On Ramsey sets on spheres. *J. Combin. Theory Ser. A*, 70:30–44, 1995.
- [Nech02] O. Nechushtan. A note on the space chromatic number. *Discrete Math.*, 256:499–507, 2002.
- [O'D00a] P. O'Donnell. Arbitrary girth, 4-chromatic unit distance graphs in the plane; Part 1: Graph Description. *Geombinatorics*, 9:145–150, 2000.
- [O'D00b] P. O'Donnell. Arbitrary girth, 4-chromatic unit distance graphs in the plane; Part 2: Graph Embedding. *Geombinatorics*, 9:180–193, 2000.
- [RT03] R. Radoičić and G. Tóth. Note on the chromatic number of the space. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift, Algorithms Combin.*, pages 695–698. Springer-Verlag, Berlin, 2003.
- [Sch93] P. Schmitt. Problems in discrete and combinatorial geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, volume A. North-Holland, Amsterdam, 1993.

- [Sch94a] J.H. Schmerl. Personal communication, 1994.
- [Sch94b] J.H. Schmerl. Triangle-free partitions of Euclidean space. *Bull. London Math. Soc.*, 26:483–486, 1994.
- [Sha76] L. Shader. All right triangles are Ramsey in \mathbb{E}^2 ! *J. Combin. Theory Ser. A*, 20:385–389, 1976.
- [She88] S. Shelah. Primitive recursive bounds for van der Waerden numbers. *J. Amer. Math. Soc.*, 1:683–697, 1988.
- [Soi91] A. Soifer. Chromatic number of the plane: A historical survey. *Geombinatorics*, 1:13–14, 1991.
- [Soi92] A. Soifer. A six-coloring of the plane. *J. Combin. Theory Ser. A*, 61:292–294, 1992.
- [SW89] L.A. Székely and N. Wormald. Bounds on the measurable chromatic number of \mathbb{R}^n . *Discrete Math.*, 75:343–372, 1989.
- [Sze73] G. Szekeres. A combinatorial problem in geometry: Reminiscences. In J. Spencer, editor, *Paul Erdős: The Art of Counting, Selected Writings*, pages xix–xxii. The MIT Press, Cambridge, 1973.
- [Sze75] E. Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arith.*, 27:199–245, 1975.
- [Tót96] G. Tóth. A Ramsey-type bound for rectangles. *J. Graph Theory*, 23:53–56, 1996.
- [TV98] G. Tóth and P. Valtr. Note on the Erdős-Szekeres theorem. In J. Pach, editor, Erdős Memorial Issue, *Discrete Comput. Geom.*, 19:457–459, 1998.
- [vdW27] B.L. van der Waerden. Beweis einer Baudetschen Vermutung. *Nieuw Arch. Wisk.*, 15:212–216, 1927.

12 DISCRETE ASPECTS OF STOCHASTIC GEOMETRY

Rolf Schneider

INTRODUCTION

Stochastic geometry studies randomly generated geometric objects. The present chapter deals with some discrete aspects of stochastic geometry. We describe work that has been done on familiar objects of discrete geometry, such as finite point sets, their convex hulls, discrete point sets, arrangements of flats, and tessellations of space, under various assumptions of randomness. Most of the results to be mentioned concern expectations of geometrically defined random variables, or probabilities of events defined by random geometric configurations. The selection of topics must necessarily be restrictive. We leave out the great number of special elementary geometric probability problems which can be solved explicitly by direct, though possibly intricate, analytic calculations. We pay special attention to either asymptotic results, where the number of points considered tends to infinity, or to inequalities, or to identities where the proofs involve more delicate geometric or combinatorial arguments. The close ties of discrete geometry to convexity are reflected: we consider convex hulls of random points, intersections of random halfspaces, and tessellations of space into convex sets generated either by discrete random hyperplane systems or, as Voronoi or Delaunay mosaics, by discrete random point sets. Topics not covered are, for example, optimization problems with random data, and the average-case analysis of geometric algorithms.

12.1 CONVEX HULLS OF RANDOM POINTS

The setup for this section is a finite number of random points in a topological space Σ . Mostly the space Σ is \mathbb{R}^d , the d -dimensional Euclidean space, with scalar product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$. Other spaces that occur are the sphere $S^{d-1} := \{x \in \mathbb{R}^d \mid \|x\| = 1\}$ or more general submanifolds of \mathbb{R}^d . By $B^d := \{x \in \mathbb{R}^d \mid \|x\| \leq 1\}$ we denote the unit ball of \mathbb{R}^d . The volume of B^d is denoted by κ_d .

GLOSSARY

Random point in Σ : A Borel-measurable mapping from some probability space into Σ .

Distribution of a random point X in Σ : The probability measure μ on Σ such that $\mu(B)$, for a Borel set $B \subset \Sigma$, is the probability that $X \in B$.

i.i.d. random points: Stochastically independent random points (on the same probability space) with the same distribution.

NOTATION

X_1, \dots, X_n	i.i.d. random points in \mathbb{R}^d
μ	the common probability distribution of X_i
φ	a measurable real function defined on polytopes in \mathbb{R}^d
$\varphi(\mu, n)$	the random variable $\varphi(\text{conv}\{X_1, \dots, X_n\})$
$\varphi(K, n)$	$= \varphi(\mu, n)$, if μ is the uniform distribution in K
\mathbb{E}	expectation of a random variable
f_k	number of k -faces
ψ_j	1 on polytopes with j vertices, 0 otherwise
V_j	j th intrinsic volume (see Chapter 16); in particular:
V_d	d -dimensional volume
S	$= 2V_{d-1}$, surface area; dS element of surface area
$D_j(K, n)$	$= V_j(K) - V_j(K, n)$

12.1.1 DISTRIBUTION-INDEPENDENT RESULTS

There are a few general results on convex hulls of i.i.d. random points in \mathbb{R}^d that do not require special assumptions on the distribution of these points. A classical result due to Wendel [Wen62] concerns the probability, say $p_{d,n}$, that $0 \notin \text{conv}\{X_1, \dots, X_n\}$. If the distribution of the i.i.d. random points $X_1, \dots, X_n \in \mathbb{R}^d$ is symmetric with respect to 0 and assigns measure zero to every hyperplane through 0, then

$$p_{d,n} = \frac{1}{2^{n-1}} \sum_{k=0}^{d-1} \binom{n-1}{k}. \quad (12.1.1)$$

This follows from a combinatorial result of Schläfli, on the number of d -dimensional cells in a partition of \mathbb{R}^d by n hyperplanes through 0 in general position. It was proved surprisingly late that the symmetric distributions are extremal: Wagner and Welzl [WaW01] showed that if the distribution of the points is absolutely continuous with respect to Lebesgue measure, then $p_{d,n}$ is at least the right-hand side of (12.1.1).

The expected values $\mathbb{E}V_d(\mu, n)$ for different numbers n are connected by a sequence of identities. For an arbitrary probability distribution μ on \mathbb{R}^d , Buchta [Buc90] proved the recurrence relations

$$\mathbb{E}V_d(\mu, d+2m) = \frac{1}{2} \sum_{k=1}^{2m-1} (-1)^{k+1} \binom{d+2m}{k} \mathbb{E}V_d(\mu, d+2m-k)$$

and, consequently,

$$\mathbb{E}V_d(\mu, d+2m) = \sum_{k=1}^m (2^{2k} - 1) \frac{B_{2k}}{k} \binom{d+2m}{2k-1} \mathbb{E}V_d(\mu, d+2m-2k+1)$$

for $m \in \mathbb{N}$, where the constants B_{2k} are the Bernoulli numbers.

12.1.2 NATURAL DISTRIBUTIONS

In geometric problems about random points, a few distributions have been considered as particularly natural, for different reasons. Such reasons may be invariance

properties, or relations to measures of geometric significance, but there are also more subtle viewpoints, as explained, for example, in Ruben and Miles [RuM80]. The distributions of a random point in \mathbb{R}^d shown in Table 12.1.1 underlie many investigations.

TABLE 12.1.1 Natural distributions of a random point in \mathbb{R}^d .

NAME OF DISTRIBUTION	PROBABILITY DENSITY AT $x \in \mathbb{R}^d$
Uniform in K	\propto indicator function of K at x
Standard normal	$\propto \exp\left(-\frac{1}{2}\ x\ ^2\right)$
Beta type 1	$\propto (1 - \ x\ ^2)^q \times$ indicator function of B^d at x , $q > -1$
Beta type 2	$\propto \ x\ ^{\alpha-1}(1 + \ x\)^{-(\alpha+\beta)}$, $\alpha, \beta > 0$
Spherically symmetric	function of $\ x\ $

Here $K \subset \mathbb{R}^d$ is a given closed set of positive, finite volume, often a **convex body** (a compact, convex set with interior points). Usually the name of the distribution of a random point is also associated with the random point itself. General rotationally symmetric distributions have mostly been considered under additional tail assumptions. If F is a smooth compact hypersurface in \mathbb{R}^d , a random point is uniform on F if its distribution is proportional to the area measure on F . This distribution is particularly natural for the unit sphere S^{d-1} , since it is the unique rotation-invariant probability measure on S^{d-1} .

For combinatorial problems about n -tuples of random points in \mathbb{R}^d , the following approach leads to a natural distribution. Every configuration of $n > d$ numbered points in general position in \mathbb{R}^d is affinely equivalent to the orthogonal projection of the set of numbered vertices of a fixed regular simplex $T^{n-1} \subset \mathbb{R}^{n-1}$ onto a unique d -dimensional linear subspace of \mathbb{R}^{n-1} . This establishes a one-to-one correspondence between the (orientation-preserving) affine equivalence classes of such configurations and an open dense subset of the Grassmannian $G(n-1, d)$ of oriented d -spaces in \mathbb{R}^{n-1} . The unique rotation-invariant probability measure on $G(n-1, d)$ thus leads to a probability distribution on the set of affine equivalence classes of n -tuples of points in general position in \mathbb{R}^d . References for this **Grassmann approach**, which was proposed by Vershik and by Goodman and Pollack, are given in Affentranger and Schneider [AfS92]. Baryshnikov and Vitale [BaV94] proved that an affine-invariant functional of n -tuples with this distribution is stochastically equivalent to the same functional taken at an i.i.d. n -tuple of standard normal points in \mathbb{R}^d . Baryshnikov [Bar97] has made clear, in a strong sense, the unique role that is played in this correspondence by the vertex sets of regular simplices.

12.1.3 UNIFORM RANDOM POINTS IN CONVEX BODIES

A considerable amount of work has been done on convex hulls of a finite number of i.i.d. random points with uniform distribution in a given convex body K in \mathbb{R}^d .

Some of the expectations of $\varphi(K, n)$ for different functions φ are connected by

identities. Two classical results of Efron [Efr65],

$$\mathbb{E}\psi_{d+1}(K, n) = \binom{n}{d+1} \frac{\mathbb{E}V_d^{n-d-1}(K, d+1)}{V_d^{n-d-1}(K)} \quad (12.1.2)$$

and

$$\mathbb{E}f_0(K, n+1) = \frac{n+1}{V_d(K)} \mathbb{E}D_d(K, n), \quad (12.1.3)$$

have found far-reaching generalizations in work of Buchta's [Buc02]. He extended (12.1.3) to higher moments of the volume, showing that

$$\frac{\mathbb{E}V_d^k(K, n)}{V_d^k(K)} = \mathbb{E} \prod_{i=1}^k \left(1 - \frac{f_0(K, n+k)}{n+i}\right)$$

for $k \in \mathbb{N}$. As a consequence, the k th moment of $V_d(K, n)$ can be expressed linearly by the first k moments of $f_0(K, n+k)$. Further consequences are variance estimates for $D_d(K, n)$ and $f_0(K, n)$ for sufficiently smooth convex bodies K .

Of combinatorial interest is the expectation $\mathbb{E}\psi_i(K, n)$, which is the probability that the convex hull of n i.i.d. uniform random points in K has exactly i vertices. For this, Buchta [Buc02] proved that

$$\mathbb{E}\psi_i(K, n) = (-1)^i \binom{n}{i} \sum_{j=1}^i (-1)^j \binom{i}{j} \frac{\mathbb{E}V_d^{n-j}(K, j)}{V_d^{n-j}(K)},$$

which for $i = d+1$ reduces to (12.1.2). Sylvester's classical problem asked for $\mathbb{E}\psi_3(K, 4)$ (or the complementary probability) for a convex body $K \subset \mathbb{R}^2$. More generally, one may ask for $\mathbb{E}\psi_{d+1}(K, n)$ for a convex body $K \subset \mathbb{R}^d$ and $n > d+1$, the probability that the convex hull of n i.i.d. uniform points in K is a simplex. From (12.1.2) and results of Miles [Mil71b], the values $\mathbb{E}\psi_{d+1}(B^d, n)$ are known. At the other end, $\mathbb{E}\psi_n(K, n)$ is of interest, the probability that n i.i.d. uniform points in K are in convex position. Valtr [Val95] determined $\mathbb{E}\psi_n(P, n)$ if P is a parallelogram, and in [Val96] if P is a triangle. For convex bodies $K \subset \mathbb{R}^2$ of area one, Bárány [Bár99] obtained the astonishing limit relation

$$\lim_{n \rightarrow \infty} n^2 \sqrt[n]{\mathbb{E}\psi_n(K, n)} = \frac{1}{4} e^2 A^3(K),$$

where $A(K)$ is the supremum of the affine perimeters of all convex bodies contained in K . Bárány has even established a law of large numbers for convergence to a limit shape. There is a unique convex body $\tilde{K} \subset K$ with affine perimeter $A(K)$. If K_n denotes the convex hull of n i.i.d. uniform points in K and δ is the Hausdorff metric, then Bárány's result says that

$$\lim_{n \rightarrow \infty} \text{Prob}\{\delta(K_n, \tilde{K}) > \epsilon \mid f_0(K_n) = n\} = 0$$

for every $\epsilon > 0$.

For balls in increasing dimensions, earlier work of Buchta was extended by Bárány and Füredi [BáF88], who proved that

$$\begin{aligned} \mathbb{E}\psi_{n(d)}(B^d, n(d)) &\rightarrow 1 & \text{if } n(d) = 2^{d/2}d^{-\epsilon}, \\ \mathbb{E}\psi_{m(d)}(B^d, m(d)) &\rightarrow 0 & \text{if } m(d) = 2^{d/2}d^{(3/4)+\epsilon} \end{aligned}$$

when $d \rightarrow \infty$, for every fixed $\epsilon > 0$. The authors also investigated k -neighborliness of the convex hull.

We turn to random variables $\varphi(K, n)$ connected with intrinsic volumes and face numbers. First we mention the rare instances where information on the whole distribution is available. Some special results for $d = 2$ due to Alagar, Reed, and Henze are quoted in [Sch88, Section 4]. For example, Henze showed that the distribution function F_K of $V_2(K, 3)$ for a convex body $K \subset \mathbb{R}^2$ satisfies $F_T \leq F_K \leq F_E$, where T is a triangle and E is an ellipse, provided that K, T, E have the same area. Results on the distribution of $V_r(B^d, r + 1)$ for $r = 1, \dots, d$ are listed in a more general context in Section 12.1.5.

In the plane, a few remarkable central limit type theorems have been obtained. For a convex polygon $P \subset \mathbb{R}^2$ with r vertices, Groeneboom [Gro88] proved that

$$\frac{f_0(P, n) - \frac{2}{3}r \log n}{\sqrt{\frac{10}{27}r \log n}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$$

for $n \rightarrow \infty$, where $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution and $\mathcal{N}(0, 1)$ is the standard normal distribution. From this, Massé [Mas00] deduced that

$$\lim_{n \rightarrow \infty} \frac{3f_0(P, n)}{2r \log n} = 1 \quad \text{in probability.}$$

For the circular disk, Groeneboom showed

$$\frac{f_0(B^2, n) - 2\pi c_1 n^{1/3}}{\sqrt{2\pi c_2 n^{1/3}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$$

with $c_1 = (\frac{2}{3\pi})^{1/3} \Gamma(\frac{5}{3}) \approx 0.53846$ and c_2 given by an integral which was evaluated numerically. For a polygon P with r vertices, a result of Cabo and Groeneboom [CaG94], in a version suggested by Buchta [Buc02], says that

$$\frac{V_2(P)^{-1} D_2(P, n) - \frac{2}{3}r \frac{\log n}{n}}{\sqrt{\frac{28}{27}r \frac{\log n}{n^2}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1).$$

For the circular disk, a result of Hsing [Hsi94] was made more explicit by Buchta [Buc02] and now gives, as $n \rightarrow \infty$, $\pi^{-2} \text{var} D_2(B^2, n) \sim 2\pi(\frac{1}{3}c_1 + c_2)n^{-5/3}$ and

$$\frac{\pi^{-1} D_2(B^2, n) - 2\pi c_1 n^{-2/3}}{\sqrt{2\pi(\frac{1}{3}c_1 + c_2)n^{-5/3}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1).$$

A thorough study of the asymptotic properties of $D_2(\mu, n)$ and $D_1(\mu, n)$ was presented by Bräker and Hsing [BH98], for rather general distributions μ (including the uniform distribution) concentrated on a convex body K in the plane, where K is either sufficiently smooth and of positive curvature or a polygon. Bräker, Hsing, and Bingham [BHB98] investigated the asymptotic distribution of the Hausdorff distance between a planar convex body K (either smooth or a polygon) and the convex hull of n i.i.d. uniform points in K .

Küfer [Küf94] studied the asymptotic behavior of $D_d(B^d, n)$ and showed, in particular, that its variance is at most of order $n^{-(d+3)/(d+1)}$, as $n \rightarrow \infty$.

Most of the known results about the random variables $\varphi(K, n)$ concern their expectations. Explicit formulas for $\mathbb{E} \varphi(K, n)$ for convex bodies $K \subset \mathbb{R}^d$ and arbitrary $n \geq d + 1$ are known in the cases listed in [Table 12.1.2](#).

 TABLE 12.1.2 Expected value of $\varphi(K, n)$.

DIMENSION d	CONVEX BODY K	FUNCTIONAL φ	SOURCES
2	polygon	V_2	Buchta [Buc84a]
2	polygon	f_0	Buchta and Reitzner [BuR97a]
2	ellipse	V_2	Buchta [Buc84b]
3	ellipsoid	V_3	Buchta [Buc84b]
≥ 2	ball	S , mean width, f_{d-1}	Buchta and Müller [BuM84]
≥ 2	ball	V_d	Affentranger [Aff88]

Affentranger's result is given in the form of an integral, which can be evaluated for given d and n ; it implies the corresponding result for ellipsoids.

A well-known problem, popularized by Klee, is the explicit determination of $\mathbb{E}V_d(T^d, d+1)$ for a d -simplex T^d . Klee's opinion that $\mathbb{E}V_3(T^3, 4)$ "might yield to brute force" was justified. The result

$$\mathbb{E}V_3(T^3, 4) = \frac{13}{720} - \frac{\pi^2}{15015} = 0.0173982\dots \quad (12.1.4)$$

was announced by Buchta and Reitzner [BuR92], as well as a more general formula for $\mathbb{E}V_3(T^3, n)$. Independently, (12.1.4) was established by Mannion [Man94], who made heavy use of computer algebra. Finally, Buchta and Reitzner [BuR01] published a readable version of their admirable proof for the formula

$$\mathbb{E}V_3(T^3, n) = p_n - \pi^2 r_n,$$

with explicitly given rational numbers p_n, r_n .

If explicit formulas for $\mathbb{E}\varphi(K, n)$ are not available, one can try to obtain inequalities or asymptotic expansions for increasing n . For $\mathbb{E}V_d(K, n)$, the following estimates are known. The quotient $\mathbb{E}V_d(K, n)/V_d(K)$, for $n \geq d+1$, is minimal for ellipsoids (Groemer). The conjecture that it is maximal for simplices is only proved for $d=2$. (References for these and related results are given in the survey part of [BaS95].) If the convex body $K \subset \mathbb{R}^d$ is not a simplex, then the quotient $\mathbb{E}V_d(K, n)/V_d(K)$ is strictly less than its value for a simplex, for all $n \geq n_0(K)$, see [BáB93]. If $V_d(K) = 1$ and f is a continuous strictly increasing function, the expectation $\mathbb{E}f(V_j(K, n))$ is minimal if K is a ball; this was proved by Hartzoulaki and Paouris [HaP03].

We turn to asymptotic results for expectations. Buchta [Buc84c] considered the perimeter and proved for plane polygons P that

$$\mathbb{E}D_1(P, n) = c(P) \left(\frac{n}{V_2(P)} \right)^{-1/2} + o(n^{\epsilon-1})$$

for any fixed $\epsilon > 0$, where the constant $c(P)$ is given explicitly in terms of the angles of P . For a convex polygon P with r vertices (and area one), Buchta and Reitzner [BuR97a] obtained

$$\mathbb{E}f_0(P, n) = \frac{2r}{3} \log n + c_0(P) + \frac{c_1(P)}{n} + \frac{c_2(P)}{n^2} + \dots$$

as $n \rightarrow \infty$, with explicit constants $c_i(K)$; this strengthens a result of Rényi and Sulanke. Further work of the latter authors for the plane is described in [Sch88,

Section 5], as well as some particular results for \mathbb{R}^d , in part superseded by the following ones. For d -dimensional polytopes P , Bárány and Buchta [BáB93] were able to show that

$$\mathbb{E}f_0(P, n) = \frac{T(P)}{(d+1)^{d-1}(d-1)!} \log^{d-1} n + O(\log^{d-2} n \log \log n), \quad (12.1.5)$$

where $T(P)$ denotes the number of chains $F_0 \subset F_1 \subset \dots \subset F_{d-1}$ where F_i is an i -dimensional face of P . They establish a corresponding relation for the volume, from which (12.1.5) follows by (12.1.3). This work was the culmination of a series of papers by other authors, among them Affentranger and Wieacker [AfW91], who settled the case of simple polytopes, which is applied in [BáB93]. Bárány and Buchta mention that their methods permit one to extend (12.1.5) to $\mathbb{E}f_k(P, n)$ for $k = 0, \dots, d-1$, with the denominator replaced by a constant depending on d and k .

For convex bodies $K \subset \mathbb{R}^d$ with a boundary of class C^3 and positive Gauss-Kronecker curvature κ , Bárány [Bár92] obtained relations of the form

$$\mathbb{E}D_j(K, n) = c_2^{(j,d)}(K)n^{-2/(d+1)} + O(n^{-3/(d+1)} \log^2 n) \quad (12.1.6)$$

for $j = 1, \dots, d$. For $j = 1$, such a result (with explicit $c_2^{(1,d)}$) was obtained earlier by Schneider and Wieacker [ScWi80]. For simplicity, one can assume that $V_d(K) = 1$. Then, for $j = d$, the coefficient is given by

$$c_2^{(d,d)}(K) = c_2^{(d,d)} \int_{\partial K} \kappa^{1/(d+1)} dS$$

and thus is a constant multiple ($c_2^{(d,d)}$ depending only on d) of the affine surface area of K . The limit relation

$$\lim_{n \rightarrow \infty} n^{2/(d+1)} \mathbb{E}D_d(K, n) = c_2^{(d,d)} \int_{\partial K} \kappa^{1/(d+1)} dS$$

was extended by Schütt [Schü94] to arbitrary convex bodies (of volume one), with the Gauss-Kronecker curvature generalized accordingly. The other coefficients $c_2^{(j,d)}(K)$ in (12.1.6) are given by

$$c_2^{(j,d)}(K) = c_2^{(j,d)} \int_{\partial K} \kappa^{1/(d+1)} H_{d-j} dS,$$

where H_{d-j} denotes the $(d-j)$ th normalized elementary symmetric function of the principal curvatures of ∂K and $c_2^{(j,d)}$ depends only on j and d . These values were given by Reitzner [Rei01b], thus correcting the coefficients shown in [Bár92].

Under stronger differentiability assumptions, more precise asymptotic expansions are possible. If K has a boundary of class C^{k+3} , $k \geq 2$, and positive curvature (and is of volume one), then

$$\begin{aligned} & \mathbb{E}D_j(K, n) \\ &= c_2^{(j,d)}(K)n^{-2/(d+1)} + \dots + c_k^{(j,d)}(K)n^{-k/(d+1)} + O(n^{-(k+1)/(d+1)}) \end{aligned}$$

as $n \rightarrow \infty$, where additional information on the coefficients is available. This was proved by Reitzner [Rei01b] (for $d = j = 2$, see also Reitzner [Rei01a]). Under the same assumptions on K , Gruber [Gru96] had obtained earlier an analogous asymptotic expansion for $\eta(C) - \mathbb{E}\eta(C, n)$, where $\eta(C)$ is the value of the support function of the convex body C at a given vector $u \in S^{d-1}$.

For general convex bodies K , the approximation behavior is typically irregular, hence the main interest will be in sharp estimates. A first result concerns V_1 (essentially the mean width). For a convex body $K \subset \mathbb{R}^d$, Schneider [Sch87] showed the existence of positive constants $a_1(K), a_2(K)$ such that

$$a_1(K)n^{-2/(d+1)} < \mathbb{E}D_1(K, n) < a_2(K)n^{-1/d} \quad (12.1.7)$$

for $n \in \mathbb{N}$. Smooth bodies (left) and polytopes (right) show that the orders are best possible.

For general convex bodies K , a powerful method for investigating the polytopes $K_n := \text{conv}\{X_1, \dots, X_n\}$, for i.i.d. uniform points X_1, \dots, X_n in K , was invented by Bárány and Larman [BáL88]. For K of volume one and for sufficiently small $t > 0$, they introduced the floating body $K[t] := \{x \in K \mid V_d(K \cap H) \geq t \text{ for every halfspace } H \text{ with } x \in H\}$. Their main result says that K_n and $K[1/n]$ approximate K of the same order and that $K \setminus K_n$ is close to $K \setminus K[1/n]$ in a precise sense. From this, several results on the expectations $\mathbb{E}\varphi(K, n)$ for various φ were obtained by Bárány and Larman [BáL88], by Bárány [Bár89], for example

$$c_1(d)(\log n)^{d-1} < \mathbb{E}f_j(K, n) < c_2(d)n^{(d-1)/(d+1)} \quad (12.1.8)$$

for $j \in \{0, \dots, d\}$ with positive constants $c_i(d)$ (the orders are best possible), and by Bárány and Vitale [BáV93].

The inequalities (12.1.7) show that for general K the approximation, measured in terms of $D_1(K, \cdot)$, is not worse than for polytopes and not better than for smooth bodies. For approximation measured by $D_d(K, \cdot)$, the class of polytopes and the class of smooth bodies interchange their roles, since

$$b_1(K)n^{-1}(\log n)^{d-1} < \mathbb{E}D_d(K, n) < b_2(K)n^{-2/(d+1)},$$

as follows from [BáL88] (or from (12.1.8) for $j = 0$ and (12.1.3)). This observation lends additional interest to Problem 12.1.3 below.

OPEN PROBLEMS

PROBLEM 12.1.1 (Valtr [Val96])

Is it true, for a convex body $K \subset \mathbb{R}^2$ and for $n \geq 4$, that $\mathbb{E}\psi_n(K, n)$, the probability that n uniform i.i.d. points in K are in convex position, is minimal if K is a triangle and maximal if K is an ellipse?

PROBLEM 12.1.2

For a convex body $K \subset \mathbb{R}^d$ with a boundary of class C^3 and positive Gauss-Kronecker curvature κ , and for the numbers of k -faces, one expects that

$$\mathbb{E}f_k(K, n) = b(d, k) \int_{\partial K} \kappa^{1/(d+1)} dS \left(\frac{n}{V_d(K)} \right)^{(d-1)/(d+1)} (1 + o(1)) \quad (12.1.9)$$

with a constant $b(d, k)$. For $k = 0$, this follows from (12.1.6); for $k = d - 1$ (which implies the case $k = d - 2$) the result goes back to Raynaud and Wieacker; see [Bár92] and [Sch88, p. 222] for references.

PROBLEM 12.1.3 (Bárány [Bár89])

Is it true for a general convex body $K \subset \mathbb{R}^d$ that the surface area satisfies

$$c_1(K)n^{-1/2} < \mathbb{E}D_{d-1}(K, n) < c_2(K)n^{-2/(d+1)}$$

with positive constants $c_1(K), c_2(K)$?

12.1.4 RANDOM POINTS ON CONVEX SURFACES

If μ is a probability distribution on the boundary ∂K of a convex body K and μ has density h with respect to the area measure of ∂K , we write $\varphi(\mu, n) := \varphi(\partial K, h, n)$ and $D_j(\partial K, h, n) := V_j(K) - V_j(\partial K, h, n)$. Some references concerning $\mathbb{E}\varphi(\partial K, h, n)$ are given in [Sch88, p. 224]. Most of them are superseded by an investigation of Reitzner [Rei02a]. For K with a boundary of class C^2 and positive Gauss curvature, $j \in \{1, \dots, d\}$, and continuous $h > 0$, he showed that

$$\mathbb{E}D_j(\partial K, h, n) = b_2^{(j,d)} \int_{\partial K} h^{-2/(d-1)} \kappa^{1/(d-1)} H_{d-j} dS \cdot n^{-2/(d-1)} + o(n^{-2/(d-1)})$$

as $n \rightarrow \infty$. Under stronger differentiability assumptions on K and h , an asymptotic expansion with more terms was established. Similar results for support functions were obtained earlier by Gruber [Gru96].

For $j = d$, there is an asymptotic relation for general convex bodies K satisfying only a weak regularity assumption. In a long and intricate proof, Schütt and Werner [ScWe03] proved that

$$\lim_{n \rightarrow \infty} n^{2/(d-1)} \mathbb{E}D_d(\partial K, h, n) = b_2^{(d,d)} \int_{\partial K} h^{-2/(d-1)} \kappa^{1/(d-1)} dS,$$

provided that the lower and upper curvatures of K are between two fixed positive and finite bounds.

Let $(X_k)_{k \in \mathbb{N}}$ be an i.i.d. sequence of uniform random points on the boundary ∂K of a convex body K . Let δ denote the Hausdorff metric. Dürmbgen and Walther [DüW96] showed that $\delta(K, \text{conv}\{X_1, \dots, X_n\})$ is almost surely of order $O((\log n/n)^{1/(d-1)})$ for general K , and of order $O((\log n/n)^{2/(d-1)})$ under a smoothness assumption.

OPEN PROBLEM

PROBLEM 12.1.4

Let $K \subset \mathbb{R}^d$ be a convex body with a boundary of class C^3 and positive Gauss-Kronecker curvature κ . Let $(X_k)_{k \in \mathbb{N}}$ be an i.i.d. sequence of random points in ∂K , the distribution of which has a continuous positive density h with respect to the area measure. We conjecture that

$$\lim_{n \rightarrow \infty} \left(\frac{n}{\log n} \right)^{2/(d-1)} \delta(K, \text{conv}\{X_1, \dots, X_n\}) = \frac{1}{2} \left(\frac{1}{\kappa_{d-1}} \max \frac{\sqrt{\kappa}}{h} \right)^{2/(d-1)}$$

with probability one. For $d = 2$, this is true, and similar results hold with the Hausdorff distance replaced by area or perimeter difference; this was proved by Schneider [Sch88]. For $d > 2$ and with convergence in probability instead of almost sure convergence, the result was proved by Glasauer and Schneider [GIS96].

12.1.5 CONVEX HULLS FOR OTHER DISTRIBUTIONS

Convex hulls of i.i.d. random points have been investigated for each of the distributions listed in Table 12.1.1, and occasionally for more general ones. The following setup has been studied repeatedly. For $0 \leq p \leq r + 1 \leq d - 1$, one considers $r + 1$ independent random points, of which the first p are uniform in the ball B^d and the last $r + 1 - p$ are uniform on the boundary sphere S^{d-1} . Precise information on the moments and the distribution of the r -dimensional volume of the convex hull is available; see the references in [Sch88, pp. 219, 224] and the work of Affentranger [Aff88].

Among spherically symmetric distributions, the beta distributions are particularly tractable. For these, again, the r -dimensional volume of the convex hull of $r + 1$ i.i.d. random points has frequently been studied. We refer to the references given in [Sch88] and Chu [Chu93]. Affentranger [Aff91] determined the asymptotic behavior, as $n \rightarrow \infty$, of the expectation $\mathbb{E}V_j(\mu, n)$, where μ is either the beta type-1 distribution, the uniform distribution in B^d , or the standard normal distribution in \mathbb{R}^d . The asymptotic behavior of $\mathbb{E}f_{d-1}(\mu, n)$ was also found for these cases. Further information is contained in the book of Mathai [Mat99].

For normally distributed points in the plane, Hueter [Hue94] proved central limit type results for the number of vertices, the perimeter, and the area of the convex hull. For $d \geq 2$, she obtained in [Hue99] a central limit theorem for $f_0(\mu, n)$, for a class of spherically symmetric distributions μ in \mathbb{R}^d including the normal family. For the normal distribution μ_2 in the plane, Massé [Mas00] derived from [Hue94] that

$$\lim_{n \rightarrow \infty} \frac{f_0(\mu_2, n)}{\sqrt{8\pi \log n}} = 1 \quad \text{in probability.}$$

For the expectations $\mathbb{E}f_k(\mu_d, n)$ ($k \in \{0, \dots, d-1\}$), where μ_d is the standard normal distribution in \mathbb{R}^d , one knows that

$$\mathbb{E}f_k(\mu_d, n) \sim \frac{2^d}{\sqrt{d}} \binom{d}{k+1} \beta_{k,d-1} (\pi \log n)^{(d-1)/2} \quad (12.1.10)$$

as $n \rightarrow \infty$, where $\beta_{k,d-1}$ is the interior angle of the regular $(d-1)$ -dimensional simplex at one of its k -dimensional faces. This follows from [AfS92], where the Grassmann approach was used, due to the equivalence of [BaV94] explained in Section 12.1.2. For the Grassmann approach, Vershik and Sporyshev [VeS92] have made a careful study of the asymptotic behavior of the number of k -faces, if k and the dimension d grow linearly with the number n .

Relation (12.1.10) also describes the asymptotic behavior of the number of k -faces of the orthogonal projection of a regular $(n-1)$ -simplex onto a randomly chosen isotropic d -subspace. In a similar investigation, Böröczky and Henk [Böh99] replaced the regular simplex by the regular crosspolytope and found, surprisingly, the same asymptotic behavior.

For more general spherically symmetric distributions μ , the asymptotic behavior of the random variables $\varphi(\mu, n)$ will essentially depend on the tail behavior

of the distribution. Extending work of Carnal (1970), Dwyer [Dwy91] obtained asymptotic estimates for $\mathbb{E}f_0(\mu, n)$, $\mathbb{E}f_{d-1}(\mu, n)$, $\mathbb{E}V_d(\mu, n)$, and $\mathbb{E}S(\mu, n)$. Devroye [Dev91] showed that for any monotone sequence $\omega_n \uparrow \infty$ and for every $\epsilon > 0$, there is a radially symmetric distribution μ in the plane for which $\mathbb{E}f_0(\mu, n) \geq n/\omega_n$ infinitely often and $\mathbb{E}f_0(\mu, n) \leq 4 + \epsilon$ infinitely often. For $\omega_n \uparrow \infty$ strictly increasing and satisfying $\omega_n \leq n$, Massé [Mas99] constructed a distribution μ in the plane such that the variance satisfies $\text{var}f_0(\mu, n) \geq n^2/\omega_n$ infinitely often. Aldous *et al.* [AlFGP91] considered an i.i.d. sequence $(X_k)_{k \in \mathbb{N}}$ in \mathbb{R}^2 with a spherically symmetric (or more general) distribution. Under an assumption of slowly varying tail, they determined a limiting distribution for $f_0(\mu, n)$. Massé [Mas00] constructed a distribution μ in the plane for which $\mathbb{E}f_0(\mu, n) \rightarrow \infty$ for $n \rightarrow \infty$, but $(f_0(\mu, n)/\mathbb{E}f_0(\mu, n))_{n \in \mathbb{N}}$ does not converge to 1 in probability.

12.2 RANDOM POINTS – OTHER ASPECTS

For a finite set of points, the relative position of its elements may be viewed under various geometric and combinatorial aspects. For randomly generated point sets, the probabilities of particular configurations may be of interest, but are in general hard to obtain. We list some contributions to problems of this type.

For infinite sets of points in the whole space, the natural generalization of i.i.d. points in a compact domain are homogeneous Poisson processes.

12.2.1 GEOMETRIC CONFIGURATIONS

Bokowski *et al.* [BoRS92] made a simulation study to estimate the probabilities of certain order types, using the Grassmann approach.

Related to k -sets (see Chapter 1 of this Handbook) is the following investigation of Bárány and Steiger [BáS94]. If X is a set of n points in general position in \mathbb{R}^d , a subset $S \subset X$ of d points is called a k -simplex if X has exactly k points on one side of the affine hull of S . The authors study $E_d(k, n)$, the expected number of k -simplices for n i.i.d. random points. For continuous spherically symmetric distributions they show that

$$E_d(k, n) \leq c(d)n^{d-1}.$$

Further results concern the uniform distribution in a convex body in \mathbb{R}^2 .

For a given distribution μ on \mathbb{R}^2 , let $P_1, \dots, P_j, Q_1, \dots, Q_k$ be i.i.d. points distributed according to μ . Let $p_{jk}(\mu)$ be the probability that the convex hull of P_1, \dots, P_j is disjoint from the convex hull of Q_1, \dots, Q_k . Continuing earlier work of L.C.G. Rogers and of Buchta, Buchta and Reitzner [BuR97a] investigated $p_{jk}(\mu)$. For the uniform distribution μ in a convex domain K , they connected $p_{jk}(\mu)$ to equiaffine inner parallel curves of K , found an explicit representation in the case of polygons, and proved, among other results, that

$$\lim_{n \rightarrow \infty} \frac{p_{nn}(\mu)}{n^{3/2}4^{-n}} \geq \frac{8\sqrt{\pi}}{3},$$

with equality if K is centrally symmetric. The investigation was continued by Buchta and Reitzner in [BuR97b].

Various elementary geometric questions can be asked, even about a small number of random points. For example, if three uniform i.i.d. points in a convex body K are given, what is the probability that the triangle formed by them is obtuse, or what is the probability that the circle (almost surely) determined by these points is contained in K ? Known results on probabilities of these types are listed in [BaS95]. The following result is due to Affentranger. The probability that the sphere spanned (almost surely) by $d + 1$ i.i.d. uniform random points in a convex body K is entirely contained in K attains its maximum precisely if K is a ball. In [BaS95] it is shown that the probability that the circumball of $n \geq 2$ i.i.d. uniform points in K is contained in K is maximal if and only if K is a ball. The value of this maximum is $n/(2n - 1)$ if $d = 2$, but is unknown for $d > 2$.

Many special problems are treated, and references are given, in the book of Mathai [Mat99].

12.2.2 SHAPE

Two subsets of \mathbb{R}^d may be said to have the same shape if they differ only by a similarity. D.G. Kendall's theory of shape yields natural probability distributions on shapes of labeled n -tuples of points in \mathbb{R}^d . The possible shapes of such n -tuples of points (not all coincident) can canonically be put in one-to-one correspondence with points of a certain topological space, and the resulting "shape spaces" carry natural probability measures. For this extensive theory and its statistical applications, we refer to the survey given by Kendall [Ken89] and to the book of Kendall *et al.* [KeBCL99].

A different approach to more general notions of shape and probability distributions for them is followed by Ambartsumian [Amb90]. He uses factorization of products of invariant measures to obtain corresponding probability densities, for example, for the affine shape of a tetrad of points in the plane.

12.2.3 POINT PROCESSES

The investigations described so far concerned finite systems of random points. For randomly generated infinite discrete point sets, suitable models are provided by stochastic point processes.

GLOSSARY

Locally finite: $M \subset \mathbb{R}^d$ is locally finite if $\text{card}(M \cap B) < \infty$ for every compact set $B \subset \mathbb{R}^d$.

\mathcal{M} : The set of all locally finite subsets of \mathbb{R}^d .

\mathbf{M} : The smallest σ -algebra on \mathcal{M} for which every function $M \mapsto \text{card}(M \cap B)$ is measurable, where $B \subset \mathbb{R}^d$ is a Borel set.

(Simple) point process X on \mathbb{R}^d : A measurable map X from some probability space (Ω, \mathcal{A}, P) into $(\mathcal{M}, \mathbf{M})$.

Distribution of X : The image measure P_X of P under X .

Intensity measure Λ of X : $\Lambda(B) = \mathbb{E} \text{card}(X \cap B)$, for Borel sets $B \subset \mathbb{R}^d$.

Stationary (or **homogeneous**): X is a stationary point process if the distribution P_X is invariant under translations.

The point process X on \mathbb{R}^d , with intensity measure Λ (assumed to be finite on compact sets), is a **Poisson process** if, for any finitely many pairwise disjoint Borel sets B_1, \dots, B_k , the random variables $\text{card}(X \cap B_1), \dots, \text{card}(X \cap B_k)$ are independent and Poisson distributed. Thus, a Poisson point process X satisfies

$$\text{Prob}\{\text{card}(X \cap B) = k\} = e^{-\Lambda(B)} \frac{\Lambda(B)^k}{k!}$$

for $k \in \mathbb{N}_0$ and every Borel set B . If it is stationary, then the intensity measure Λ is γ times the Lebesgue measure, and the number γ is called the **intensity** of X . Let X be a stationary Poisson process and $C \subset \mathbb{R}^d$ a compact set, and let $k \in \mathbb{N}_0$. Under the condition that exactly k points of the process fall into C , these points are equivalent to k i.i.d. uniform points in C . This fact clearly illustrates the geometric significance of stationary Poisson point processes, as does the following. Consider n i.i.d. uniform points in the ball rB^d . The Poisson process with intensity measure equal to the Lebesgue measure can be considered as the limit process that is obtained if n and r tend to infinity in such a way that $n/V_d(rB^d) \rightarrow 1$.

A detailed study of geometric properties of stationary Poisson processes in the plane was made by Miles [Mil70].

For much of the theory of point processes, the underlying space \mathbb{R}^d can be replaced by a locally compact topological space Σ with a countable base. Of importance for stochastic geometry are, in particular, the cases where Σ is the space of r -flats in \mathbb{R}^d (see [Section 12.3.3](#)) or the space of convex bodies in \mathbb{R}^d .

12.3 RANDOM FLATS

Next to random points, randomly generated r -dimensional flats in \mathbb{R}^d are the objects of study in stochastic geometry that are particularly close to discrete geometry. Like convex hulls of random points, intersections of random halfspaces yield random polytopes in a natural way. Random flats through convex bodies as well as infinite arrangements of random hyperplanes give rise to a variety of questions.

12.3.1 RANDOM HYPERPLANES AND HALFSPACES

Intersections of random halfspaces appear as solution sets of systems of linear inequalities with random coefficients. Therefore, such random polyhedra play a role in the average case analysis of linear programming algorithms (see the book by Borgwardt [Bor87] and its bibliography). Under various assumptions on the distribution of the coefficients, one has information on the expected number of vertices of the solution sets. Extending earlier work of Prékopa, Buchta [Buc87a] obtained several estimates, of which the following is an example.

Let $E(v)$ be the expected number of vertices of the polyhedron given by the inequalities $\sum_{i=1}^n a_{ij} x_j \leq b$ ($i = 1, \dots, m$), $x_j \geq 0$ ($j = 1, \dots, n$). If the coefficients a_{ij} are nonnegative and distributed independently, continuously, and symmetrically with respect to the same number $c > 0$, then

$$E(v) = \frac{1}{2^{m-1}} \binom{n}{m} + \frac{m}{2^{m-1}} \binom{n}{m-1} + O(n^{m-2})$$

for $n \rightarrow \infty$. Buchta [Buc87b] also has formulas and estimates for $E(v)$ in the

case of the polyhedron given by $\sum_{j=1}^n a_{ij}x_j \leq 1$ ($i = 1, \dots, m$), where the points (a_{i1}, \dots, a_{in}) ($i = 1, \dots, m$) are i.i.d. uniform on the sphere S^{d-1} .

In a certain duality to convex hulls of random points in a convex body, one may consider intersections of halfspaces containing a convex body. Let $K \subset \mathbb{R}^d$ be a convex body with a boundary of class C^3 and with positive Gauss curvature κ ; suppose that $0 \in \text{int } K$ and let $r > 0$. Call a random closed halfspace $H_{u,t}^- := \{x \in \mathbb{R}^d \mid \langle x, u \rangle \leq t\}$ with $u \in S^{d-1}$ and $t > 0$ “ (K, r) -adapted” if the unit normal vector u is uniform on S^{d-1} and the distance t is independent of u and is, for given u , uniform in the interval for which $H_{u,t}^-$ contains K but not rB^d . Let $\mathbb{E}\tilde{V}_d(K, n)$ be the expected volume of the intersection of rB^d with n i.i.d. (K, r) -adapted random halfspaces. Then Kaltenbach [Kal90] proved that

$$\begin{aligned} \mathbb{E}\tilde{V}_d(K, n) - V_d(K) &= c_1(d) \int_{\partial K} \kappa^{1/(d+1)} dS \left(\frac{n}{V_1(rB^d) - V_1(K)} \right)^{-2/(d+1)} + \\ &\quad + O(n^{-3/(d+1)}) + O(r^d(1-\epsilon)^n) \end{aligned}$$

for $n \rightarrow \infty$, where $0 < \epsilon < 1$ is fixed.

Let X_1, \dots, X_n be i.i.d. random points on the boundary of a smooth convex body K . Let $K_{(n)}$ be the intersection of the supporting halfspaces of K at X_1, \dots, X_n (intersected with some fixed large cube, to make it bounded), and put $D_{(j)}(K, n) := V_j(K_{(n)}) - V_j(K)$. Under the assumption that the distribution of the X_i has a positive density h and that K and h are sufficiently smooth, Böröczky and Reitzner [BöR02] have obtained asymptotic expansions, as $n \rightarrow \infty$, for $\mathbb{E}D_{(j)}(K, n)$ in the cases $j = d$, $d-1$, and 1 .

Let $(X_i)_{i \in \mathbb{N}}$ be a sequence of i.i.d. random points on ∂K , and let $K_{(n)}$ and h be defined as above. If ∂K is of class C^2 and positive curvature and h is positive and continuous, one may ask whether $n^{2/(d-1)}D_{(j)}(K, n)$ converges almost surely to a positive constant, if $n \rightarrow \infty$. For $d = 2$ and $j = 1, 2$, this was shown by Schneider [Sch88]. Reitzner [Rei02b] was able to prove such a result for $d \geq 2$ and $j = d$. He deduced that random approximation, in this sense, is very close to best approximation.

12.3.2 RANDOM FLATS THROUGH CONVEX BODIES

The notion of a uniform random point in a convex body K in \mathbb{R}^d is extended by that of a uniform random r -flat through K . Let \mathcal{E}_r^d be the space of r -dimensional affine subspaces of \mathbb{R}^d with the usual topology and Borel structure ($r \in \{0, \dots, d-1\}$). A random r -flat is a measurable map from some probability space into \mathcal{E}_r^d . It is a ***uniform (isotropic uniform*** random r -flat through K if its distribution can be obtained from a translation invariant (resp. rigid-motion invariant) measure on \mathcal{E}_r^d , by restricting it to the r -flats meeting K and normalizing to a probability measure. (For details, see [WeW93, Section 2] and [ScW00, Chapter 4].)

A random r -flat E (uniform or not) through K generates the random secant $E \cap K$, which has often been studied, particularly for $r = 1$. References are in [ScW92, Chapter 6] and [ScWi93, Section 7]. Finitely many i.i.d. random flats through K lead to combinatorial questions. Associated random variables, such as the number of intersection points inside K if $d = 2$ and $r = 1$, are hard to attack; for work of Sulanke (1965) and Gates (1984) see [ScWi93]. Of special interest is the case of $i \leq d$ i.i.d. uniform hyperplanes H_1, \dots, H_i through a convex body $K \subset \mathbb{R}^d$.

Let $p_i(K)$ denote the probability that the intersection $H_1 \cap \dots \cap H_i$ also meets K . In some special cases, the maximum of this probability (which depends on K and on the distribution of the hyperplanes) is known, but not in general. References for this and related problems and a conjecture are found in [BaS95]. If $N > d$ i.i.d. uniform hyperplanes through K are given, they give rise to a random cell decomposition of $\text{int } K$. For $k \in \{0, \dots, d\}$, the expected number, $\mathbb{E}\nu_k$, of k -dimensional cells of this decomposition is given by

$$\mathbb{E}\nu_k = \sum_{i=d-k}^d \binom{i}{d-k} \binom{N}{i} p_i(K),$$

with $p_i(K)$ as defined above (Schneider [Sch82]). If the hyperplanes are isotropic uniform, then

$$\mathbb{E}\nu_k = \sum_{i=d-k}^d \binom{i}{d-k} \binom{N}{i} \frac{i! \kappa_i}{2^i} \frac{V_i(K)}{V_1^n(K)}.$$

OPEN PROBLEM

PROBLEM 12.3.1

For $i \leq d$ i.i.d. uniform random hyperplanes through a convex body K , find the sharp upper bound for the probability that their intersection also intersects K .

12.3.3 POISSON FLATS

A suitable model for infinite discrete random arrangements of r -flats in \mathbb{R}^d is provided by a point process in the space \mathcal{E}_r^d . Stationary Poisson processes are the simplest and geometrically most interesting examples (stationarity again means translation invariance of the distribution). Basic work was done by Miles [Mil71a] and Matheron [Math75]. In the case $r = d - 1$, one speaks of a **stationary Poisson hyperplane process**. For a hyperplane process, an i th intersection density Δ_i can be defined, in such a way that, for a Borel set $A \subset \mathbb{R}^d$, the expectation of the total i -dimensional volume inside A of the intersections of any $d - i$ hyperplanes of the process is given by Δ_i times the Lebesgue measure of A . Given the intensity Δ_{d-1} , the maximal i th intersection density Δ_i (for an $i \in \{0, \dots, d-2\}$) is achieved if the process is isotropic (its distribution is rigid-motion invariant); this result is due to Thomas (1984, see [ScW00, Section 4.5]). (His result and method were carried over to deterministic discrete hyperplane systems by Schneider [Sch95].) Similar questions can be asked for stationary Poisson r -flats with $r < d - 1$, for example for $2r \geq d$ and intersections of any two r -flats. Here nonisotropic extremal cases occur, such as in the case $r = 2$, $d = 4$ solved by Mecke [Mec88]. Various other cases have been treated; see Mecke [Mec91], Keutel [Keu91], and the references given there.

12.4 RANDOM CONGRUENT COPIES

The following is a typical question on randomly moving sets. Let $K_0, K \subset \mathbb{R}^d$ be given convex bodies. An isotropic random congruent copy of K meeting K_0 is of

the form gK , where g is a random element of the motion group G_d of \mathbb{R}^d , and the distribution of g is obtained from the Haar measure on G_d by restricting it to the set $\{g \in G_d \mid K_0 \cap gK \neq \emptyset\}$ and normalizing. Let K_1, \dots, K_n be convex bodies, let $g_i K_i$ be an isotropic random copy of K_i meeting K_0 , and suppose that g_1, \dots, g_n are stochastically independent. What is the probability that the random bodies $g_1 K_1, \dots, g_n K_n$ have a common point inside K_0 ? This question and similar ones can be given explicit answers by means of integral geometry. We refer to the books of Santaló [San76] and of Schneider and Weil [ScW92].

12.5 RANDOM MOSAICS

By a **tessellation** of \mathbb{R}^d , or a **mosaic** in \mathbb{R}^d , we understand a collection of d -dimensional polytopes such that their union is \mathbb{R}^d , the intersection of any two of the polytopes is either empty or a face of each of them, and any bounded set meets only finitely many of the polytopes. A **random mosaic** can be modeled by a point process in the space of convex polytopes, such that the properties above are satisfied almost surely. General references are [Møl89], [MeSSW90, Chapter 3], [WeW93, Section 7], [StKM95, Chapter 10], and [ScW00, Chapter 6].

NOTATION

X	stationary random mosaic in \mathbb{R}^d
$X^{(k)}$	process of its k -dimensional faces
$d_j^{(k)}$	density of the j th intrinsic volume of the polytopes in $X^{(k)}$
$\gamma^{(k)}$	$= d_0^{(k)}$, k -face intensity of X
$Z^{(k)}$	typical k -face of X
n_{jk}	expected number of elements of $X^{(k)}$ that are typically incident with a j -face of X

Under a natural assumption on the stationary random mosaic X , the notions of ‘density’ and ‘typical’ exist with a precise meaning. The density $d_j^{(k)}$ is then the intensity $\gamma^{(k)}$ times the expectation of the j th intrinsic volume of the typical k -face $Z^{(k)}$. Here, we can only convey the intuitive idea that one averages over expanding bounded regions of the mosaic and performs a limit procedure. Exact definitions can be found in [ScW00]; we refer also to Chapter 6 of that book for the results listed below and for all the related references.

12.5.1 GENERAL MOSAICS

For arbitrary stationary random mosaics, there are a number of identities relating averages of combinatorial quantities. Basic examples are:

$$\sum_{k=0}^j (-1)^k n_{jk} = 1, \quad \sum_{k=j}^d (-1)^{d-k} n_{jk} = 1, \quad \gamma^{(j)} n_{jk} = \gamma^{(k)} n_{kj},$$

$$\sum_{i=j}^d (-1)^i d_j^{(i)} = 0, \quad \text{and in particular} \quad \sum_{i=0}^d (-1)^i \gamma^{(i)} = 0.$$

If the random mosaic X is normal, meaning that every k -face is contained in exactly $d - k + 1$ d -polytopes of X ($k = 0, \dots, d - 1$), then

$$(1 - (-1)^k)\gamma^{(k)} = \sum_{j=0}^{k-1} (-1)^j \binom{d+1-j}{k-j} \gamma^{(j)}.$$

Lurking in the background are, of course, the polytopal relations of Euler, Dehn-Sommerville, and Gram; see [Chapters 16](#) and [18](#) of this Handbook.

12.5.2 HYPERPLANE TESSELLATIONS

A random mosaic X is called a stationary hyperplane tessellation if it is induced, in the obvious way, by a stationary hyperplane process (as defined in Section 12.3.3). Such random mosaics have special properties. Under an assumption of general position (satisfied, for example, by Poisson hyperplane processes) one has, for $0 \leq j \leq k \leq d$,

$$d_j^{(k)} = \binom{d-j}{d-k} d_j^{(j)}, \quad \text{in particular} \quad \gamma^{(k)} = \binom{d}{k} \gamma^{(0)},$$

and

$$n_{kj} = 2^{k-j} \binom{k}{j}.$$

A stationary Poisson hyperplane process, satisfying a suitable assumption of non-degeneracy, induces a stationary random mosaic X in general position, called a **Poisson hyperplane mosaic**. In the isotropic case (where the distributions are invariant under rigid motions), X is completely determined by the intensity, $\hat{\gamma}$, of the underlying Poisson hyperplane process. In this case, one has

$$d_j^{(k)} = \binom{d-j}{d-k} \binom{d}{j} \frac{\kappa_{d-1}^{d-j}}{d^{d-j} \kappa_d^{d-j-1} \kappa_j} \hat{\gamma}^{d-j},$$

in particular,

$$\gamma^{(k)} = \binom{d}{k} \frac{\kappa_{d-1}^d}{d^d \kappa_d^{d-1}} \hat{\gamma}^d, \quad \text{and} \quad \mathbb{E} V_j(Z^{(k)}) = \binom{k}{j} \left(\frac{d \kappa_d}{\kappa_{d-1}} \right)^j \frac{1}{\kappa_j \hat{\gamma}^j}.$$

The almost surely unique d -polytope of X containing 0 is called the Poisson zero-cell and denoted by Z_0 . For a stationary Poisson hyperplane mosaic, the inequalities

$$\mathbb{E} V_d(Z_0) \geq d! \kappa_d \left(\frac{2 \kappa_{d-1}}{d \kappa_d} \hat{\gamma} \right)^{-d}$$

and

$$2^d \leq \mathbb{E} f_0(Z_0) \leq 2^{-d} d! \kappa_d^2$$

are valid. In the isotropic case, equality holds in the first and on the right-hand side of the second inequality.

For the typical cell $Z^{(d)}$, the distribution of the inradius I (radius of the largest contained ball) can be determined; it is given by $\text{Prob}\{I(Z^{(d)}) \leq a\} = 1 - \exp(-2\hat{\gamma}a)$ for $a \geq 0$.

12.5.3 VORONOI AND DELAUNAY MOSAICS

A discrete point set in \mathbb{R}^d induces a Voronoi and a Delaunay mosaic (see [Chapter 23](#) for the definitions). Starting from a stationary Poisson point process \tilde{X} in \mathbb{R}^d , one obtains in this way a stationary **Poisson-Voronoi mosaic** and **Poisson-Delaunay mosaic**. Both of these are completely determined by the intensity, $\tilde{\gamma}$, of the underlying Poisson process \tilde{X} . For a Poisson-Voronoi mosaic and for $k \in \{0, \dots, d\}$, one has

$$d_k^{(k)} = \frac{2^{d-k+1} \pi^{\frac{d-k}{2}}}{d(d-k+1)!} \frac{\Gamma\left(\frac{d^2-kd+k+1}{2}\right) \Gamma\left(1 + \frac{d}{2}\right)^{d-k+\frac{k}{d}} \Gamma\left(d-k + \frac{k}{d}\right)}{\Gamma\left(\frac{d^2-kd+k}{2}\right) \Gamma\left(\frac{d+1}{2}\right)^{d-k} \Gamma\left(\frac{k+1}{2}\right)} \tilde{\gamma}^{\frac{d-k}{d}}.$$

In particular, the vertex density is given by

$$\gamma^{(0)} = \frac{2^{d+1} \pi^{\frac{d-1}{2}}}{d^2(d+1)} \frac{\Gamma\left(\frac{d^2+1}{2}\right)}{\Gamma\left(\frac{d^2}{2}\right)} \left[\frac{\Gamma\left(1 + \frac{d}{2}\right)}{\Gamma\left(\frac{d+1}{2}\right)} \right]^d \tilde{\gamma}.$$

For many other parameters, their explicit values in terms of $\tilde{\gamma}$ are known, especially in small dimensions.

For a Poisson-Delaunay mosaic one can, in a certain sense, explicitly determine the distribution of the typical d -cell and the moments of its volume.

12.6 SOURCES AND RELATED MATERIAL

SOURCES FOR STOCHASTIC GEOMETRY IN GENERAL

Stoyan, Kendall, and Mecke [StKM95]: A monograph on theoretical foundations and applications of stochastic geometry.

Matheron [Math75]: A monograph on basic models of stochastic geometry and applications of integral geometry.

Santaló [San76]: The classical work on integral geometry and its applications to geometric probabilities.

Schneider and Weil [ScW00]: An introduction to the mathematical models of stochastic geometry, with emphasis on the application of integral geometry and functionals from convexity.

Kendall and Moran [KeM63]: A collection of problems on geometric probabilities.

Solomon [Sol78]: A selection of topics from geometric probability theory.

Mathai [Mat99]: A comprehensive collection of results on geometric probabilities, in particular of those types where analytic calculations lead to explicit results.

Klain and Rota [KlR97]: An introduction to typical results of integral geometry, their interpretations in terms of geometric probabilities, and counterparts of discrete and combinatorial character.

Ambartzumian [Amb90]: Develops a special approach to stochastic geometry via factorization of measures, with various applications.

Moran [Mor66], [Mor69], Little [Lit74], Baddeley [Bad77]: “Notes on recent research in geometrical probability,” useful surveys with many references.

Baddeley [Bad82]: An introduction and reading list for stochastic geometry.

Baddeley [Bad84]: Connections of stochastic geometry with image analysis.

Weil and Wieacker [WeW93]: A comprehensive handbook article on stochastic geometry.

RELATED CHAPTERS

Several topics are outside the scope of this chapter, although they could be subsumed under probabilistic aspects of discrete geometry. Among these are randomization and average-case analysis of geometric algorithms and the probabilistic analysis of optimization problems in Euclidean spaces. Two classical topics of discrete geometry, namely packing and covering, were also excluded, for the reason that the existing probabilistic results are in a spirit rather far from discrete geometry.

Chapters of this Handbook in which these and related topics are covered are:

[Chapter 1: Finite point configurations](#)

[Chapter 2: Packing and covering](#)

[Chapter 16: Basic properties of convex polytopes](#)

[Chapter 18: Face numbers of polytopes and complexes](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 40: Randomization and derandomization](#)

[Chapter 46: Mathematical programming](#)

RELEVANT SURVEYS AND FURTHER SOURCES

Some of the topics treated have been the subjects of earlier surveys. The following sources contain references to the excluded topics as well as to work within the scope of this chapter.

Borgwardt [Bor87], [Bor99], Shamir [Sha93]: Information on the probabilistic analysis of linear programming algorithms under different model assumptions.

Dwyer [Dwy88] and later work: Contributions to the average-case analysis of geometric algorithms.

Hall [Hal88]: A monograph devoted to the probabilistic analysis of coverage problems.

Buchta [Buc85]: A survey on random polytopes.

Schneider [Sch88], Affentranger [Aff92]: Surveys on approximation of convex bodies by random polytopes.

Gruber [Gru97], Schütt [Schü02]: Surveys comparing best and random approximation of convex bodies by polytopes.

Bauer and Schneider [BaS95]: A collection of information on inequalities and extremum problems for geometric probabilities.

REFERENCES

- [Aff88] F. Affentranger. The expected volume of a random polytope in a ball. *J. Microscopy*, 151:277–287, 1988.
- [Aff91] F. Affentranger. The convex hull of random points with spherically symmetric distributions. *Rend. Sem. Mat. Univ. Politec. Torino*, 49:359–383, 1991.
- [Aff92] F. Affentranger. Aproximación aleatoria de cuerpos convexos. *Publ. Mat.*, 36:85–109, 1992.
- [AfS92] F. Affentranger and R. Schneider. Random projections of regular simplices. *Discrete Comput. Geom.*, 7:219–226, 1992.
- [AfW91] F. Affentranger and J.A. Wieacker. On the convex hull of uniform random points in a simple d -polytope. *Discrete Comput. Geom.*, 6:291–305, 1991.
- [AlFGP91] D.J. Aldous, B. Fristedt, P.S. Griffin, and W.E. Pruitt. The number of extreme points in the convex hull of a random sample. *J. Appl. Probab.*, 28:287–304, 1991.
- [Amb90] R.V. Ambartzumian. *Factorization Calculus and Geometric Probability*. Volume 33 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1990.
- [Bad77] A.J. Baddeley. A fourth note on recent research in geometrical probability. *Adv. in Appl. Probab.*, 9:824–860, 1977.
- [Bad82] A.J. Baddeley. Stochastic geometry: An introduction and reading list. *Internat. Statist. Rev.*, 50:179–193, 1982.
- [Bad84] A.J. Baddeley. Stochastic geometry and image analysis. *CWI Newslett.*, 4:2–20, 1984.
- [Bár89] I. Bárány. Intrinsic volumes and f -vectors of random polytopes. *Math. Ann.*, 285:671–699, 1989.
- [Bár92] I. Bárány. Random polytopes in smooth convex bodies. *Mathematika*, 39:81–92, 1992.
- [Bár99] I. Bárány. Sylvester's question: the probability that n points are in convex position. *Ann. Probab.*, 27:2020–2034, 1999.
- [BáB93] I. Bárány and C. Buchta. Random polytopes in a convex polytope, independence of shape, and concentration of vertices. *Math. Ann.*, 297:467–497, 1993.
- [BáF88] I. Bárány and Z. Füredi. On the shape of the convex hull of random points. *Probab. Theory Related Fields*, 77:231–240, 1988.
- [BáL88] I. Bárány and D. Larman. Convex bodies, economic cap coverings, random polytopes. *Mathematika*, 35:274–291, 1988.
- [BáS94] I. Bárány and W. Steiger. On the expected number of k -sets. *Discrete Comput. Geom.*, 11:243–263, 1994.
- [BáV93] I. Bárány and R.A. Vitale. Random convex hulls: floating bodies and expectations. *J. Approx. Theory*, 75:130–135, 1993.
- [Bar97] Y.M. Baryshnikov. Gaussian samples, regular simplices, and exchangeability. *Discrete Comput. Geom.*, 17:257–261, 1997.
- [BaV94] Y.M. Baryshnikov and R.A. Vitale. Regular simplices and Gaussian samples. *Discrete Comput. Geom.*, 11:141–147, 1994.
- [BaS95] C. Bauer and R. Schneider. Extremal problems for geometric probabilities involving convex bodies. *Adv. in Appl. Probab.*, 27:20–34, 1995.
- [BoRS92] J. Bokowski, J. Richter-Gebert, and W. Schindler. On the distribution of order types. *Comput. Geom. Theory Appl.*, 1:127–142, 1992.

- [Bor87] K.-H. Borgwardt. *The Simplex Method—a Probabilistic Approach*. Springer-Verlag, Berlin, 1987.
- [Bor99] K.-H. Borgwardt. A sharp upper bound for the expected number of shadow vertices in LP-polyhedra under orthogonal projection on two-dimensional planes. *Math. Oper. Res.*, 24:544–603. Erratum: 24:925–984, 1999.
- [BöH99] K. Böröczky, Jr. and M. Henk. Random projections of regular polytopes. *Arch. Math.*, 73:465–473, 1999.
- [BöR02] K. Böröczky, Jr. and M. Reitzner. Approximation of smooth convex bodies by random circumscribed polytopes. Preprint, 2002.
- [BH98] H. Bräker and T. Hsing. On the area and perimeter of a random convex hull in a bounded convex set. *Probab. Theory Related Fields*, 111:517–550, 1998.
- [BHB98] H. Bräker, T. Hsing and N.H. Bingham. On the Hausdorff distance between a convex set and an interior random convex hull. *Adv. in Appl. Probab.*, 30:295–316, 1998.
- [Buc84a] C. Buchta. Zufallspolygone in konvexen Vielecken. *J. Reine Angew. Math.*, 347:212–220, 1984.
- [Buc84b] C. Buchta. Das Volumen von Zufallspolyedern im Ellipsoid. *Anz. Österreich. Akad. Wiss. Math.-Natur. Kl.*, 121:1–4, 1984.
- [Buc84c] C. Buchta. Stochastische Approximation konvexer Polygone. *Z. Wahrscheinlichkeitstheorie Verw. Gebiete*, 67:283–304, 1984.
- [Buc85] C. Buchta. Zufällige Polyeder—Eine Übersicht. In: E. Hlawka, editor, *Zahlentheoretische Analysis*, volume 1114 of *Lecture Notes in Math.*, pages 1–13. Springer-Verlag, Berlin, 1985.
- [Buc87a] C. Buchta. On nonnegative solutions of random systems of linear inequalities. *Discrete Comput. Geom.*, 2:85–95, 1987.
- [Buc87b] C. Buchta. On the number of vertices of random polyhedra with a given number of facets. *SIAM J. Algebraic Discrete Methods*, 8:85–92, 1987.
- [Buc90] C. Buchta. Distribution-independent properties of the convex hull of random points. *J. Theoret. Probab.*, 3:387–393, 1990.
- [Buc02] C. Buchta. An identity relating moments of functionals of convex hulls. Preprint, 2002.
- [BuM84] C. Buchta and J. Müller. Random polytopes in a ball. *J. Appl. Probab.*, 21:753–762, 1984.
- [BuR92] C. Buchta and M. Reitzner. What is the expected volume of a tetrahedron whose vertices are chosen at random from a given tetrahedron? *Anz. Österreich. Akad. Wiss. Math.-Natur. Kl.*, 129:63–68, 1992.
- [BuR97a] C. Buchta and M. Reitzner. Equiaffine inner parallel curves of a plane convex body and the convex hulls of randomly chosen points. *Probab. Theory Related Fields*, 108:385–415, 1997.
- [BuR97b] C. Buchta and M. Reitzner. On a theorem of G. Herglotz about random polygons. *Rend. Circ. Mat. Palermo*, Ser. II, Suppl., 50:89–102, 1997.
- [BuR01] C. Buchta and M. Reitzner. The convex hull of random points in a tetrahedron: Solution of Blaschke’s problem and more general results. *J. Reine Angew. Math.*, 536:1–29, 2001.
- [CaG94] A.J. Cabo and P. Groeneboom. Limit theorems for functionals of convex hulls. *Probab. Theory Related Fields*, 100:31–55, 1994.

- [Chu93] D.P.T. Chu. Random r -content of an r -simplex from beta-type-2 random points. *Canad. J. Statist.*, 21:285–293, 1993.
- [Dev91] L. Devroye. On the oscillation of the expected number of extreme points of a random set. *Statist. Probab. Lett.*, 11:281–286, 1991.
- [DüW96] L. Dümbgen and G. Walther. Rates of convergence for random approximations of convex sets. *Adv. in Appl. Probab.*, 28:384–393, 1996.
- [Dwy88] R.A. Dwyer. *Average-Case Analysis of Algorithms for Convex Hulls and Voronoi Diagrams*. Ph.D. Thesis, Carnegie-Mellon Univ., Pittsburgh, 1988.
- [Dwy91] R.A. Dwyer. Convex hulls of samples from spherically symmetric distributions. *Discrete Appl. Math.*, 31:113–132, 1991.
- [Efr65] B. Efron. The convex hull of a random set of points. *Biometrika*, 52:331–343, 1965.
- [Gis96] S. Glasauer and R. Schneider. Asymptotic approximation of smooth convex bodies by polytopes. *Forum Math.*, 8:363–377, 1996.
- [Gro88] P. Groeneboom. Limit theorems for convex hulls. *Probab. Theory Related Fields*, 79:327–368, 1988.
- [Gru96] P.M. Gruber. Expectation of random polytopes. *Manuscripta Math.*, 91:393–419, 1996.
- [Gru97] P.M. Gruber. Comparisons of best and random approximation of convex bodies by polytopes. *Rend. Circ. Mat. Palermo*, Ser. II, Suppl., 50:189–216, 1997.
- [Hal88] P. Hall. *Introduction to the Theory of Coverage Processes*. Wiley, New York, 1988.
- [HaP03] M. Hartzoulaki and G. Paouris. Quermassintegrals of a random polytope in a convex body. *Arch. Math.*, 80:430–438, 2003.
- [Hsi94] T. Hsing. On the asymptotic distribution of the area outside a random convex hull in a disk. *Ann. Appl. Probab.*, 4:478–493, 1994.
- [Hue94] I. Hueter. The convex hull of a normal sample. *Adv. in Appl. Probab.*, 26:855–875, 1994.
- [Hue99] I. Hueter. Limit theorems for the convex hull of random points in higher dimensions. *Trans. Amer. Math. Soc.*, 351:4337–4363, 1999.
- [Kal90] F.J. Kaltenbach. *Asymptotisches Verhalten zufälliger konvexer Polyeder*. Dissertation, Univ. Freiburg i. Br., 1990.
- [Ken89] D.G. Kendall. A survey of the statistical theory of shape. *Statist. Sci.*, 4:87–120, 1989.
- [KeM63] M.G. Kendall and P.A.P. Moran. *Geometrical Probability*. Griffin, New York, 1963.
- [KeBCL99] D.G. Kendall, D. Barden, T.K. Carne, and H. Le. *Shape and Shape Theory*. Wiley, Chichester, 1999.
- [Keu91] J. Keutel. *Ein Extremalproblem für zufällige Ebenen und für Ebenenprozesse in höherdimensionalen Räumen*. Dissertation, Univ. Jena, 1991.
- [KlR97] D.A. Klain and G.-C. Rota. *Introduction to Geometric Probability*. Cambridge University Press, 1997.
- [Küf94] K.-H. Küfer. On the approximation of a ball by random polytopes. *Adv. in Appl. Probab.*, 26:876–892, 1994.
- [Lit74] D.V. Little. A third note on recent research in geometrical probability. *Adv. in Appl. Probab.*, 6:103–130, 1974.
- [Man94] D. Mannion. The volume of a tetrahedron whose vertices are chosen at random in the interior of a parent tetrahedron. *Adv. in Appl. Probab.*, 26:577–596, 1994.

- [Mas99] B. Massé. On the variance of the number of extreme points of a random convex hull. *Statist. Probab. Lett.*, 44:123–130, 1999.
- [Mas00] B. Massé. On the LLN for the number of vertices of a random convex hull. *Adv. in Appl. Probab.*, 32:675–681, 2000.
- [Mat99] A.M. Mathai. *An Introduction to Geometrical Probability: Distributional Aspects with Applications*. Gordon and Breach, Singapore, 1999.
- [Math75] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.
- [Mec88] J. Mecke. An extremal property of random flats. *J. Microscopy*, 151:205–209, 1988.
- [Mec91] J. Mecke. On the intersection density of flat processes. *Math. Nachr.*, 151:69–74, 1991.
- [MeSSW90] J. Mecke, R. Schneider, D. Stoyan, and W. Weil. *Stochastische Geometrie*. Volume 16 of *DMV Sem.*, Birkhäuser, Basel, 1990.
- [Mil70] R.E. Miles. On the homogeneous planar Poisson point process. *Math. Biosci.*, 6:85–127, 1970.
- [Mil71a] R.E. Miles. Poisson flats in Euclidean spaces. II: Homogeneous Poisson flats and the complementary theorem. *Adv. in Appl. Probab.*, 3:1–43, 1971.
- [Mil71b] R.E. Miles. Isotropic random simplices. *Adv. in Appl. Probab.*, 3:353–382, 1971.
- [Møl89] J. Møller. Random tessellations in \mathbb{R}^d . *Adv. in Appl. Probab.*, 21:37–73, 1989.
- [Mor66] P.A.P. Moran. A note on recent research in geometrical probability. *J. Appl. Probab.*, 3:453–463, 1966.
- [Mor69] P.A.P. Moran. A second note on recent research in geometrical probability. *Adv. in Appl. Probab.*, 1:73–89, 1969.
- [Rei01a] M. Reitzner. The floating body and the equiaffine inner parallel curve of a plane convex body. *Geom. Dedicata*, 84:151–167, 2001.
- [Rei01b] M. Reitzner. Stochastical approximation of smooth convex bodies. *Mathematika*, to appear.
- [Rei02a] M. Reitzner. Random points on the boundary of smooth convex bodies. *Trans. Amer. Math. Soc.*, 354:2243–2278, 2002.
- [Rei02b] M. Reitzner. Random polytopes are nearly best approximating. *Rend. Circ. Mat. Palermo*, Ser. II, Suppl. vol. II, 70:263–278, 2002.
- [RuM80] H. Ruben and R.E. Miles. A canonical decomposition of the probability measure of sets of isotropic random points in \mathbb{R}^n . *J. Multivariate Anal.*, 10:1–18, 1980.
- [San76] L.A. Santaló. *Integral Geometry and Geometric Probability*. Volume 1 of *Encyclopedia of Mathematics*, Addison-Wesley, Reading, 1976.
- [Sch82] R. Schneider. Random hyperplanes meeting a convex body. *Z. Wahrscheinlichkeitstheorie Verw. Gebiete*, 61:379–387, 1982.
- [Sch87] R. Schneider. Approximation of convex bodies by random polytopes. *Aequationes Math.*, 32:304–310, 1987.
- [Sch88] R. Schneider. Random approximation of convex sets. *J. Microscopy*, 151:211–227, 1988.
- [Sch95] R. Schneider. Isoperimetric inequalities for infinite hyperplane systems. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift. Discrete Comput. Geom.*, 13:609–627, 1995.
- [ScW92] R. Schneider and W. Weil. *Integralgeometrie*. Teubner, Stuttgart, 1992.
- [ScW00] R. Schneider and W. Weil. *Stochastische Geometrie*. Teubner, Stuttgart, 2000.

- [ScWi80] R. Schneider and J.A. Wieacker. Random polytopes in a convex body. *Z. Wahrscheinlichkeitstheorie Verwandte Gebiete*, 52:69–73, 1980.
- [ScWi93] R. Schneider and J.A. Wieacker. Integral geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 1349–1390. Elsevier, Amsterdam, 1993.
- [Schü94] C. Schütt. Random polytopes and affine surface area. *Math. Nachr.*, 170:227–249, 1994.
- [Schü02] C. Schütt. Best and random approximation of convex bodies by polytopes. *Rend. Circ. Mat. Palermo*, Ser. II, Suppl. vol. II, 70:315–334, 2002.
- [ScWe03] C. Schütt and E. Werner. Polytopes with vertices chosen randomly from the boundary of a convex body. In V. Milman and G. Schechtman, editors, *Israel Seminar 2001–2002*, volume 1807 of *Lecture Notes in Math.*, pages 241–422. Springer-Verlag, New York, 2003.
- [Sha93] R. Shamir. Probabilistic analysis in linear programming. *Statist. Sci.*, 8:57–64, 1993.
- [Sol78] H. Solomon. *Geometric Probability*. Soc. Industr. Appl. Math., Philadelphia, 1978.
- [StKM95] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and Its Applications*. 2nd ed., Wiley, Chichester, 1995.
- [Val95] P. Valtr. Probability that n random points are in convex position. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift. Discrete Comput. Geom.*, 13:637–643, 1995.
- [Val96] P. Valtr. The probability that n random points in a triangle are in convex position. *Combinatorica*, 16:567–573, 1996.
- [VeS92] A.M. Vershik and P.V. Sporyshev. Asymptotic behavior of the number of faces of random polyhedra and the neighborliness problem. *Selecta Math. Soviet.*, 11:181–201, 1992.
- [WaW01] U. Wagner and E. Welzl. A continuous analogue of the upper bound theorem. *Discrete Comput. Geom.*, 26:205–219, 2001.
- [WeW93] W. Weil and J.A. Wieacker. Stochastic geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 1391–1438. Elsevier, Amsterdam, 1993.
- [Wen62] J.G. Wendel. A problem in geometric probability. *Math. Scand.*, 11:109–111, 1962.

13 GEOMETRIC DISCREPANCY THEORY AND UNIFORM DISTRIBUTION

J. Ralph Alexander, J  zsef Beck, and William W.L. Chen

INTRODUCTION

A sequence s_1, s_2, \dots in $\mathbf{U} = [0, 1)$ is said to be *uniformly distributed* if, in the limit, the number of s_j falling in any given subinterval is proportional to its length. Equivalently, s_1, s_2, \dots is uniformly distributed if the sequence of equiweighted atomic probability measures $\mu_N(s_j) = 1/N$, supported by the initial N -segments s_1, s_2, \dots, s_N , converges weakly to Lebesgue measure on \mathbf{U} . This notion immediately generalizes to any topological space with a corresponding probability measure on the Borel sets.

Uniform distribution, as an area of study, originated from the remarkable paper of Weyl [Wey16], in which he established the fundamental result known nowadays as the Weyl criterion (see [Cas57, KN74]). This reduces a problem on uniform distribution to a study of related exponential sums, and provides a deeper understanding of certain aspects of Diophantine approximation, especially basic results such as Kronecker's density theorem. Indeed, careful analysis of the exponential sums that arise often leads to Erd  s-Tur  n-type upper bounds, which in turn lead to quantitative statements concerning uniform distribution.

Today, the concept of uniform distribution has important applications in a number of branches of mathematics such as number theory (especially Diophantine approximation), combinatorics, ergodic theory, discrete geometry, statistics, numerical analysis, etc. In this chapter, we focus on the geometric aspects of the theory.

13.1 UNIFORM DISTRIBUTION OF SEQUENCES

GLOSSARY

Uniformly distributed: Given a sequence $(s_n)_{n \in \mathbb{N}}$, with $s_n \in \mathbf{U} = [0, 1)$, let $Z_N([a, b)) = |\{j \leq N \mid s_j \in [a, b)\}|$. The sequence is uniformly distributed if, for every $0 \leq a < b \leq 1$, $\lim_{N \rightarrow \infty} N^{-1} Z_N([a, b)) = b - a$.

Fractional part: The fractional part $\{x\}$ of a real number x is $x - \lfloor x \rfloor$.

Kronecker sequence: A sequence of points of the form $(\{N\alpha_1\}, \dots, \{N\alpha_k\})_{N \in \mathbb{N}}$ in \mathbf{U}^k , where $1, \alpha_1, \dots, \alpha_k \in \mathbb{R}$ are linearly independent over \mathbb{Q} .

Discrepancy, or irregularity of distribution: The discrepancy of a sequence

$(s_n)_{n \in \mathbb{N}}$, with $s_n \in \mathbf{U} = [0, 1]$, in a subinterval $[a, b]$ of \mathbf{U} , is

$$\Delta_N([a, b]) = |Z_N([a, b]) - N(b - a)|.$$

More generally, the discrepancy of a sequence $(s_n)_{n \in \mathbb{N}}$, with $s_n \in \mathcal{S}$, a topological probability space, in a measurable subset $A \subset \mathcal{S}$, is $\Delta_N(A) = |Z_N(A) - N\mu(A)|$, where $Z_N(A) = |\{j \leq N \mid s_j \in A\}|$.

Aligned rectangle, aligned triangle: A rectangle (resp. triangle) in \mathbb{R}^2 two sides of which are parallel to the coordinate axes.

Hausdorff dimension: A set S in a metric space has Hausdorff dimension m , where $0 \leq m \leq +\infty$, if

- (i) for any $0 < k < m$, $\mu_k(S) > 0$;
- (ii) for any $m < k < +\infty$, $\mu_k(S) < +\infty$.

Here, μ_k is the k -dimensional **Hausdorff measure**, given by

$$\mu_k(S) = 2^{-k} \kappa_k \liminf_{\epsilon \rightarrow 0} \left\{ \sum_{i=1}^{\infty} (\text{diam } S_i)^k \mid S \subset \bigcup_{i=1}^{\infty} S_i, \text{diam } S_i \leq \epsilon \right\},$$

where κ_k is the volume of the unit ball in \mathbb{E}^k .

Remark. Throughout this chapter, the symbol c will always represent the generic absolute positive constant, depending only on the indicated parameters. The value generally varies from one appearance to the next.

It is not hard to prove that for any irrational number α , the sequence of fractional parts $\{N\alpha\}$ is everywhere dense in \mathbf{U} (here N is the running index). Suppose that the numbers $1, \alpha_1, \dots, \alpha_k$ are linearly independent over \mathbb{Q} . Then Kronecker's theorem states that the k -dimensional Kronecker sequence $(\{N\alpha_1\}, \dots, \{N\alpha_k\})$ is dense in the unit k -cube \mathbf{U}^k . It is a simple consequence of the Weyl criterion that any such Kronecker sequence is uniformly distributed in \mathbf{U}^k , a far stronger result than the density theorem. For example, letting $k = 1$, we see that $\{N\sqrt{2}\}$ is uniformly distributed in \mathbf{U} .

Weyl's work led naturally to the question: How rapidly can a sequence in \mathbf{U} become uniformly distributed as measured by the discrepancy $\Delta_N([a, b])$ of subintervals? Here, $\Delta_N([a, b]) = |Z_N([a, b]) - N(b - a)|$, where $Z_N([a, b])$ counts those $j \leq N$ for which s_j lies in $[a, b]$. Thus we see that Δ_N measures the difference between the actual number of s_j in an interval and the expected number. The sequence is uniformly distributed if and only if $\Delta_N(I) = o(N)$ for all subintervals I . The notion of discrepancy immediately extends to any topological probability space, provided there is at hand a suitable collection of measurable sets \mathcal{J} corresponding to the intervals. If A is in \mathcal{J} , set $\Delta_N(A) = |Z_N(A) - N\mu(A)|$.

From the works of Hardy, Littlewood, Ostrowski, and others, it became clear that the smaller the partial quotients in the continued fractions of the irrational number α are, the more uniformly distributed the sequence $\{N\alpha\}$ is. For instance, the partial quotients of quadratic irrationals are characterized by being cyclic, hence bounded. Studying the behavior of $\{N\alpha\}$ for these numbers has proved an excellent indicator of what might be optimal for general sequences in \mathbf{U} . Here one has $\Delta_N(I) < c(\alpha) \log N$ for all intervals I and integers $N \geq 2$. Unfortunately, one does not have anything corresponding to continued fractions in higher dimensions, and this has been an obstacle to a similar study of Kronecker sequences (see [Bec94]).

Van der Corput gave an alternative construction of a super uniformly distributed sequence of rationals in \mathbf{U} for which $\Delta_N(I) < c \log N$ for all intervals I and integers $N \geq 2$ (see [KN74, p. 127]). He also asked for the best possible estimate in this direction. In particular, he posed:

PROBLEM 13.1.1 *Van der Corput Problem* [vdC35a] [vdC35b]

Can there exist a sequence for which $\Delta_N(I) < c$ for all N and I ?

He conjectured, in a slightly different formulation, that such a sequence could not exist. This conjecture was affirmed by van Aardenne-Ehrenfest [vA-E45], who later showed that for any sequence in \mathbf{U} , $\sup_I \Delta_N(I) > c \log \log N / \log \log \log N$ for infinitely many values of N [vA-E49]. Her pioneering work gave the first nontrivial lower bound on the discrepancy of general sequences in \mathbf{U} . It is trivial to construct a sequence for which $\sup_I \Delta_N(I) \leq 1$ for infinitely many values of N .

In a classic paper, Roth showed that for any infinite sequence in \mathbf{U} , it must be true that $\sup_I \Delta_N(I) > c(\log N)^{1/2}$ for infinitely many N . Finally, in another classic paper, Schmidt used an entirely new method to prove the following result.

THEOREM 13.1.2 *Schmidt* [Sch72b]

The inequality $\sup_I \Delta_N(I) > c \log N$ holds for infinitely many N .

For a more detailed discussion of work arising from the van der Corput conjecture, see [BC87, pp. 3–6].

In light of van der Corput's sequence, as well as $\{N\sqrt{2}\}$, Schmidt's result is best possible. The following problem, which has been described as “excruciatingly difficult,” is a major remaining open question from the classical theory.

PROBLEM 13.1.3

Extend Schmidt's result to a best possible estimate of the discrepancy for sequences in \mathbf{U}^k for $k > 1$.

For a given sequence, the results above do not imply the existence of a fixed interval I in \mathbf{U} for which $\sup_N \Delta_N(I) = \infty$. Let I_α denote the interval $[0, \alpha)$, where $0 < \alpha \leq 1$. Schmidt [Sch72a] showed that for any fixed sequence in \mathbf{U} there are only countably many values of α for which $\Delta_N(I_\alpha)$ is bounded. The best result in this direction is due to Halász.

THEOREM 13.1.4 *Halász* [Hal81]

For any fixed sequence in \mathbf{U} , let A denote the set of values of α for which $\Delta_N(I_\alpha) = o(\log N)$. Then A has Hausdorff dimension 0.

For a more detailed discussion of work arising from this question, see [BC87, pp. 10–11].

The fundamental works of Roth and Schmidt opened the door to the study of discrepancy in higher dimensions, and there were surprises. In his classic paper, Roth [Rot54] transformed the heart of van der Corput's problem to a question concerning the unit square \mathbf{U}^2 . In this new formulation, Schmidt's “ $\log N$ theorem” implies that if N points are placed in \mathbf{U}^2 , there is always an aligned rectangle $I = [\gamma_1, \alpha_1) \times [\gamma_2, \alpha_2)$ having discrepancy exceeding $c \log N$. Roth also showed that it was possible to place N points in the square \mathbf{U}^2 so that the discrepancy of no aligned rectangle exceeds $c \log N$. One way is to choose $p_j = ((j-1)/N, \{j\sqrt{2}\})$ for $j \leq N$. Thus, the function $c \log N$ describes the *minimax discrepancy* for aligned

rectangles. However, Schmidt showed that there is always an aligned right triangle (the part of an aligned rectangle above, or below, a diagonal) with discrepancy exceeding $cN^{1/4-\epsilon}$! Later work has shown that $cN^{1/4}$ exactly describes the minimax discrepancy of aligned right triangles. This paradoxical behavior is not isolated.

Generally, if one studies a collection \mathcal{J} of “nice” sets such as disks, aligned boxes, rotated cubes, etc., in \mathbf{U}^k or some other convex region, it turns out that the minimax discrepancy is either bounded above by $c(\log N)^r$ or bounded below by cN^s , with nothing halfway. In \mathbf{U}^k , typically $s = (k-1)/2k$. Thus, there tends to be a logarithmic version of the Vapnik-Chervonenkis principle in operation (see [Chapter 36](#) of this Handbook for a related discussion). Later, we shall see how certain geometric properties place \mathcal{J} in one or the other of these two classes.

13.2 THE GENERAL FREE PLACEMENT PROBLEM FOR N POINTS

One can ask for bounds on the discrepancy of N variable points $\mathcal{P} = \{p_1, p_2, \dots, p_N\}$ that are freely placed in a domain \mathbf{K} in Euclidean t -space \mathbb{E}^t . By contrast, when one considers the discrepancy of a *sequence* in \mathbf{K} , the initial n -segment of $p_1, \dots, p_n, \dots, p_N$ remains fixed for $n \leq N$ as new points appear with increasing N . For a given \mathbf{K} , as the unit interval \mathbf{U} demonstrates, estimates for these two problems are quite different as functions of N . The freely placed points in \mathbf{U} need never have discrepancy exceeding 1.

With Roth’s reformulation (discussed in Section 13.1), the classical problem is easier to state and, more importantly, it generalizes in a natural manner to a wide class of problems. The bulk of geometric discrepancy problems are now posed as free placement problems. In practically all situations, the domain \mathbf{K} has a very simple description as a cube, disk, sphere, etc., and standard notation is used in the specific situations.

PROBABILITY MEASURES AND DISCREPANCY

In a free placement problem there are two probability measures in play. First, there is the atomic measure μ^+ that assigns weight $1/N$ to each p_j . Second, there is a probability measure μ^- on the Borel sets of \mathbf{K} . The measure μ^- is generally the restriction of a natural uniform measure, such as scaled Lebesgue measure. An example would be given by $\mu^- = \sigma/4\pi$ on the unit sphere \mathbf{S}^2 , where σ is the usual surface measure. It is convenient to define the signed measure $\mu = \mu^+ - \mu^-$ (in the previous section μ^- was denoted by μ). The discrepancy of a Borel set A is, as before, given by $\Delta(A) = |Z(A) - N\mu^-(A)| = N|\mu(A)|$.

The function Δ is always restricted to a very special collection \mathcal{J} of sets, and the challenge lies in obtaining estimates concerning the restricted Δ . It is the central importance of the collection \mathcal{J} that gives the study of discrepancy its distinct character. In a given problem it is sometimes possible to reduce the size of \mathcal{J} . Taking the unit interval \mathbf{U} as an example, letting \mathcal{J} be the collection of intervals $[\gamma, \alpha)$ seems to be the obvious choice. But a moment’s reflection shows that only intervals of the form $I_\alpha = [0, \alpha)$ need be considered for estimates of discrepancy. At most a factor of 2 is introduced in any estimate of bounds.

NOTIONS OF DISCREPANCY

In most interesting problems \mathcal{J} itself carries a measure ν in the sense of integral geometry, and this adds much more structure. While there is artistic latitude in the choice of ν , more often than not there is a natural measure on \mathcal{J} . In the example of \mathbf{U} , by identifying $I_\alpha = [0, \alpha)$ with its right endpoint, it is clear that Lebesgue measure on \mathbf{U} is the natural choice for ν .

Given that the measure ν exists, for $1 \leq W < \infty$ define

$$\|\Delta(\mathcal{P}, \mathcal{J})\|_W = \left(\int_{\mathcal{J}} (\Delta(A))^W d\nu \right)^{1/W} \quad \text{and} \quad \|\Delta(\mathcal{P}, \mathcal{J})\|_\infty = \sup_{\mathcal{J}} \Delta(A),$$

and for $1 \leq W \leq \infty$ define

$$D(\mathbf{K}, \mathcal{J}, W, N) = \inf_{|\mathcal{P}|=N} \{\|\Delta(\mathcal{P}, \mathcal{J})\|_W\}.$$

The determination of the “minimax” $D(\mathbf{K}, \mathcal{J}, \infty, N)$ is generally the most important as well as the most difficult problem in the study. It should be noted that the function $D(\mathbf{K}, \mathcal{J}, \infty, N)$ is defined even if the measure ν is not. The term $D(\mathbf{K}, \mathcal{J}, 2, N)$ has been shown to be intimately related to problems in numerical integration in some special cases, and is of increasing importance. These various functions $D(\mathbf{K}, \mathcal{J}, W, N)$ measure how well the continuous distribution μ^- can be approximated by N freely placed atoms.

The inequality

$$\nu(\mathcal{J})^{-1/W} \|\Delta(\mathcal{P}, \mathcal{J})\|_W \leq \|\Delta(\mathcal{P}, \mathcal{J})\|_\infty \tag{13.2.1}$$

provides a general approach for obtaining a lower bound for $D(\mathbf{K}, \mathcal{J}, \infty, N)$. The choice $W = 2$ has been especially fruitful, but good estimates of $D(\mathbf{K}, \mathcal{J}, W, N)$ for any W are of independent interest.

An upper bound on $D(\mathbf{K}, \mathcal{J}, \infty, N)$ generally is obtained by showing the existence of a favorable example. This may be done either by a direct construction, often extremely difficult to verify, or by a probabilistic argument showing such an example does exist without giving it explicitly. These comments would apply as well to upper bounds for any $D(\mathbf{K}, \mathcal{J}, W, N)$.

13.3 ALIGNED RECTANGLES IN THE UNIT SQUARE

The unit square $\mathbf{U}^2 = [0, 1] \times [0, 1]$ is by far the most thoroughly studied 2-dimensional object. The main reason for this is Roth’s reformulation of the van der Corput problem. Many of the interesting questions that arose have been answered, and we give a summary of the highlights.

For \mathbf{U}^2 one wishes to study the discrepancy of rectangles of the type $I = [\gamma_1, \alpha_1) \times [\gamma_2, \alpha_2)$. It is a trivial observation that only those I for which $\gamma_1 = \gamma_2 = 0$ need be considered, and this restricted family, denoted by \mathcal{B}^2 , is the choice for \mathcal{J} . By considering this smaller collection one introduces at most a factor of 4 on bounds. There is a natural measure ν on \mathcal{B}^2 , which may be identified with Lebesgue measure on \mathbf{U}^2 via the upper right corner points (α_1, α_2) . In the same spirit, let \mathcal{B}^1 denote the previously introduced collection of intervals $I_\alpha = [0, \alpha)$ in \mathbf{U} .

THEOREM 13.3.1 *Roth's Equivalence* [Rot54] [BC87, pp. 6–7]

Let f be a positive increasing function tending to infinity. Then the following two statements are equivalent:

- (i) There is an absolute positive constant c_1 such that for any finite sequence s_1, s_2, \dots, s_N in \mathbf{U} , there always exists a positive integer $n \leq N$ such that $\|\Delta(\mathcal{P}_n, \mathcal{B}^1)\|_\infty > c_1 f(N)$. Here, \mathcal{P}_n is the initial n -segment.
- (ii) There is an absolute positive constant c_2 such that for all positive integers N , $D(\mathbf{U}^2, \mathcal{B}^2, \infty, N) > c_2 f(N)$.

The equivalence shows that the central question of bounds for the van der Corput problem can be replaced by an elegant problem concerning the free placement of N points in the unit square \mathbf{U}^2 . The mapping $s_j \rightarrow ((j-1)/N, s_j)$ plays a role in the proof of this equivalence. If one takes as \mathcal{P}_N the image in \mathbf{U}^2 under the mapping of the initial N -segment of the van der Corput sequence, the following upper bound theorem may be proved.

THEOREM 13.3.2 *Lerch* [BC87, Theorem 4, $K = 2$]

For $N \geq 2$,

$$D(\mathbf{U}^2, \mathcal{B}^2, \infty, N) < c \log N. \quad (13.3.1)$$

The corresponding lower bound is established by the important “ $\log N$ theorem” of Schmidt.

THEOREM 13.3.3 *Schmidt* [Sch72b] [BC87, Theorem 3B]

One has

$$D(\mathbf{U}^2, \mathcal{B}^2, \infty, N) > c \log N. \quad (13.3.2)$$

By an explicit lattice construction, Davenport [Dav56] gave the best possible upper bound estimate for $W = 2$. His analysis shows that if the irrational number α has continued fractions with bounded partial quotients, then the $N = 2M$ points in \mathbf{U}^2 given by

$$p_j^\pm = ((j-1)/M, \{\pm j\alpha\}), \quad j \leq M,$$

can be taken as \mathcal{P} in proving the following theorem. Other proofs have been given by Vilenkin [Vil67], Halton and Zaremba [HZ69], and Roth [Rot76].

THEOREM 13.3.4 *Davenport* [Dav56] [BC87, Theorem 2A]

For $N \geq 2$,

$$D(\mathbf{U}^2, \mathcal{B}^2, 2, N) < c(\log N)^{1/2}. \quad (13.3.3)$$

This complements the following lower bound obtained by Roth in his classic paper.

THEOREM 13.3.5 *Roth* [Rot54] [BC87, Theorem 1A, $K = 2$]

One has

$$D(\mathbf{U}^2, \mathcal{B}^2, 2, N) > c(\log N)^{1/2}. \quad (13.3.4)$$

For $W = 1$, an upper bound $D(\mathbf{U}^2, \mathcal{B}^2, 1, N) < c(\log N)^{1/2}$ follows at once from Davenport's bound (13.3.3) by the monotonicity of $D(\mathbf{U}^2, \mathcal{B}^2, W, N)$ as a function of W . The corresponding lower bound was obtained by Halász more recently.

THEOREM 13.3.6 *Halász* [Hal81] [BC87, Theorem 1C, $K = 2$]

One has

$$D(\mathbf{U}^2, \mathcal{B}^2, 1, N) > c(\log N)^{1/2}. \quad (13.3.5)$$

Halász (see [BC87, Theorem 3C]) deduced that there is always an aligned square of discrepancy larger than $c \log N$. Of course, the square generally will not be a member of the special collection \mathcal{B}^2 . Ruzsa [Ruz93] has given a clever elementary proof that the existence of such a square follows directly from inequality (13.3.2) above.

The ideas developed in the study of discrepancy can be applied to approximations of integrals. We briefly mention two examples, both restricted to 2 dimensions for the sake of simplicity.

A function ψ is termed **M -simple** if $\psi(x) = \sum_{j=1}^M m_j \chi_{B_j}(x)$, where χ_{B_j} is the characteristic function of the aligned rectangle B_j . In this theorem, the lower bounds are nontrivial because of the logarithmic factors coming from discrepancy theory on \mathbf{U}^2 .

THEOREM 13.3.7 *Chen* [Che85] [Che87] [BC87, Theorems 5A, 5C]

Let the function f be defined on \mathbf{U}^2 by $f(x) = C + \int_{B(x)} g(y) dy$ where C is a constant, g is nonzero on a set of positive measure in \mathbf{U}^2 , and $B(x_1, x_2) = [0, x_1] \times [0, x_2]$. Then, for any M -simple function ψ ,

$$\begin{aligned} \|f - \psi\|_W &> c(f, W) M^{-1} (\log M)^{1/2}, & 1 \leq W < \infty; \\ \|f - \psi\|_\infty &> c(f) M^{-1} \log M. \end{aligned}$$

Let \mathcal{C} be the class of all continuous real valued functions on \mathbf{U}^2 , endowed with the Wiener sheet measure ω . For every function $f \in \mathcal{C}$ and every set \mathcal{P} of N points in \mathbf{U}^2 , let

$$I(f) = \int_{\mathbf{U}^2} f(x) dx \quad \text{and} \quad U(\mathcal{P}, f) = \frac{1}{N} \sum_{p \in \mathcal{P}} f(p).$$

THEOREM 13.3.8 *Woźniakowski* [Woź91]

One has

$$\inf_{|\mathcal{P}|=N} \left(\int_{\mathcal{C}} |U(\mathcal{P}, f) - I(f)|^2 d\omega \right)^{1/2} = \frac{D(\mathbf{U}^2, \mathcal{B}^2, 2, N)}{N}.$$

13.4 ALIGNED BOXES IN A UNIT k -CUBE

The van der Corput problem led to the study of $D(\mathbf{U}^2, \mathcal{B}^2, W, N)$, which in turn led to the study of $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ for general positive integers k and real $W \geq 1$. Here, \mathcal{B}^k denotes the collection of boxes $I = [0, \alpha_1] \times \dots \times [0, \alpha_k]$, and the measure ν is identified with Lebesgue measure on \mathbf{U}^k via the corner points $(\alpha_1, \dots, \alpha_k)$.

The principle of Roth's equivalence extends so that the discrepancy problem for sequences in \mathbf{U}^k reformulates as a free placement problem in \mathbf{U}^{k+1} , so that we discuss only the latter version. Inequalities (13.3.1)–(13.3.5) give the exact order of magnitude of $D(\mathbf{U}^2, \mathcal{B}^2, W, N)$ for the most natural values of W , namely

$1 \leq W \leq 2$ and $W = \infty$, with the latter being top prize. While much is known, knowledge of $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ is incomplete, especially for $W = \infty$, while there is ongoing work on the case $W = 1$ which may lead to its complete solution. It should be remarked that if k and N are fixed, then $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ is a nondecreasing function of W for $1 \leq W \leq \infty$.

As was indicated earlier, upper bound methods generally fall into two classes, explicit constructions and probabilistic existence arguments. In practice, careful constructions are made prior to a probabilistic averaging process. Chen's proof of the following upper bound theorem involved extensive combinatorial and number-theoretic constructions as well as probabilistic considerations.

THEOREM 13.4.1 *Chen* [Che80] [BC87, Theorem 2D]

For W satisfying $1 \leq W < \infty$, and integers $k \geq 2$ and $N \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, W, N) < c(W, k)(\log N)^{(k-1)/2}. \quad (13.4.1)$$

A second proof was given by Chen [Che83] (see also [BC87, Section 3.5]). Earlier, Roth [Rot80] (see also [BC87, Theorem 2C]) treated the case $W = 2$. The inequality (13.4.1) highlights one of the truly baffling aspects of the theory, namely the apparent jump discontinuity in the asymptotic behavior of $D(\mathbf{U}^k, \mathcal{B}^k, W, N)$ at $W = \infty$. This discontinuity is most dramatically established for $k = 2$, but is known to occur for any $k \geq 3$ (see (13.4.3) below).

Explicit multidimensional sequences greatly generalizing the van der Corput sequence also have been used to obtain upper bounds for $D(\mathbf{U}^k, \mathcal{B}^k, \infty, N)$. Halton constructed explicit point sets in \mathbf{U}^k in order to prove the next theorem. Faure (see [BC87, Section 3.2]) gave a different proof of the same result. If $k = 2$ is a guide, Halton's result may in fact be the best possible.

THEOREM 13.4.2 *Halton* [Hal60] [BC87, Theorem 4]

For integers $k \geq 2$ and $N \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, \infty, N) < c(k)(\log N)^{k-1}. \quad (13.4.2)$$

In order to prove (13.3.3), Davenport used properties of special lattices; but only very recently has there been further success with lattices in higher dimensions. Skriganov has established some most interesting results, which imply the following theorem. Given a region, a lattice is termed **admissible** if the region contains no member of the lattice except possibly the origin (see [Cas59]). Examples for the following theorem are given by lattices arising from algebraic integers in totally real algebraic number fields.

THEOREM 13.4.3 *Skriganov* [Skr94]

Suppose Γ is a fixed k -dimensional lattice admissible for the region $|x_1 x_2 \dots x_k| < 1$.

- (i) Halton's upper bound inequality (13.4.2) holds if the N points are obtained by intersecting \mathbf{U}^k with $t\Gamma$, where $t > 0$ is a suitably chosen real scalar.
- (ii) With the same choice of t as in part (i), there exists $x \in \mathbb{E}^k$ such that Chen's upper bound inequality (13.4.1) holds if the N points are obtained by intersecting \mathbf{U}^k with $t\Gamma + x$.

Recently, using p -adic Fourier-Walsh analysis together with ideas originating from coding theory, Chen and Skriganov [CS02] have obtained explicit constructions that give (13.4.1) in the special case $W = 2$, with an explicitly given constant $c(2, k)$.

Moving to lower bound estimates, the following theorem of Schmidt is complemented by Chen's result (13.4.1). For $W \geq 2$ this lower bound is due to Roth, since D is monotone in W .

THEOREM 13.4.4 *Schmidt* [Sch77a] [BC87, Theorem 1B]

For $W > 1$ and integers $k \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, W, N) > c(W, k)(\log N)^{(k-1)/2}.$$

Concerning $W = 1$, there is the result of Halász, which is probably not optimal. It is reasonably conjectured that $(k-1)/2$ is the correct exponent. There is ongoing work which may lead to its complete solution.

THEOREM 13.4.5 *Halász* [Hal81] [BC87, Theorem 1C]

For integers $k \geq 2$,

$$D(\mathbf{U}^k, \mathcal{B}^k, 1, N) > c(k)(\log N)^{1/2}.$$

The next lower bound estimate belongs to Baker. Although probably not best possible, it firmly establishes a discontinuity in asymptotic behavior at $W = \infty$ for all $k \geq 3$.

THEOREM 13.4.6 *Baker* [Bak99]

For integers $k \geq 3$ and $N > 20$,

$$D(\mathbf{U}^k, \mathcal{B}^k, \infty, N) > c(k)(\log N)^{(k-1)/2}(\log \log N)^{c_k}. \quad (13.4.3)$$

In fact, the exponent c_3 can be taken to be any positive real number less than $1/4$. Earlier, Beck [Bec89] had established a slightly weaker lower bound for the case $k = 3$, where c_3 can be taken to be any positive real number less than $1/8$. The work of Beck and Baker represents the first improvement of Roth's lower bound

$$D(\mathbf{U}^k, \mathcal{B}^k, \infty, N) > c(k)(\log N)^{(k-1)/2},$$

established over 30 years ago.

Can the factor $1/2$ be removed from the exponent? This is the “great open problem.” However, Beck has refined Roth's estimate in a geometric direction.

THEOREM 13.4.7 *Beck* [BC87, Theorem 19A]

Let \mathcal{J} be the collection of aligned cubes contained in \mathbf{U}^k . Then

$$D(\mathbf{U}^k, \mathcal{J}, \infty, N) > c(k)(\log N)^{(k-1)/2}. \quad (13.4.4)$$

Actually, Beck's method shows $D(\mathbf{U}^k, \mathcal{J}, 2, N) > c(k)(\log N)^{(k-1)/2}$, with respect to a natural measure ν on sets of aligned cubes. This improves Roth's inequality $D(\mathbf{U}^k, \mathcal{B}^k, 2, N) > c(k)(\log N)^{(k-1)/2}$. So far, it has not been possible to extend Ruzsa's ideas to higher dimensions in order to show that the previous theorem follows directly from Roth's estimate. However, more recently, Drmota [Drm96] has published a new proof that $D(\mathbf{U}^k, \mathcal{J}, 2, N) > c(k)D(\mathbf{U}^k, \mathcal{B}^k, 2, N)$, and this does imply (13.4.4).

13.5 MOTION-INVARIANT PROBLEMS

In this section and the next three, we discuss collections \mathcal{J} of convex sets having the property that any set in \mathcal{J} may be moved by a direct (orientation preserving) motion of \mathbb{E}^k and yet remain in \mathcal{J} . Motion-invariant problems were first extensively studied by Schmidt, and many of his estimates, obtained by a difficult technique using integral equations, were close to best possible. The book [BC87] contains an account of Schmidt's methods. But more recently, the Fourier transform method of Beck has achieved results that in general surpass those obtained by Schmidt. For a broad class of problems, Beck's Fourier method gives nearly best possible estimates for $D(\mathbf{K}, \mathcal{J}, 2, N)$.

The pleasant surprise is that if \mathcal{J} is motion-invariant, then the bounds on $D(\mathbf{K}, \mathcal{J}, \infty, N)$ turn out to be very close to those for $D(\mathbf{K}, \mathcal{J}, 2, N)$. This is shown by a probabilistic upper bound method, which generally pins $D(\mathbf{K}, \mathcal{J}, \infty, N)$ between bounds differing at most by a factor of $c(k)(\log N)^{1/2}$.

The simplest motion-invariant example is given by letting \mathcal{J} be the collection of all directly congruent copies of a given convex set A . In this situation, \mathcal{J} carries a natural measure ν , which may be identified with Haar measure on the motion group on \mathbb{E}^k . A broader choice would be to let \mathcal{J} be all sets in \mathbb{E}^k directly similar to A . Again, there is a natural measure ν on \mathcal{J} . However, for the results stated in the next two sections, the various measures ν on the choices for \mathcal{J} will not be discussed in great detail. In most situations, such measures do play an active role in the proofs through inequality (13.2.1) with $W = 2$. A complete exposition of integration in the context of integral geometry, Haar measure, etc., may be found in the book by Santaló [San76].

For any domain \mathbf{K} in \mathbb{E}^t and each collection \mathcal{J} , it is helpful to define three auxiliary collections:

Definition:

- (i) \mathcal{J}_{tor} consists of those subsets of \mathbf{K} obtained by reducing elements of \mathcal{J} modulo \mathbb{Z}^k . To avoid messiness, let us always suppose that \mathcal{J} has been restricted so that this reduction is 1–1 on each member of \mathcal{J} . For example, one might consider only those members of \mathcal{J} having diameter less than 1.
- (ii) \mathcal{J}_c consists of those subsets of \mathbf{K} that are members of \mathcal{J} .
- (iii) \mathcal{J}_i consists of those subsets of \mathbf{K} obtained by intersecting \mathbf{K} with members of \mathcal{J} .

Note that \mathcal{J}_c and \mathcal{J}_i are well defined for any domain \mathbf{K} . However, \mathcal{J}_{tor} essentially applies only to \mathbf{U}^k . If viewed as a flat torus, then \mathbf{U}^k is the proper domain for Kronecker sequences and Weyl's exponential sums. There are several general inequalities for discrepancy results involving \mathcal{J}_{tor} , \mathcal{J}_c , and \mathcal{J}_i . For example, we have $D(\mathbf{U}^k, \mathcal{J}_c, \infty, N) \leq D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N)$ because \mathcal{J}_c is contained in \mathcal{J}_{tor} . Also, if the members of \mathcal{J} have diameters less than 1, then we have $D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N) \leq 2^k D(\mathbf{U}^k, \mathcal{J}_i, \infty, N)$, since any set in \mathcal{J}_{tor} is the union of at most 2^k sets in \mathcal{J}_i .

13.6 SIMILAR OBJECTS IN THE UNIT k -CUBE

GLOSSARY

If A is a compact convex set in \mathbb{E}^k , let $d(A)$ denote the diameter of A , $r(A)$ denote the radius of the largest k -ball contained in A , and $\sigma(\partial A)$ denote the surface content of ∂A . The collection \mathcal{J} is said to be ***ds-generated*** by A if \mathcal{J} consists of all directly similar images of A having diameters not exceeding $d(A)$.

We state two pivotal theorems of Beck. As usual, if \mathcal{S} is a discrete set, $Z(B)$ denotes the cardinality of $B \cap \mathcal{S}$.

THEOREM 13.6.1 *Beck [Bec87] [BC87, Theorem 17A]*

Let \mathcal{S} be an arbitrary infinite discrete set in \mathbb{E}^k , A be a compact convex set with $r(A) \geq 1$, and \mathcal{J} be ds-generated by A . Then there is a set B in \mathcal{J} such that

$$|Z(B) - \text{vol } B| > c(k)(\sigma(\partial A))^{1/2}. \quad (13.6.1)$$

COROLLARY 13.6.2 *Beck [BC87, Corollary 17B]*

Let A be a compact convex body in \mathbb{E}^k with $r(A) \geq N^{-1/k}$, and let \mathcal{J} be ds-generated by A . Then

$$D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N) > c(A)N^{(k-1)/2k}. \quad (13.6.2)$$

The deduction of Corollary 13.6.2 from Theorem 13.6.1 involves a simple rescaling argument. Another important aspect of Beck's work is the introduction of upper bound methods based on probabilistic considerations. The following result shows that Theorem 13.6.1 is very nearly best possible.

THEOREM 13.6.3 *Beck [BC87, Theorem 18A]*

Let A be a compact convex body in \mathbb{E}^k with $r(A) \geq 1$, and let \mathcal{J} be ds-generated by A . Then there exists an infinite discrete set \mathcal{S}_0 such that for every set B in \mathcal{J} ,

$$|Z(B) - \text{vol } B| < c(k)(\sigma(\partial A))^{1/2}(\log \sigma(\partial A))^{1/2}. \quad (13.6.3)$$

COROLLARY 13.6.4 *Beck [BC87, Corollary 18C]*

Let A be a compact convex body in \mathbb{E}^k , and \mathcal{J} be ds-generated by A . Then

$$D(\mathbf{U}^k, \mathcal{J}_{tor}, \infty, N) < c(A)N^{(k-1)/2k}(\log N)^{1/2}. \quad (13.6.4)$$

Beck (see [BC87, pp. 129–130]) deduced several related corollaries from Theorem 13.6.3. The example sets \mathcal{P}_N for Corollary 13.6.4 can be taken as the initial segments of a certain fixed sequence whose choice definitely depends on A . If $d(A) = \lambda$ and A is either a disk (solid sphere) or a cube, then the right side of (13.6.2) takes the form $c(k)(\lambda^k N)^{(k-1)/2k}$. Montgomery [Mon89] has obtained a similar lower bound for cubes and disks.

The problem of estimating discrepancy for \mathcal{J}_c is even more challenging because of “boundary effects.” We state, as an example, a theorem for disks. The right inequality follows from (13.6.4).

THEOREM 13.6.5 *Beck [Bec87] [BC87, Theorem 16A]*

Let \mathcal{J} be ds -generated by a k -disk. Then

$$c_1(k, \epsilon) N^{(k-1)/2k-\epsilon} < D(\mathbf{U}^k, \mathcal{J}_c, \infty, N) < c_2(k) N^{(k-1)/2k} (\log N)^{1/2}. \quad (13.6.5)$$

Because all the lower bounds above come from \mathbf{L}^2 estimates, these various results (13.6.1)–(13.6.5) allow us to make the general statement that for W in the range $2 \leq W \leq \infty$, the magnitude of $D(\mathbf{U}^k, \mathcal{J}, W, N)$ is controlled by $N^{(k-1)/2k}$. Thus there is no extreme discontinuity in asymptotic behavior at $W = \infty$. However, recent work by Beck and Chen proves that there is a discontinuity at some W satisfying $1 \leq W \leq 2$, and the following results indicate that $W = 1$ is a likely candidate.

THEOREM 13.6.6 *Beck, Chen [BC93b]*

Let \mathcal{J} be ds -generated by a convex polygon A with $d(A) < 1$. Then

$$\begin{aligned} D(\mathbf{U}^2, \mathcal{J}_{tor}, W, N) &< c(A, W) N^{(W-1)/2W}, & 1 < W \leq 2; \\ D(\mathbf{U}^2, \mathcal{J}_{tor}, 1, N) &< c(A) (\log N)^2. \end{aligned} \quad (13.6.6)$$

In fact, Theorem 13.6.6 is motivated by the study of discrepancy with respect to halfplanes, and is established by ideas used to establish Theorem 13.9.8 below. Note the similarities of the inequalities (13.6.6) and (13.9.5). After all, a convex polygon is the intersection of a finite number of halfplanes, and so the proof of Theorem 13.6.6 involves carrying out the idea of the proof of Theorem 13.9.8 a finite number of times.

The next theorem shows that powers of N other than $N^{(k-1)/2k}$ may appear for $2 \leq W \leq \infty$. It deals with what has been termed the *isotropic discrepancy* in \mathbf{U}^k .

THEOREM 13.6.7 *Schmidt [Sch75] [BC87, Theorem 15]*

Let \mathcal{J} be the collection of all convex sets in \mathbb{E}^k . Then

$$D(\mathbf{U}^k, \mathcal{J}_i, \infty, N) > c(k) N^{(k-1)/(k+1)}. \quad (13.6.7)$$

The function $N^{(k-1)/(k+1)}$ dominates $N^{(k-1)/2k}$, so that this largest possible choice for \mathcal{J} does in fact yield a larger discrepancy. Beck has shown by probabilistic techniques that the inequality (13.6.7), excepting a possible logarithmic factor, is best possible for $k = 2$.

The following result of Larcher [Lar91] shows that for certain rotation-invariant \mathcal{J} the discrepancy of Kronecker sequences (defined in Section 13.1) will not behave as $cN^{(k-1)/2k}$, but as the square of this quantity.

THEOREM 13.6.8 *Larcher*

Let the sequence of point sets \mathcal{P}_N be the initial segments of a Kronecker sequence in \mathbf{U}^k , and let \mathcal{J} be ds -generated by a cube of edge length $\lambda < 1$. Then, for each N ,

$$\|\Delta(\mathcal{P}_N, \mathcal{J}_i)\|_\infty > c(k) \lambda^{k-1} N^{(k-1)/k}.$$

Furthermore, the exponent $(k-1)/k$ cannot be increased.

13.7 CONGRUENT OBJECTS IN THE UNIT k -CUBE

GLOSSARY

If \mathcal{J} consists of all directly congruent copies of a convex set A , we say that A **dm-generates** \mathcal{J} . Simple examples are given by the collection of all k -disks of a fixed radius r or by the collection of all k -cubes of a fixed edge length λ .

Given a convex set A , there is some evidence for the conjecture that the discrepancy for the dm-generated collection will be essentially as large as that for the ds-generated collection. However, this is generally very difficult to establish, even in very specific situations. There are the following results in this direction. The upper bound inequalities all come from Corollary 13.6.4 above.

THEOREM 13.7.1 *Beck [BC87, Theorem 22A]*

Let \mathcal{J} be dm-generated by a square of edge length λ . Then

$$c_1(\lambda)N^{1/8} < D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) < c_2(\lambda)N^{1/4}(\log N)^{1/2}.$$

It is felt that $N^{1/4}$ gives the proper lower bound, and for \mathcal{J}_i this is definitely true. The lower bound in the next result follows at once from the work of Alexander [Ale91] described in Section 13.9.

THEOREM 13.7.2 *Alexander, Beck*

Let \mathcal{J} be dm-generated by a k -cube of edge length λ . Then

$$c_1(\lambda, k)N^{(k-1)/2k} < D(\mathbf{U}^k, \mathcal{J}_i, \infty, N) < c_2(\lambda, k)N^{(k-1)/2k}(\log N)^{1/2}.$$

A similar result probably holds for k -disks, but this has been established only for $k = 2$.

THEOREM 13.7.3 *Beck [BC87, Theorem 22B]*

Let \mathcal{J} be dm-generated by a 2-disk of radius r . Then

$$c_1(r)N^{1/4} < D(\mathbf{U}^2, \mathcal{J}_i, \infty, N) < c_2(r)N^{1/4}(\log N)^{1/2}.$$

13.8 WORK OF MONTGOMERY

It should be reported that Montgomery [Mon89] has independently developed a lower bound method which, as does Beck's method, uses techniques from harmonic analysis. Montgomery's method, especially in dimension 2, obtains for a number of special classes \mathcal{J} estimates comparable to those obtained by Beck's method. In particular, Montgomery has considered \mathcal{J} that are ds-generated by a region whose boundary is a piecewise smooth simple closed curve.

13.9 HALFSPLANES AND RELATED OBJECTS

GLOSSARY

Segment: Given a compact subset \mathbf{K} and a closed halfspace H in \mathbb{E}^k , $K \cap H$ is called a segment of \mathbf{K} .

Slab: The region between two parallel hyperplanes.

Spherical slice: The intersection of two open hemispheres on a sphere.

Let H be a closed halfspace in \mathbb{E}^k . Then the collection \mathcal{H}^k of all closed halfspaces is dm-generated by H , and if we associate H with the oriented hyperplane ∂H , there is a well known invariant measure ν on \mathcal{H}^k . Further information concerning this and related measures may be found in Chapter 12 of Santaló [San76]. For a compact domain \mathbf{K} in \mathbb{E}^k , it is clear that only the collection \mathcal{H}_i^k , the *segments* of \mathbf{K} , are proper for study, since \mathcal{H}_c^k is empty and \mathcal{H}_{tor}^k is unsuitable.

In this section, it is necessary for the domain \mathbf{K} to be somewhat more general; hence we make only the following broad assumptions:

- (i) \mathbf{K} lies on the boundary of a fixed convex set \mathbf{M} in \mathbb{E}^{k+1} ;
- (ii) $\sigma(\mathbf{K}) = 1$, where σ is the usual k -measure on $\partial\mathbf{M}$.

Since \mathbb{E}^k is the boundary of a convex body in \mathbb{E}^{k+1} , any set in \mathbb{E}^k of unit Lebesgue k -measure satisfies these assumptions. The normalization of assumption (ii) is for convenience, and, by rescaling, the inequalities of this section may be applied to any uniform probability measure on a domain \mathbf{K} in \mathbb{E}^{k+1} . Such rescaling only affects dimensional constants; for standard domains, such as the unit k -sphere \mathbf{S}^k and the unit k -disk \mathbf{D}^k , this will be done without comment.

Although in applications \mathbf{K} will have a simple geometric description, the next theorem treats the general situation and obtains the essentially exact magnitude of $D(\mathbf{K}, \mathcal{H}_i^{k+1}, 2, N)$. If \mathbf{K} lies in \mathbb{E}^k , then \mathcal{H}^{k+1} may be replaced by \mathcal{H}^k . If ν is properly normalized, this change invokes no rescaling.

THEOREM 13.9.1 Alexander [Ale91]

Let \mathcal{K} be the collection of all \mathbf{K} satisfying assumptions (i) and (ii) above. Then

$$c_1(k)N^{(k-1)/2k} < \inf_{K \in \mathcal{K}} D(\mathbf{K}, \mathcal{H}_i^{k+1}, 2, N) < c_2(\mathbf{M})N^{(k-1)/2k}. \quad (13.9.1)$$

The upper bound of (13.9.1) can be proved by an indirect probabilistic method introduced by Alexander [Ale72] for $\mathbf{K} = \mathbf{S}^2$, but the method of Beck and Chen [BC90] also may be applied for standard choices of \mathbf{K} such as \mathbf{U}^k and \mathbf{D}^k . When $\mathbf{M} = \mathbf{K} = \mathbf{S}^k$, the segments are the spherical caps. For this important special case the upper bound is due to Stolarsky [Sto73], while the lower bound is due to Beck [Bec84] (see also [BC87, Theorem 24B]).

Since the ν -measure of the halfspaces that separate \mathbf{M} is less than $c(k)d(\mathbf{M})$, inequality (13.2.1) may be applied to obtain a lower bound for $D(\mathbf{K}, \mathcal{H}_i^{k+1}, \infty, N)$. The upper bound in the following theorem should be taken in the context of actual

applications such as \mathbf{M} being a k -sphere \mathbf{S}^k , a compact convex body in \mathbb{E}^k , or more generally, a compact convex hypersurface in \mathbb{E}^{k+1} .

THEOREM 13.9.2 *Alexander, Beck*

Let \mathcal{K} be the collection of \mathbf{K} satisfying assumptions (i) and (ii) above. Furthermore, suppose that \mathbf{M} is of finite diameter. Then

$$c_3(k)(d(\mathbf{M}))^{-1/2}N^{(k-1)/2k} < \inf_{K \in \mathcal{K}} D(\mathbf{K}, \mathcal{H}_i^{k+1}, \infty, N) < c_4(\mathbf{M})N^{(k-1)/2k}(\log N)^{1/2}. \quad (13.9.2)$$

For $\mathbf{M} = \mathbf{K} = \mathbf{S}^k$, inequalities (13.9.2) are due to Beck, improving a slightly weaker lower bound by Schmidt [Sch69]. Consideration of $\mathbf{K} = \mathbf{U}^2$ makes it obvious that there exists an aligned right triangle with discrepancy at least $cN^{1/4}$, as stated in Section 13.1. For the case $\mathbf{M} = \mathbf{K} = \mathbf{D}^2$, a unit 2-disk (Roth's *disk-segment problem*), Beck [Bec83] (see also [BC87, Theorem 23A]) obtained inequalities (13.9.2), excepting a factor $(\log N)^{-7/2}$ in the lower bound. Later, Alexander [Ale90] improved the lower bound, and Matoušek [Mat95] obtained essentially the same upper bound. Matoušek's work on \mathbf{D}^2 makes it seem likely that Beck's factor $(\log N)^{1/2}$ in his general upper bound theorem might be removable in many specific situations, but this is very challenging.

THEOREM 13.9.3 *Alexander, Matoušek*

For Roth's disk-segment problem,

$$c_1 N^{1/4} < D(\mathbf{D}^2, \mathcal{H}_i^2, \infty, N) < c_2 N^{1/4}. \quad (13.9.3)$$

Alexander's lower bound method, by the nature of the convolutions employed, gives information on the discrepancy of slabs. This is especially apparent in the recent work of Chazelle, Matoušek, and Sharir, who have developed a more direct and geometrically transparent version of Alexander's method. The following theorem on the discrepancy of thin slabs is a corollary to their technique. It is clear that if a slab has discrepancy Δ , then one of the two bounding halfspaces has discrepancy at least $\Delta/2$.

THEOREM 13.9.4 *Chazelle, Matoušek, Sharir* [CMS95]

Let N points lie in the unit cube \mathbf{U}^k . Then there exists a slab \mathbf{T} of width $c_1(k)N^{-1/k}$ such that $\Delta(\mathbf{T}) > c_2(k)N^{(k-1)/2k}$.

Alexander [Ale94] has investigated the effect of the dimension k on the discrepancy of halfspaces, and obtained somewhat complicated inequalities that imply the following result.

THEOREM 13.9.5 *Alexander*

For the lower bounds in inequalities (13.9.1) and (13.9.2) above, there is an absolute positive constant c such that one may choose $c_1(k) > ck^{-3/4}$ and $c_3(k) > ck^{-1}$.

Schmidt [Sch69] studied the discrepancy of spherical slices (the intersection of two open hemispheres) on \mathbf{S}^k . Associating a hemisphere with its pole, Schmidt identified ν with the normalized product measure on $\mathbf{S}^k \times \mathbf{S}^k$. Blümlinger [Blü91] demonstrated a surprising relationship between halfspace (spherical cap) and slice discrepancy for \mathbf{S}^k . However, his definition for ν in terms of Haar measure on $SO(k+1)$ differed somewhat from Schmidt's.

THEOREM 13.9.6 *Blümlinger*

Let \mathcal{S}^k be the collection of slices of \mathbf{S}^k . Then

$$c(k)D(\mathbf{S}^k, \mathcal{H}_i^{k+1}, 2, N) < D(\mathbf{S}^k, \mathcal{S}^k, 2, N). \quad (13.9.4)$$

For the next result, the left inequality follows from inequalities (13.2.1), (13.9.1), and (13.9.4). Blümlinger uses a version of Beck's probabilistic method to establish the right inequality.

THEOREM 13.9.7 *Blümlinger*

For slice discrepancy on \mathbf{S}^k ,

$$c_1(k)N^{(k-1)/2k} < D(\mathbf{S}^k, \mathcal{S}^k, \infty, N) < c_2(k)N^{(k-1)/2k}(\log N)^{1/2}.$$

Grabner [Gra91] has given an Erdős-Turán type upper bound on spherical cap discrepancy in terms of spherical harmonics. This adds to the considerable body of results extending inequalities for exponential sums to other sets of orthonormal functions, and thereby extends the Weyl theory.

All of the results so far in this section treat $2 \leq W \leq \infty$. For W in the range $1 \leq W < 2$ there is mystery, but we do have the following result, related to inequality (13.6.6), showing that a dramatic change in asymptotic behavior occurs in the range $1 \leq W \leq 2$. For \mathbf{U}^2 , Beck and Chen show that regular grid points will work for the upper bound example for $W = 1$, and they are able to modify their method to apply to any bounded convex domain in \mathbb{E}^2 .

THEOREM 13.9.8 *Beck, Chen* [BC93a]

Let \mathbf{K} be a bounded convex domain in \mathbb{E}^2 . Then

$$\begin{aligned} D(\mathbf{K}, \mathcal{H}_i^2, W, N) &< c(\mathbf{K}, W)N^{(W-1)/2W}, & 1 < W \leq 2; \\ D(\mathbf{K}, \mathcal{H}_i^2, 1, N) &< c(\mathbf{K})(\log N)^2. \end{aligned} \quad (13.9.5)$$

13.10 BOUNDARIES OF GENERATORS FOR HOMOTHETICALLY INVARIANT J

We have already noted several factors that play a role in determining whether $D(\mathbf{K}, \mathcal{J}, W, N)$ behaves like N^r as opposed to $(\log N)^s$. Beck's work shows that if \mathcal{J} is dm-generated, $D(\mathbf{K}, \mathcal{J}, \infty, N)$ behaves as N^r . However, the work of Beck and Chen clearly shows that if W is sufficiently small, then even for motion-invariant \mathcal{J} , it may be that $D(\mathbf{K}, \mathcal{J}, W, N)$ is bounded above by $(\log N)^s$.

Beck [Bec88] has extensively studied $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N)$ under the assumption that \mathcal{J} is homothetically invariant, and in this section we shall record some of the results obtained.

It turns out that the boundary shape of a generator is the critical element in determining to which, if either, class \mathcal{J} belongs. Remarkably, for the "typical" homothetically invariant class \mathcal{J} , $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N)$ oscillates infinitely often to be larger than $N^{1/4-\epsilon}$ and smaller than $(\log N)^{4+\epsilon}$.

GLOSSARY

The convex set A ***h-generates*** \mathcal{J} if \mathcal{J} consists of all homothetic images B of A with $d(B) \leq d(A)$.

Blaschke-Hausdorff metric: The metric on the space $\text{CONV}(2)$ of all compact convex sets in \mathbb{E}^2 in which the distance between two sets is the minimum distance from any point of one set to the other.

A set is of ***first category*** if it is a countable union of nowhere dense sets.

If one considers the two examples of \mathcal{J} being h-generated by an aligned square and by a disk, previously stated results make it very likely that shape strongly affects discrepancy for homothetically invariant \mathcal{J} . The first two theorems quantify this phenomenon for two very standard boundary shapes, first polygons, then smooth closed curves.

THEOREM 13.10.1 Beck [Bec88] [BC87, Corollary 20D]

Let \mathcal{J} be h-generated by a convex polygon A . Then, for any $\epsilon > 0$,

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) = o((\log N)^{4+\epsilon}). \quad (13.10.1)$$

Beck and Chen [BC89] have given a less complicated argument that obtains $o((\log N)^{5+\epsilon})$ on the right side of (13.10.1).

THEOREM 13.10.2 Beck [Bec88] [BC87, Corollary 19F]

Let A be a compact convex set in \mathbb{E}^2 with a twice continuously differentiable boundary curve having strictly positive curvature. If A h-generates \mathcal{J} , then for $N \geq 2$,

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) > c(A)N^{1/4}(\log N)^{-1/2}. \quad (13.10.2)$$

Recently, for sufficiently smooth positively curved bodies, Drmota [Drm93] has extended (13.10.2) into higher dimensions and also removed the logarithmic factor. Thus, he obtains a lower bound of the form $c(A)N^{(k-1)/2k}$, along with the standard upper bound obtained by Beck's probabilistic method.

Let $\text{CONV}(2)$ denote the usual locally compact space of all compact convex sets in \mathbb{E}^2 endowed with the Blaschke-Hausdorff metric. There is the following surprising result, which quantifies the oscillatory behavior mentioned above.

THEOREM 13.10.3 Beck [BC87, Theorem 21]

Let $\epsilon > 0$ be given. For all A in $\text{CONV}(2)$, excepting a set of first category, if \mathcal{J} is h-generated by A , then each of the following two inequalities is satisfied infinitely often:

- (i) $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) < (\log N)^{4+\epsilon}$.
- (ii) $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) > N^{1/4}(\log N)^{-(1+\epsilon)/2}$.

In fact, the final theorem of this section will say more about the rationale of such estimates.

The next theorem gives the best lower bound estimate known if it is assumed only that the generator A has nonempty interior, certainly a minimal hypothesis.

THEOREM 13.10.4 *Beck [Bec88] [BC87, Corollary 19G]*

If \mathcal{J} is h-generated by a compact convex set A having positive area, then

$$D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) > c(A)(\log N)^{1/2}.$$

Possibly the right side should be $c(A) \log N$, which would be best possible as the example of aligned squares demonstrates. Lastly, we discuss the important theorem underlying most of these results about h-generated \mathcal{J} . Let A be a member of CONV(2) with nonempty interior, and for each integer $l \geq 3$ let A_l be an inscribed l -gon of maximal area. The N th **approximability number** $\xi_N(A)$ is defined as the smallest integer l such that the area of $A \setminus A_l$ is less than l^2/N .

THEOREM 13.10.5 *Beck [Bec88] [BC87, Corollary 19H, Theorem 20C]*

Let A be a member of CONV(2) with nonempty interior. Then if \mathcal{J} is h-generated by A , we have

$$c_1(A)(\xi_N(A))^{1/2}(\log N)^{-1/4} < D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N) < c_2(A, \epsilon)\xi_N(A)(\log N)^{4+\epsilon}. \quad (13.10.3)$$

The proof of the preceding fundamental theorem, which is in fact the join of two major theorems, is long, but the import is clear; namely, that for h-generated \mathcal{J} , if one understands $\xi_N(A)$, then one essentially understands $D(\mathbf{U}^2, \mathcal{J}_{tor}, \infty, N)$. If $\xi_N(A)$ remains nearly constant for long intervals, then A acts like a polygon and D will drift below $(\log N)^{4+2\epsilon}$. If, at some stage, ∂A behaves as if it consists of circular arcs, then $\xi_N(A)$ will begin to grow as $cN^{1/2}$.

For still more information concerning the material in this section, along with the proofs, see [BC87, Chapter 7]. Károlyi [Kár95a, Kár95b] has extended the idea of approximability number to higher dimensions and obtained upper bounds analogous to those in (13.10.3).

13.11 $D(\mathbf{K}, \mathcal{J}, 2, N)$ IN LIGHT OF DISTANCE GEOMETRY

Although knowledge of $D(\mathbf{K}, \mathcal{J}, \infty, N)$ is our highest aim, in the great majority of problems this is achieved by first obtaining bounds on $D(\mathbf{K}, \mathcal{J}, 2, N)$. In this section, we briefly show how this function fits nicely into the theory of metric spaces of negative type. In our situation, the distance between points will be given by a Crofton formula with respect to the measure ν on \mathcal{J} . This approach evolved from a paper written in 1971 by Alexander and Stolarsky investigating extremal problems in distance geometry, and has been developed in a number of subsequent papers by both authors studying special cases. However, we reverse history and leap immediately to a formulation suitable for our present purposes. We avoid mention of certain technical assumptions concerning \mathcal{J} and ν which cause no difficulty in practice.

Assume that \mathbf{K} is a compact convex set in \mathbb{E}^k and that $\mathcal{J} = \mathcal{J}_c$. This latter assumption causes no loss of generality since one can always just redefine \mathcal{J} . Let ν , as usual, be a measure on \mathcal{J} , with the further assumption that $\nu(\mathcal{J}) < \infty$.

Definition: If p and q are points in \mathbf{K} , the set A in \mathcal{J} is said to **separate** p and q if A contains exactly one of these two points. The **distance function** ρ on \mathbf{K} is

defined by the Crofton formula $\rho(p, q) = (1/2) \nu\{J \mid J \text{ separates } p \text{ and } q\}$, and if μ is any signed measure on \mathbf{K} having finite positive and negative parts, one defines the functional $\mathbf{I}(\mu)$ by

$$\mathbf{I}(\mu) = \iint \rho(p, q) d\mu(p) d\mu(q).$$

With these definitions one obtains the following representation for $\mathbf{I}(\mu)$.

THEOREM 13.11.1 *Alexander* [Ale91]

One has

$$\mathbf{I}(\mu) = \int_{\mathcal{J}} \mu(A) \mu(\mathbf{K} \setminus A) d\nu(A). \quad (13.11.1)$$

For μ satisfying the condition of total mass zero, $\int_K d\mu = 0$, the integrand in (13.11.1) becomes $-(\mu(A))^2$. The signed measures $\mu = \mu^+ - \mu^-$ that we are considering, with μ^- being a uniform probability measure on \mathbf{K} and μ^+ consisting of N atoms of equal weight $1/N$, certainly have total mass zero. Here one has $\Delta(A) = N\mu(A)$. Hence there is the following corollary.

COROLLARY 13.11.2

For the signed measures μ presently considered, if \mathcal{P} denotes the N points supporting μ^+ , then

$$-N^2 \mathbf{I}(\mu) = \int_{\mathcal{J}} (\Delta(A))^2 d\nu(A) = (\|\Delta(\mathcal{P}, \mathcal{J})\|_2)^2. \quad (13.11.2)$$

Thus if one studies the metric ρ , it may be possible to prove that $-\mathbf{I}(\mu) > f(N)$, whence it follows that $(D(\mathbf{K}, \mathcal{J}, 2, N))^2 > N^2 f(N)$. If \mathcal{J} consists of the halfspaces of \mathbb{E}^k , then ρ is the Euclidean metric. In this important special case, Alexander [Ale91] was able to make good estimates. Chazelle, Matoušek, and Sharir [CMS95] and A.D. Rogers [Rog94] contributed still more techniques for treating the halfspace problem.

If μ_1 and μ_2 are any two signed measures of total mass 1 on \mathbf{K} , then one can define the **relative discrepancy** $\Delta(A) = N(\mu_1(A) - \mu_2(A))$. The first equality of (13.11.2) still holds if $\mu = \mu_1 - \mu_2$. A signed measure μ_0 of total mass 1 is termed **optimal** if it solves the integral equation $\int_K \rho(x, y) d\mu(y) = \lambda$ for some positive number λ . If an optimal measure μ_0 exists, then $\mathbf{I}(\mu_0) = \lambda$ maximizes \mathbf{I} on the class of all signed Borel measures of total mass 1 on \mathbf{K} . In the presence of an optimal measure, one has the following very pretty identity.

THEOREM 13.11.3 *Generalized Stolarsky Identity*

Suppose that the measure μ_0 is optimal on \mathbf{K} , and that μ is any signed measure of total mass 1 on \mathbf{K} . If Δ is the relative discrepancy with respect to μ_0 and μ , then

$$N^2 \mathbf{I}(\mu) + \int_{\mathcal{J}} (\Delta(A))^2 d\nu(A) = N^2 \mathbf{I}(\mu_0). \quad (13.11.3)$$

The first important example of this formula is due to Stolarsky [Sto73] where he treated the sphere \mathbf{S}^k , taking as μ the uniform atomic measure supported by N variable points. For \mathbf{S}^k it is clear that the uniform probability measure μ_0 is optimal. His integrals involving the spherical caps are equivalent, up to a scale factor, to integrals with respect to the measure on the halfspaces of \mathbb{E}^k for which

ρ is the Euclidean metric. Stolarsky's tying of a geometric extremal problem to Schmidt's work on the discrepancy of spherical caps was a major step forward in the study of discrepancy and of distance geometry.

Very little has been done to investigate the deeper nature of the individual metrics ρ determined by classes \mathcal{J} other than halfspaces. They are all metrics of ***negative type***, which essentially means that $\mathbf{I}(\mu) \leq 0$ if μ has total mass 0. There is a certain amount of general theory, begun by Schoenberg and developed by a number of others, but it does not apply directly to the problem of estimating discrepancy.

13.12 UNIFORM PLACEMENT OF POINTS ON SPHERES

As demonstrated by Stolarsky, formula (13.11.3) shows that if one places N points on \mathbf{S}^k so that the sum of all distances is maximized, then $D(\mathbf{S}^k, \mathcal{H}_i^k, 2, N)$ is achieved by this arrangement. Berman and Hanes [BH77] have given a pretty algorithm that searches for optimal configurations. For $k = 2$, while the exact configurations are not known for $N \geq 5$, this algorithm appears to be successful for $N \leq 50$. For such an N surprisingly few rival configurations will be found. Lubotsky, Phillips, and Sarnak [LPS86] have given an algorithm, based on iterations of a specially chosen element in $SO(3)$, which can be used to place many thousands of reasonably well distributed points on \mathbf{S}^2 . Difficult analysis shows that these points are well placed, but not optimally placed, relative to \mathcal{H}_i^2 . On the other hand, it is shown that these points are essentially optimally placed with respect to a nongeometric operator discrepancy. Data concerning applications to numerical integration are also included in the paper. More recently, Rakhmanov, Saff, and Zhou [RSZ94] have studied the problem of placing points uniformly on a sphere relative to optimizing certain functionals, and they state a number of interesting conjectures.

In yet another theoretical direction, the existence of very well distributed point sets on \mathbf{S}^k allows the sphere, after difficult analysis, to be closely approximated by equi-edged zonotopes (sums of line segments). The recent papers of Wagner [Wag93] and of Bourgain and Lindenstrauss [BL93] treat this problem.

13.13 COMBINATORIAL DISCREPANCY

GLOSSARY

A **2-coloring** of \mathbf{X} is a mapping $\chi : \mathbf{X} \rightarrow \{-1, 1\}$. For each such χ there is a natural integer-valued set function μ_χ on the finite subsets of \mathbf{X} defined by $\mu_\chi(A) = \sum_{x \in A} \chi(x)$, and if \mathcal{J} is a given family of finite subsets of \mathbf{X} we define

$$D(\mathbf{X}, \mathcal{J}) = \min_{\chi} \max_{A \in \mathcal{J}} |\mu_\chi(A)|.$$

Degree: If \mathcal{J} is a collection of subsets of a finite set X , $\deg \mathcal{J} = \max\{|\mathcal{J}(x)| \mid x \in \mathbf{X}\}$, where $\mathcal{J}(x)$ is the subcollection consisting of those members of \mathcal{J} that contain x .

The collection \mathcal{J} **shatters** a set $S \subset X$ if, for any given subset $B \subset S$, there exists A in \mathcal{J} such that $B = A \cap S$. The **VC-dimension** of \mathcal{J} is defined by $\dim_{vc} \mathcal{J} = \max\{|S| \mid S \subset \mathbf{X}, \mathcal{J} \text{ shatters } S\}$. For $m \leq |\mathbf{X}|$, the **primal shatter function** $\pi_{\mathcal{J}}$ is defined by

$$\pi_{\mathcal{J}}(m) = \max_{\substack{Y \subset \mathbf{X} \\ |Y| \leq m}} |\{Y \cap A \mid A \in \mathcal{J}\}|.$$

The **dual shatter function** is defined by $\pi_{\mathcal{J}}^*(m) = \pi_{\mathcal{J}^*}(m)$, where $\mathbf{X}^* = \mathcal{J}$, and $\mathcal{J}^* = \{\mathcal{J}(x) \mid x \in \mathbf{X}\}$.

Techniques in combinatorial discrepancy theory have proved very powerful in this geometric setting. Here one 2-colors a discrete set and studies the discrepancy of a special class \mathcal{J} of subsets as measured by $|\#\text{red} - \#\text{blue}|$. If one 2-colors the first N positive integers, then the beautiful “1/4 theorem” of Roth [Rot64] says that there will always be an arithmetic progression having discrepancy at least $cN^{1/4}$. This result should be compared to van der Waerden’s theorem, which says that there is a long monochromatic progression, whose discrepancy obviously will be its length. However, it is known that this length need not be more than $\log N$, and the minimax might be as small as $\log \log \dots \log N$ (here the number of iterated logarithms may be arbitrarily large). Moreover, general results concerning combinatorial discrepancy, for example, those that use the Vapnik-Chervonenkis dimension, are very useful in computational geometry; cf. [Chapter 44](#).

Combinatorial discrepancy theory involves discrepancy estimates arising from 2-colorings of a set \mathbf{X} . Upper bound estimates of combinatorial discrepancy have proved to be very helpful in obtaining upper bound estimates of geometric discrepancy. In this final section we briefly discuss various properties of the collection \mathcal{J} that lead to useful upper bound estimates of combinatorial discrepancy.

The simplest property of the collection \mathcal{J} is its cardinality $|\mathcal{J}|$. Here, Spencer obtained a fine result.

THEOREM 13.13.1 *Spencer* [AS93]

Let \mathbf{X} be a finite set. If $|\mathcal{J}| \geq |\mathbf{X}|$, then

$$D(\mathbf{X}, \mathcal{J}) \leq c \left(|\mathbf{X}| \log \left(1 + \frac{|\mathcal{J}|}{|\mathbf{X}|} \right) \right)^{1/2}.$$

Applications and extensions of the following theorem may be found in [BC87, [Chapter 8](#)].

THEOREM 13.13.2 *Beck, Fiala* [BF81] [BC87, Lemma 8.5.]

Let \mathbf{X} be a finite set. Then

$$D(\mathbf{X}, \mathcal{J}) \leq 2 \deg \mathcal{J} - 1.$$

Since $\pi_{\mathcal{J}}(m) = 2^m$ if and only if $\dim_{vc} \mathcal{J} \geq m$, the function $\pi_{\mathcal{J}}$ contains much more information than does VC-dimension alone. If $\dim_{vc} \mathcal{J} = d$, then $\pi_{\mathcal{J}}(m)$ is polynomially bounded by cm^d . However, in many geometric situations this bound on the shatter function can be improved, leading to better discrepancy bounds. Detailed discussions may be found in the papers by Haussler and Welzl [HW87] and by Chazelle and Welzl [CW89].

Dual objects are defined in the usual manner (see [Glossary](#)). We state several recent results.

THEOREM 13.13.3 *Matoušek, Welzl, Wernisch* [MWW93]

Suppose that $(\mathbf{X}, \mathcal{J})$ is a finite set system with $|\mathbf{X}| = n$. If $\pi_{\mathcal{J}}(m) \leq c_1 m^d$ for $m \leq n$, then

$$\begin{aligned} D(\mathbf{X}, \mathcal{J}) &\leq c_2 n^{(d-1)/2d} (\log n)^{1+1/2d}, & d > 1, \\ D(\mathbf{X}, \mathcal{J}) &\leq c_3 (\log n)^{5/2}, & d = 1. \end{aligned} \quad (13.13.1)$$

If $\pi_{\mathcal{J}}^*(m) \leq c_4 m^d$ for $m \leq |\mathcal{J}|$, then

$$\begin{aligned} D(\mathbf{X}, \mathcal{J}) &\leq c_5 n^{(d-1)/2d} \log n, & d > 1, \\ D(\mathbf{X}, \mathcal{J}) &\leq c_6 (\log n)^{3/2}, & d = 1. \end{aligned} \quad (13.13.2)$$

More recently, Matoušek [Mat95] has shown that the factor $(\log n)^{1+1/2d}$ may be dropped from inequality (13.13.1) for $d > 1$, and has applied this result to half-spaces with great effect (see inequality (13.9.3)). One part of Matoušek's argument depends on combinatorial results of Haussler [Hau95].

13.14 SOURCES AND RELATED MATERIAL

FURTHER READING

The principal surveys on discrepancy theory are [BC87], [Cha00], [DT97], [KN74], [Mat99] and [Sch77b].

Auxiliary texts relating to this chapter include [AS93], [Cas57], [Cas59], and [San76].

RELATED CHAPTERS

- [Chapter 1: Finite point configurations](#)
- [Chapter 2: Packing and covering](#)
- [Chapter 11: Euclidean Ramsey theory](#)
- [Chapter 12: Discrete aspects of stochastic geometry](#)
- [Chapter 36: Range searching](#)
- [Chapter 40: Randomization and derandomization](#)
- [Chapter 44: The discrepancy method in computational geometry](#)
- [Chapter 49: Computer graphics](#)

REFERENCES

- [Ale72] J.R. Alexander. On the sum of distances between n points on a sphere. *Acta Math. Hungar.*, 23:443–448, 1972.
- [Ale90] J.R. Alexander. Geometric methods in the study of irregularities of distribution. *Combinatorica*, 10:115–136, 1990.

- [Ale91] J.R. Alexander. Principles of a new method in the study of irregularities of distribution. *Invent. Math.*, 103:279–296, 1991.
- [Ale94] J.R. Alexander. The effect of dimension on certain geometric problems of irregularities of distribution. *Pacific J. Math.*, 165:1–15, 1994.
- [AS93] N. Alon and J. Spencer. *The Probabilistic Method*. Wiley, New York, 1993.
- [Bak99] R.C. Baker. On irregularities of distribution II. *J. London Math. Soc.*, 59:50–64, 1999.
- [Bec83] J. Beck. On a problem of K.F. Roth concerning irregularities of point distribution. *Invent. Math.*, 74:477–487, 1983.
- [Bec84] J. Beck. Sums of distances between points on a sphere – an application of the theory of irregularities of distribution to discrete geometry. *Mathematika*, 31:33–41, 1984.
- [Bec87] J. Beck. Irregularities of distribution I. *Acta Math.*, 159:1–49, 1987.
- [Bec88] J. Beck. Irregularities of distribution II. *Proc. London Math. Soc.*, 56:1–50, 1988.
- [Bec89] J. Beck. A two-dimensional van Aardenne-Ehrenfest theorem in irregularities of distribution. *Compositio Math.*, 72:269–339, 1989.
- [Bec94] J. Beck. Probabilistic diophantine approximation I: Kronecker sequences. *Ann. of Math.*, 140:449–502, 1994.
- [BC87] J. Beck and W.W.L. Chen. *Irregularities of Distribution*. Volume 89 of *Cambridge Tracts in Math.*, Cambridge University Press, 1987.
- [BC89] J. Beck and W.W.L. Chen. Irregularities of point distribution relative to convex polygons. In G. Halász and V.T. Sós, editors, *Irregularities of Partitions*, volume 8 of *Algorithms Combin.*, pages 1–22. Springer-Verlag, Berlin, 1989.
- [BC90] J. Beck and W.W.L. Chen. Note on irregularities of distribution II. *Proc. London Math. Soc.*, 61:251–272, 1990.
- [BC93a] J. Beck and W.W.L. Chen. Irregularities of point distribution relative to half planes I. *Mathematika*, 40:102–126, 1993.
- [BC93b] J. Beck and W.W.L. Chen. Irregularities of point distribution relative to convex polygons II. *Mathematika*, 40:127–136, 1993.
- [BF81] J. Beck and T. Fiala. Integer-making theorems. *Discrete Appl. Math.*, 3:1–8, 1981.
- [BH77] J. Berman and K. Hanes. Optimizing the arrangement of points on the unit sphere. *Math. Comp.*, 31:1006–1008, 1977.
- [Blü91] M. Blümlinger. Slice discrepancy and irregularities of distribution on spheres. *Mathematika*, 38:105–116, 1991.
- [BL93] J. Bourgain and J. Lindenstrauss. Approximating the ball by a Minkowski sum of segments with equal length. *Discrete Comput. Geom.*, 9:131–144, 1993.
- [Cas57] J.W.S. Cassels. *An Introduction to Diophantine Approximation*. Volume 45 of *Cambridge Tracts in Math.*, Cambridge University Press, 1957.
- [Cas59] J.W.S. Cassels. *An Introduction to the Geometry of Numbers*. Volume 99 of *Grundlehren Math. Wiss.*, Springer-Verlag, Berlin, 1959.
- [Cha00] B. Chazelle. *The Discrepancy Method*. Cambridge University Press, 2000.
- [CMS95] B. Chazelle, J. Matoušek, and M. Sharir. An elementary approach to lower bounds in geometric discrepancy. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift*, *Discrete Comput. Geom.*, 13:363–381, 1995.
- [CW89] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489, 1989.

- [Che80] W.W.L. Chen. On irregularities of distribution. *Mathematika*, 27:153–170, 1980.
- [Che83] W.W.L. Chen. On irregularities of distribution II. *Quart. J. Math. Oxford*, 34:257–279, 1983.
- [Che85] W.W.L. Chen. On irregularities of distribution and approximate evaluation of certain functions. *Quart. J. Math. Oxford*, 36:173–182, 1985.
- [Che87] W.W.L. Chen. On irregularities of distribution and approximate evaluation of certain functions II. In A.C. Adolphson, J.B. Conrey, A. Ghosh and R.I. Yager, editors, *Analytic Number Theory and Diophantine Problems*, volume 70 of *Progress in Mathematics*, pages 75–86. Birkhäuser-Verlag, Boston, 1987.
- [CS02] W.W.L. Chen and M.M. Skriganov. Explicit constrictions in the classical mean squares problem in irregularities of point distribution. *J. Reine Angew. Math.*, 545:67–95, 2002.
- [Dav56] H. Davenport. Note on irregularities of distribution. *Mathematika*, 3:131–135, 1956.
- [Drm93] M. Drmota. Irregularities of distribution and convex sets. *Grazer Math. Ber.*, 318:9–16, 1993.
- [Drm96] M. Drmota. Irregularities of distribution with respect to polytopes. *Mathematika*, 43:108–119, 1996.
- [DT97] M. Drmota and R.F. Tichy. *Sequences, Discrepancies and Applications*. Volume 1651 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1997.
- [Gra91] P.J. Grabner. Erdős-Turán type discrepancy bounds. *Monatsh. Math.*, 111:127–135, 1991.
- [Hal81] G. Halász. On Roth's method in the theory of irregularities of point distributions. In H. Halberstam and C. Hooley, editors, *Recent Progress in Analytic Number Theory, Volume 2*, pages 79–94. Academic Press, London, 1981.
- [Hal60] J.H. Halton. On the efficiency of certain quasirandom sequences of points in evaluating multidimensional integrals. *Num. Math.*, 2:84–90, 1960.
- [HZ69] J.H. Halton and S.K. Zaremba. The extreme and L^2 discrepancies of some plane sets. *Monatsh. Math.*, 73:316–328, 1969.
- [Hau95] D. Haussler. Sphere packing numbers for subsets of the Boolean n -cube with bounded Vapnik-Chervonenkis dimension. *J. Combin. Theory Ser. A*, 69:217–232, 1995.
- [HW87] D. Haussler and E. Welzl. ϵ -nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.
- [Kár95a] G. Károlyi. Geometric discrepancy theorems in higher dimensions. *Studia Sci. Math. Hungar.*, 30:59–94, 1995.
- [Kár95b] G. Károlyi. Irregularities of point distributions with respect to homothetic convex bodies. *Monatsh. Math.*, 120:247–279, 1995.
- [KN74] L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*. Wiley, New York, 1974.
- [Lar91] G. Larcher. On the cube discrepancy of Kronecker sequences. *Arch. Math. (Basel)*, 57:362–369, 1991.
- [LPS86] A. Lubotsky, R. Phillips, and P. Sarnak. Hecke operators and distributing points on a sphere. *Comm. Pure Appl. Math.*, 39:149–186, 1986.
- [Mat95] J. Matoušek. Tight upper bounds for the discrepancy of half-spaces. In I. Bárány and J. Pach, editors, *The László Fejes Tóth Festschrift, Discrete Comput. Geom.*, 13:593–601, 1995.

- [Mat99] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Volume 18 of *Algorithms Combin.*, Springer-Verlag, Berlin, 1999.
- [MWW93] J. Matoušek, E. Welzl, and L. Wernisch. Discrepancy and approximations for bounded VC-dimension. *Combinatorica*, 13:455–467, 1993.
- [Mon89] H.L. Montgomery. Irregularities of distribution by means of power sums. In *Congress of Number Theory (Zarautz)*, pages 11–27. Universidad del País Vasco, Bilbao, 1989.
- [RSZ94] E.A. Rakhmanov, E.B. Saff, and Y.M. Zhou. Minimal discrete energy on the sphere. *Math. Res. Lett.*, 1:647–662, 1994.
- [Rog94] A.D. Rogers. A functional from geometry with applications to discrepancy estimates and the Radon transform. *Trans. Amer. Math. Soc.*, 341:275–313, 1994.
- [Rot54] K.F. Roth. On irregularities of distribution. *Mathematika*, 1:73–79, 1954.
- [Rot64] K.F. Roth. Remark concerning integer sequences. *Acta Arith.*, 9:257–260, 1964.
- [Rot76] K.F. Roth. On irregularities of distribution II. *Comm. Pure Appl. Math.*, 29:749–754, 1976.
- [Rot80] K.F. Roth. On irregularities of distribution IV. *Acta Arith.*, 37:67–75, 1980.
- [Ruz93] I.Z. Ruzsa. The discrepancy of rectangles and squares. *Grazer Math. Ber.*, 318:135–140, 1993.
- [San76] L.A. Santaló. *Integral Geometry and Geometric Probability*. Volume 1 of *Encyclopedia of Mathematics*, Addison-Wesley, Reading, 1976.
- [Sch69] W.M. Schmidt. Irregularities of distribution III. *Pacific J. Math.*, 29:225–234, 1969.
- [Sch72a] W.M. Schmidt. Irregularities of distribution VI. *Compositio Math.*, 24:63–74, 1972.
- [Sch72b] W.M. Schmidt. Irregularities of distribution VII. *Acta Arith.*, 21:45–50, 1972.
- [Sch75] W.M. Schmidt. Irregularities of distribution IX. *Acta Arith.*, 27:385–396, 1975.
- [Sch77a] W.M. Schmidt. Irregularities of distribution X. In H. Zassenhaus, editor, *Number Theory and Algebra*, pages 311–329. Academic Press, New York, 1977.
- [Sch77b] W.M. Schmidt. *Irregularities of Distribution*. Volume 56 of *Lecture Notes on Mathematics and Physics*, Tata, Bombay, 1977.
- [Skr94] M.M. Skriganov. Constructions of uniform distributions in terms of geometry of numbers. *St. Petersburg Math. J. (Algebra i. Analiz)*, 6:200–230, 1994.
- [Sto73] K.B. Stolarsky. Sums of distances between points on a sphere II. *Proc. Amer. Math. Soc.*, 41:575–582, 1973.
- [vA-E45] T. van Aardenne-Ehrenfest. Proof of the impossibility of a just distribution of an infinite sequence of points over an interval. *Nederl. Akad. Wetensch. Proc.*, 48:266–271, 1945 (*Indagationes Math.*, 7:71–76, 1945).
- [vA-E49] T. van Aardenne-Ehrenfest. On the impossibility of a just distribution. *Nederl. Akad. Wetensch. Proc.*, 52:734–739, 1949 (*Indagationes Math.*, 11:264–269, 1949).
- [vdC35a] J.G. van der Corput. Verteilungsfunktionen I. *Proc. Kon. Ned. Akad. v. Wetensch.*, 38:813–821, 1935.
- [vdC35b] J.G. van der Corput. Verteilungsfunktionen II. *Proc. Kon. Ned. Akad. v. Wetensch.*, 38:1058–1066, 1935.
- [Vil67] I.V. Vilenkin. Plane nets of integration. *USSR Comput. Math. and Math. Phys.*, 7:258–267, 1967.
- [Wag93] G. Wagner. On a new method for constructing good point sets on spheres. *Discrete Comput. Geom.*, 9:111–129, 1993.

- [Wey16] H. Weyl. Über die Gleichverteilung von Zahlen mod Eins. *Math. Ann.*, 77:313–352, 1916.
- [Woź91] H. Woźniakowski. Average case complexity of multivariate integration. *Bull. Amer. Math. Soc.*, 24:185–194, 1991.

14 TOPOLOGICAL METHODS

Rade T. Živaljević

INTRODUCTION

A problem is solved or some other goal achieved by “topological methods” if in our arguments we appeal to the “form,” the “shape,” or the “global” rather than “local” structure of the object or configuration space associated with the phenomenon we are interested in. This configuration space is typically a manifold or a simplicial complex. The global properties of the configuration space are usually expressed in terms of its homology and homotopy groups, which capture the idea of the higher (dis)connectivity of a geometric object and to some extent provide “an analysis properly geometric or linear that expresses location directly as algebra expresses magnitude.”¹

Thesis: *Any global effect that depends on the object as a whole and that cannot be localized is of homological nature, and should be amenable to topological methods.*

WHERE HAS TOPOLOGY BEEN APPLIED IN COMPUTER SCIENCE?

The references [Car03] and [BEA+99] provide a broad overview of many current applications of algebraic topology in computer science and vice versa as well as an insight into promising new developments. The field is undergoing a rapid expansion and the following list should be understood as a sample of some of the main themes or aspects of potential future research.

- (a) Algebraic topology (AT) is viewed as a useful tool in solving combinatorial or discrete geometric problems of relevance to computing and the analysis of algorithms, [Mat02, Mat03, Živ98].
- (b) *Computational topology* emerges [BEA+99] as a separate branch of computational geometry unifying topological questions in computer applications such as image processing, cartography, computer graphics, solid modeling, mesh generation, and molecular modeling [BEA+99, DEG99].
- (c) Effective algebraic topology deals with algorithmic and computational aspects of topology including the recognition problem (3-manifolds), effective computations of topological invariants (homology, homotopy groups, knot invariants), etc. [Dun, Ser].
- (d) Combinatorial proofs of statements originally obtained by nonconstructive topological methods were discovered [Mat, Zie02].
- (e) The methods of AT can provide qualitative and shape information unavailable by the use of other methods. For example AT provides a tool for visualization

¹A dream of G.W. Leibniz expressed in a letter to C. Huygens dated 1697; see [Bre93, Chap. 7].

and feature identification in highly complex empirical data, e.g., in biogeometry [BioG].

- (f) AT provides a useful framework for analyzing problems in distributed and concurrent computing [HR95, HR00].

HOW IS TOPOLOGY APPLIED IN DISCRETE GEOMETRIC PROBLEMS?

In this chapter we put some emphasis on the role of (equivariant) topological methods in solving combinatorial or discrete geometric problems that have proven to be of relevance for computational geometry and computational mathematics in general. The versatile *configuration space/test map* scheme was developed in numerous research papers over the years and formally codified in [Živ98]. Its essential features are the following two steps:

Step 1: The problem is rephrased in topological terms.

The problem should give us a clue how to define a “natural” *configuration space* X and how to rephrase the question in terms of zeros or coincidences of the associated *test maps*. Alternatively the problem may be divided into several subproblems, in which case one is often led to the question of when the subsets of X corresponding to the various subproblems have *nonempty intersection*.

Step 2: Standard topological techniques are used to solve the rephrased problem.

The topological technique that is most frequently used in discrete geometric problems is based on the technique of *intersecting homology classes* and on *generalized Borsuk-Ulam theorems*.

14.1 THE CONFIGURATION SPACE/TEST MAP PARADIGM

GLOSSARY

Configuration space/test map scheme (CS/TM): A very useful and general scheme for proving combinatorial or geometric facts. The problem is reduced to the question of showing that there does not exist a G -equivariant map $f : X \rightarrow V \setminus Z$ (Section 14.5) where X is the configuration space, V the test space, and Z the test subspace associated with the problem, while G is a naturally arising group of symmetries.

Configuration space: In general, any topological space X that parameterizes a class of configurations of geometric objects (e.g., arrangements of points, lines, fans, flags, etc.) or combinatorial structures (trees, graphs, partitions, etc.). Given a problem \mathcal{P} , an associated configuration or **candidate space** $X_{\mathcal{P}}$ collects all geometric configurations that are (reasonable) candidates for a solution of \mathcal{P} .

Test map and test space : A map $t : X_{\mathcal{P}} \rightarrow V$ from the configuration space $X_{\mathcal{P}}$ into the so-called test space V that tests the validity of a candidate $p \in X_{\mathcal{P}}$ as

a solution of \mathcal{P} . The final ingredient is the *test subspace* $Z \subset V$, where $p \in X$ is a solution to the problem if and only if $t(p) \in Z$. Usually $V \cong \mathbb{R}^d$ while Z is just the origin $\{0\} \subset V$ or more generally a linear subspace arrangement in V .

Equivariant map: The final ingredient in the CS/TM-scheme is a group G of symmetries that acts on both the configuration space $X_{\mathcal{P}}$ and the test space V (keeping the test subspace Z invariant). The test map t is always assumed G -equivariant, i.e., $t(g \cdot x) = g \cdot t(x)$ for each $g \in G$ and $x \in X_{\mathcal{P}}$. Some of the methods and tools of equivariant topology are outlined in Section 14.5.

EXAMPLE 14.1.1 (Y. Soibelman [Soi02])

Suppose that ρ is a metric on \mathbb{R}^2 that induces the same topology as the usual Euclidean metric. In other words we assume that for each sequence of points $(x_n)_{n \geq 0}$, $\rho(x_n, x_0) \rightarrow 0$ if and only if $|x_n - x_0| \rightarrow 0$. Then there exists a ρ -equilateral triangle, i.e., a triple (a, b, c) of distinct points in \mathbb{R}^2 such that $\rho(a, b) = \rho(b, c) = \rho(c, a)$.

This is our first example that illustrates the CS/TM-scheme. The configuration space X should collect all candidates for the solution, so a first, “naive” choice is the space of all (ordered) triples $(x, y, z) \in \mathbb{R}^2$. Of course we can immediately rule out some obvious nonsolutions, e.g., degenerate triangles (x, y, z) such that at least one of numbers $\rho(x, y), \rho(y, z), \rho(z, x)$ is zero. (This illustrates the fact that in general there may be several possible choices for a configuration space associated to the initial problem.) Our choice is $X := (\mathbb{R}^2)^3 \setminus \Delta$ where $\Delta := \{(x, x, x) \mid x \in \mathbb{R}^2\}$. A “triangle” $(x, y, z) \in X$ is ρ -equilateral if and only if $(\rho(x, y), \rho(y, z), \rho(z, x)) \in Z$, where $Z := \{(u, u, u) \in \mathbb{R}^3 \mid u \in \mathbb{R}\}$. Hence a test map $t : X \rightarrow \mathbb{R}^3$ is defined by $t(x, y, z) = (\rho(x, y), \rho(y, z), \rho(z, x))$, the test space is $V = \mathbb{R}^3$, and $Z \subset \mathbb{R}^3$ is the associated test subspace. A triangle $\{x, y, z\}$, viewed as a set of vertices, is in general labeled by six different triples in the configuration space X . This redundancy is a motivation for introducing the group of symmetries $G = S_3$, which acts on both the configuration space X and the test space V . The test map t is clearly S_3 -equivariant. If the image of t is disjoint from Z , there arises an S_3 -equivariant map from X to $V \setminus Z$. If S^1 is the unit circle in a 2-plane in $V = \mathbb{R}^3$ orthogonal to $Z \cong \mathbb{R}^1$, then projection and normalization give an S_3 -equivariant map $\alpha : V \setminus Z \rightarrow S^1$. The unit 3-sphere S^3 in a 4-plane orthogonal to Δ is S_3 -invariant, hence the inclusion map $\beta : S^3 \rightarrow X$ is S_3 -equivariant. Finally, the composition $f := \alpha \circ t \circ \beta : S^3 \rightarrow S^1$ is also S_3 -equivariant, hence \mathbb{Z}_3 -equivariant, which leads to a contradiction. One way to prove this is to use Theorem 14.5.1, since the sphere S^3 is clearly 1-connected and the action of \mathbb{Z}_3 on S^3 is free.

Here is another example of how topology comes into play and proves useful in geometric and combinatorial problems. The *configuration space* associated to the next problem is a 2-dimensional torus $T^2 \cong S^1 \times S^1$. This time, however, the test map is not explicitly given. Instead, the problem is reduced to counting intersection points of two “test subspaces” in T^2 .

EXAMPLE 14.1.2 A watch with two equal hands

A watch was manufactured with a defect so that both hands (minute and hour) are identical. Otherwise the watch works well and the question is to determine the number of ambiguous positions, i.e., the positions for which it is not possible to determine the exact time.

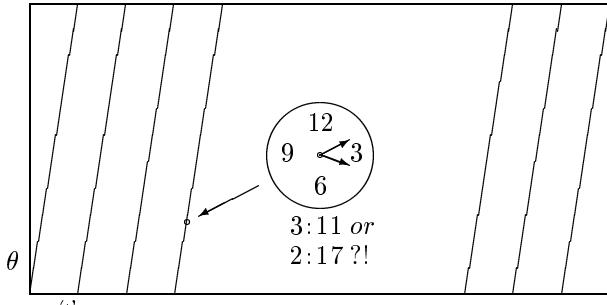


FIGURE 14.1.1
The configuration space of the two hands is a torus.

First of all we observe that every position of a hand is determined by an angle $\omega \in [0, 2\pi]$, so that the configuration space of all possible positions of a hand is homeomorphic to the unit circle S^1 . Two independent hands have the 2-dimensional torus $T^2 \cong S^1 \times S^1$ as their configuration space, i.e., the space representing all allowed states or positions of the system. A usual model of a torus is a square or a rectangle (see Figure 14.1.1) with the opposite sides glued together. If θ corresponds to the minute hand and ω is the coordinate of the hour hand, then the fact that the first hand is twelve times faster is recorded by the equation $\theta = 12\omega$. This equation describes a curve Γ_1 on the torus T^2 , which is just a circle winding 12 times in the direction of the θ axis while it winds only once in the direction of ω axis. The curve Γ_1 is represented in our picture as the union of 12 line segments, seven of them indicated in Figure 14.1.1. If the hands change places then the corresponding curve Γ_2 has equation $\omega = 12\theta$. The ambiguous positions are exactly the intersection points of these two curves (except those that belong to the diagonal $\Delta := \{(\theta, \omega) | \theta = \omega\}$, when it is still possible to tell the exact time without knowing which hand is for hours and which for minutes). The reader can now easily find the number of these intersection points and compute that there are 143 of them in the intersection $\Gamma_1 \cap \Gamma_2$, and 11 in the intersection $\Gamma_1 \cap \Gamma_2 \cap \Delta$, which shows that there are all together 132 ambiguous positions.

REMARK 14.1.3

Let us note that the “watch with equal hands” problem reduces to counting points or 0-dimensional manifolds in the intersection of two circles, viewed as 1-dimensional submanifolds of the 2-dimensional manifold T^2 . More generally, one may be interested in how many points there are in the intersection of two or more submanifolds of a higher-dimensional ambient manifold. Topology gives us a versatile tool for computing this and much more, in terms of the so-called *intersection product* $\alpha \frown \beta$ of homology classes α and β in a manifold M . This intersection product is, via Poincaré duality, equivalent to the “cup” product, and has the usual properties [Mun84]. In our Example 14.1.2, keeping in mind that $a \frown b = -b \frown a$ for all 1-dimensional classes, and in particular that $a \frown a = 0$ if $\dim(a) = 1$, we have $[\Gamma_1] \frown [\Gamma_2] = ([\theta] + 12[\omega]) \frown ([\omega] + 12[\theta]) = [\theta] \frown [\omega] + 12[\omega] \frown [\omega] + 12[\theta] \frown [\theta] + 144[\omega] \frown [\theta] = 143[\omega] \frown [\theta]$ and, taking the orientation into account, we conclude that the number of intersection points is 143.

14.2 PARTITIONS OF MASS DISTRIBUTIONS

Problems of partitioning mass distributions in the plane, 3-space, or spaces of higher dimension form the first circle of discrete geometric problems where topological methods have traditionally been applied with great success.

An (open) ham sandwich is a collection of three measurable sets in \mathbb{R}^3 , representing a slice of bread, a slice of ham, and a slice of cheese. It turns out that there always exists a plane simultaneously halving all three measurable sets or, in other words, that a ham sandwich can be cut fairly into two pieces by a single straight cut. Suppose, on the other hand, that you want to split an irregularly shaped slice of pizza with a hungry friend who is supposed to divide the pizza into two pieces by a straight knife-cut, but who can cut anywhere he likes. You are allowed to mark your piece in advance by specifying a single point that will lie in your piece. Then, if you are very careful about marking your piece, you can count on at least one third of the pizza. These two results are instances of *the ham sandwich theorem* and *the center point theorem* which, together with their relatives, often require topological methods in their proofs.

GLOSSARY

Measure: An abstract function μ defined on a class of sets that has all the formal properties (additivity, positivity) of the usual *volume* or *area* functions.

Measurable set: Any set in the domain of the function μ .

Mass distribution and density function: A density function is an integrable function $f : \mathbb{R}^d \rightarrow [0, +\infty)$ representing the density of a “mass distribution” (measure) on \mathbb{R}^d . The measure μ arising this way is defined by $\mu(A) := \int_A f dx$.

Halving hyperplane: A hyperplane that simultaneously bisects a family of measurable sets.

Grassmann and Stiefel manifolds: The Grassmann manifold $G_k(\mathbb{R}^n)$ of all k -dimensional linear subspaces of \mathbb{R}^n and the Stiefel manifold $V_k(\mathbb{R}^n)$ of all orthonormal k -frames in \mathbb{R}^n are frequently used in the construction of configuration spaces associated to measure partitioning problems.

14.2.1 THE HAM SANDWICH THEOREM

Given a collection of d measurable sets (mass distributions, finite sets) in \mathbb{R}^d , the problem is to simultaneously bisect all of them by a single hyperplane. Often a measurable set is a geometric object $A \subset \mathbb{R}^d$, say a polytope, whose measure is simply its volume $\text{vol } A$. More generally, a measurable set A is an arbitrary subset of \mathbb{R}^d if it is clear from the context what we mean by its “measure” $\mu(A)$. Typically, A is a Lebesgue-measurable set and $\mu(A) = m(A)$ its Lebesgue measure which, in the usual cases, reduces to the measure vol described above. More generally, if $f : \mathbb{R}^d \rightarrow \mathbb{R}^+$ is an integrable density function, then $\mu(A) := \int_A f dm = \int_{\mathbb{R}^d} f \phi_A dm$ is the measure or the mass distribution associated with the function f , where ϕ_A is the characteristic function of A (1 on A , 0 otherwise). An important special

case arises if $f = \phi_B$ for a Lebesgue-measurable set B , where $\mu(A) = m(A \cap B)$. Finally, if $S \subset \mathbb{R}^d$ is a finite set, then $\mu(A) := |A \cap S|$ is the so-called **counting measure** induced by the set S . All of these examples are subsumed by the case of a positive, σ -additive Borel measure μ . This means that μ is defined on a σ -algebra \mathcal{F} of subsets of \mathbb{R}^d that includes all closed halfspaces and other sets that arise naturally in geometric problems. The reader should, in principle, not have any difficulty reformulating any of the following results for whatever special class of measures she may be interested in.

THEOREM 14.2.1 *Ham Sandwich Theorem* [Bor33]

Let $\mu_1, \mu_2, \dots, \mu_d$ be a collection of measures (mass distributions, measurable sets, finite sets) in the sense above. Then there exists a hyperplane H such that for all $i = 1, \dots, d$, $\mu_i(H^+) \geq 1/2 \mu_i(\mathbb{R}^d)$ and $\mu_i(H^-) \geq 1/2 \mu_i(\mathbb{R}^d)$, where H^+ and H^- are the closed halfspaces associated with the hyperplane H .

In the special case where $\mu(H) = 0$, i.e., where the hyperplane itself has measure zero, H is called a **halving hyperplane** since $\mu_i(H^+) = \mu_i(H^-) = 1/2 \mu_i(\mathbb{R}^d)$ for all i . A halving hyperplane H is also called a “ham sandwich cut,” for the reasons noted above.

TOPOLOGICAL BACKGROUND

The topological result lying behind the ham sandwich theorem is the Borsuk-Ulam theorem, [Ste85, Mat03]. The proof of the ham sandwich theorem historically marks one of the first applications of the CS/TM-scheme, with the $(d-1)$ -sphere as the configuration space, \mathbb{R}^d as the test space, and $G = \mathbb{Z}_2$ as the group of symmetries associated to the problem. Given a collection $\{A_i\}_{i=1}^d$ of d measurable sets, the test map $t : S^{d-1} \rightarrow \mathbb{R}^d$ is defined by $t(e) = (\alpha_1, \dots, \alpha_d)$, with α_i determined by the condition that $H_i := \{x \in \mathbb{R}^d \mid \langle x, e \rangle = \alpha_i\}$ is a median halving hyperplane for the measurable set A_i . (The median halving hyperplane in any direction is the mid-hyperplane between the two extreme halving hyperplanes in that direction.) The test space is the diagonal $Z := \{(\alpha, \dots, \alpha) \in \mathbb{R}^d \mid \alpha \in \mathbb{R}\}$. The test map t is obviously “odd,” or \mathbb{Z}_2 -equivariant, in the sense that $t(-e) = -t(e)$.

THEOREM 14.2.2 *Borsuk-Ulam Theorem* [Bor33]

For every continuous map $f : S^n \rightarrow \mathbb{R}^n$ from an n -dimensional sphere into n -dimensional Euclidean space, there exists a point $x \in S^n$ such that $f(x) = f(-x)$.

An important special case of the Borsuk-Ulam theorem arises if f is an odd map. The conclusion is that a continuous odd map must have a zero on the sphere, i.e., $f(x) = 0$ for some $x \in S^d$. This is precisely the reason why the test map t for the ham sandwich theorem has the property $t(e) \in Z$ for some $e \in S^{d-1}$. Note that the general Borsuk-Ulam theorem follows from the special case if the latter is applied to the map $\phi : S^d \rightarrow \mathbb{R}^d$ given by $\phi(x) := f(x) - f(-x)$.

There is a different topological approach to the ham sandwich theorem closer to the earlier example about a watch with two indistinguishable hands. Here we mention only that the role of the torus T^2 is played by a manifold M representing all hyperplanes in \mathbb{R}^d (the configuration space), while the curves Γ_1 and Γ_2 are replaced by suitable submanifolds N_i of M , one for each of the measures μ_i , $i = 1, \dots, d$. N_i is defined as the space of all halving hyperplanes for the measurable set A_i .

APPLICATIONS AND RELATED RESULTS

Let S_1, \dots, S_d be a collection of finite sets, called “colors,” in \mathbb{R}^d . Assume that the size of each of these sets is n and that the points are all in general position. Then, according to Akiyama and Alon [AA89], the ham sandwich theorem implies that there exists a partition of $S := \bigcup_{i=1}^d S_i$ into n nonempty, pairwise disjoint sets D_1, \dots, D_n that are multicolored in the sense that $|D_i \cap S_j| = 1$ for all i and j , such that the simplices $\text{conv } D_1, \dots, \text{conv } D_n$ are pairwise disjoint.

14.2.2 THE CENTER POINT THEOREM

THEOREM 14.2.3 *Center Point Theorem* [Rad46]

Let $A \subset \mathbb{R}^d$ be a Lebesgue-measurable subset of \mathbb{R}^d or, more generally, one of the measures μ described prior to Theorem 14.2.1. Then there exists a point $x \in \mathbb{R}^d$ such that for every closed halfspace $P \subset \mathbb{R}^d$, if $x \in P$ then

$$\text{vol}(P \cap A) \geq \frac{\text{vol}(A)}{d+1}.$$

When formulated for a more general measure μ , the result guarantees that $\mu(P) \geq \mu(\mathbb{R}^d)/(d+1)$ for every closed halfspace $P \ni x$.

TOPOLOGICAL BACKGROUND

If the Borsuk-Ulam theorem is responsible for the ham sandwich theorem, then R. Rado’s center point theorem can be seen as a consequence of another well-known topological result, Brouwer’s fixed point theorem. Note that the usual formulation about self-maps $f : K \rightarrow K$ generalizes easily to the following formulation.

THEOREM 14.2.4 *Brouwer’s Fixed Point Theorem* [Bro75, Kak41]

Let K be a compact, convex body in \mathbb{R}^n . Suppose $f : K \rightarrow \mathbb{R}^n$ is a continuous map such that for each $x \in K$ the image $f(x)$ belongs to the supporting cone of K at x , $\text{cone}_x(K) := \bigcup_{\lambda \geq 0} (x + \lambda(K - x))$. Then $f(x) = x$ for some $x \in K$.

Very often it is more convenient to use Kakutani’s theorem, which is a generalization of Brouwer’s theorem to “multivalued functions” $f : B \rightarrow \mathbb{R}^n$.

The center point theorem is deduced from Brouwer’s theorem roughly as follows. Let $x \in B$, where B is a “large” ball containing the set A . If x is not a center point, then there is a vector $e \in S^{d-1}$ pointing in a direction in which x can be moved to make it closer to being one. In this way we define a function $x \mapsto f(x)$, and a fixed point, i.e., a point that doesn’t need to be moved, is a center point.

Recall that the center point theorem was originally deduced (by R. Rado) from Helly’s theorem about intersecting families of convex sets, which also has several topological relatives. For this reason, it is often viewed as a measure-theoretic equivalent of Helly’s theorem.

APPLICATIONS AND RELATED RESULTS

As noted by Miller and Thurston (see [MTTV97, MTTV98]), the center point theorem and the Koebe theorem on the disk representation of planar graphs can be used to prove the existence of a small separator for a planar graph, a result proved originally (by Lipton and Tarjan) by different methods.

The *regression depth* $\text{rd}_{\mathcal{P}}(H)$ of a hyperplane H relative to a collection \mathcal{P} of n points in \mathbb{R}^d is the minimum number of points that H must pass through in moving to the vertical position. Dually, given an arrangement \mathcal{H} of n hyperplanes in \mathbb{R}^d , the regression depth $\text{rd}_{\mathcal{H}}(x)$ of a point x relative to \mathcal{H} is the smallest k such that x cannot escape to infinity without crossing (or moving parallel to) at least k hyperplanes. The problem of finding a point (resp. hyperplane) with maximum regression depth relative to \mathcal{H} (resp. \mathcal{P}) is shown in [AET00] to be intimately connected with the problem of finding center points. The main result (confirming a conjecture of Rousseeuw and Hubert) is that there always exists a point with regression depth $\lceil n/(d+1) \rceil$; cf. [Chapter 57](#) of this Handbook.

14.2.3 CENTER TRANSVERSAL THEOREM

THEOREM 14.2.5 *Center Transversal Theorem* [ZV90]

Let A_0, A_1, \dots, A_k , $0 \leq k \leq d-1$, be a collection of Lebesgue-measurable sets in \mathbb{R}^d or, more generally, let $\mu_0, \mu_1, \dots, \mu_k$ be a sequence of measures. Then there exists a k -dimensional affine subspace $D \subset \mathbb{R}^d$ such that for every closed halfspace $H(v, \alpha) := \{x \in \mathbb{R}^d \mid \langle x, v \rangle \leq \alpha\}$ and every $i \in \{0, 1, \dots, k\}$,

$$D \subset H(v, \alpha) \implies m(A_i \cap H(v, \alpha)) \geq \frac{m(A_i)}{d-k+1}.$$

If formulated for a sequence μ_0, \dots, μ_k of more general measures, the result guarantees that $\mu_i(H(v, \alpha)) \geq \mu_i(\mathbb{R}^d)/(d-k+1)$ for all i and all $H(v, \alpha) \supseteq D$.

TOPOLOGICAL BACKGROUND

The center transversal theorem contains the ham sandwich and center point theorems as two boundary cases [ZV90]. The topological principle that is at the root of this result should be strong enough for this purpose. This result has several incarnations. One of them, in the language of the CS/TM-scheme, is a theorem of E. Fadell and S. Husseini [FH88] that claims the nonexistence of a $\mathbb{Z}_2^{\oplus k}$ -equivariant map $f : V_{n,k} \rightarrow (\mathbb{R}^k)^{n-k} \setminus \{0\}$ from the Stiefel manifold of all orthonormal k -frames in \mathbb{R}^n to the sum of $n-k$ copies of \mathbb{R}^k . The group $\mathbb{Z}_2^{\oplus k}$ can be identified with the group of all diagonal matrices in $SO(k)$ and its action on \mathbb{R}^k is induced by the obvious action of $SO(k)$. A related result [FH88, ZV90] is that the vector bundle $\xi_k^{\oplus(n-k)}$ does not admit a nonzero, continuous cross-section, where ξ_k is the tautological k -plane bundle over the Grassmann manifold $G_k(\mathbb{R}^n)$.

APPLICATIONS AND RELATED RESULTS

The following Helly-type transversal theorem, due to Dol'nikov, is a consequence of the same topological principle that is at the root of the center transversal theorem. Moreover, the center transversal theorem is related to Dol'nikov's result in the same way that the center point theorem is related to Helly's theorem.

THEOREM 14.2.6 [Dol'93]

Let $\mathcal{K}_0, \dots, \mathcal{K}_k$ be families of compact convex sets. Suppose that for every i , and for each k -dimensional subspace $V \subset \mathbb{R}^d$, there exists a translate V_i of V intersecting every set in \mathcal{K}_i . Then there exists a common k -dimensional transversal of the family $\mathcal{K} := \bigcup_{i=0}^k \mathcal{K}_i$, i.e., there exists an affine k -dimensional subspace of \mathbb{R}^d intersecting all the sets in \mathcal{K} .

Let $\mathcal{K} = \{K_0, \dots, K_k\}$ be a family of convex bodies in \mathbb{R}^n , $1 \leq k \leq n - 1$. Then an affine l -plane $A \subset \mathbb{R}^n$ is called a **common maximal l -transversal** of \mathcal{K} if $m(K_i \cap A) \geq m(K_i \cap (A + x))$ for each $i \in \{0, \dots, k\}$ and each $x \in \mathbb{R}^n$, where m is l -dimensional Lebesgue measure in A and $A + x$, respectively. It was shown in [MVZ01] that, given a family $\mathcal{K} = \{K_i\}_{i=0}^k$ of convex bodies in \mathbb{R}^n ($k < l$), the set $C_l(\mathcal{K})$ of all common maximal l -transversals of \mathcal{K} has to be “large” from both the measure-theoretic and the topological point of view. Here again one uses the same topological principle responsible for all results in this section together with some integral geometry calculations to show that a cohomologically “big” subspace of the Grassmann manifold $G_k(\mathbb{R}^n)$ has to be large also in a measure-theoretic sense.

14.2.4 EQUIPARTITION OF MASSES BY HYPERPLANES

Every measurable set $A \subset \mathbb{R}^3$ can be partitioned by three planes into 8 pieces of equal measure. This is an instance of the general problem of characterizing all triples (d, j, k) such that for any j mass distributions (measurable sets) in \mathbb{R}^d , there exist k hyperplanes, $k \leq d$, such that each of the 2^k “orthants” contains the fraction $1/2^k$ of each of the masses. Such a triple (d, j, k) will be called *admissible*. For example, the ham sandwich theorem implies that $(d, d, 1)$ is admissible. It is known (E. Ramos, [Ram96]) that $d \geq j(2^k - 1)/k$ is a necessary condition and $d \geq j2^{k-1}$ a sufficient one for a triple (d, j, k) to be admissible. Ramos's method yields many interesting results in lower dimensions, including the admissibility of the triples $(9, 3, 3)$, $(9, 5, 2)$, and $(5, 1, 4)$. The most interesting special case that still seems to be out of reach is the triple $(4, 1, 4)$. The key idea in these proofs is to use, for this purpose, a specially designed, generalized form of the Borsuk-Ulam theorem for continuous, “even-odd” maps of the form $f : S^{d-1} \times \dots \times S^{d-1} \rightarrow \mathbb{R}^l$.

APPLICATIONS AND RELATED RESULTS

According to [Mat03], an early interest of computer scientists in partitioning mass distributions by hyperplanes was stimulated in part by *geometric range searching*; cf. [Chapter 36](#) of this Handbook. As noted by Matoušek, the classical mass partitioning results were eventually superseded by random sampling and related results. However, one still wonders about the possible impact of a positive answer to the

following conjecture (a special case of the conjecture that $(4, 1, 4)$ is admissible) to the construction and complexity of geometric algorithms.

CONJECTURE 14.2.7

For each collection of 16 distinct points A_1, \dots, A_{16} in \mathbb{R}^4 , there exist 4 hyperplanes H_1, \dots, H_4 such that each of the associated 16 open orthants contains at most one of the given points.

It is known that the answer to the conjecture is positive if the points are distributed along a **convex curve** in \mathbb{R}^4 (a curve in \mathbb{R}^m is convex if, like the moment curve, it intersects each hyperplane in at most m distinct points). This special case of the conjecture follows [Ram96] from the existence of uniform Gray codes on 4-dimensional cubes [Knu]. Recall that a uniform Gray code on a k -dimensional cube is a Hamiltonian circuit on the graph of all edges of the cube that is balanced in the sense that it uses the same number of edges from each of k parallel classes.

14.2.5 RADIAL PARTITIONS BY POLYHEDRAL FANS

An old result of R. Buck and E. Buck [BB49] says that for each continuous mass distribution in the plane, there exist three concurrent lines $l_1, l_2, l_3 \subset \mathbb{R}^2$ that partition \mathbb{R}^2 into six sectors of equal measure. It is natural to search for higher dimensional analogs of this result.

Suppose that $Q \subset \mathbb{R}^d$ is a convex polytope and assume that the origin $O \in \mathbb{R}^d$ belongs to the interior $\text{int}(Q)$ of Q . Let $\{F_i\}_{i=1}^k$ be the collection of all facets of Q . Let $\mathcal{F} := \text{fan}(Q)$ be the associated **fan**, i.e., $\mathcal{F} = \{C_1, \dots, C_k\}$ where $C_i = \text{cone}(F_i)$ is the convex closed cone with vertex O generated by F_i .

THEOREM 14.2.8 [Mak01]

Let Q be a regular dodecahedron with the origin $O \in \mathbb{R}^3$ as its barycenter. Then for any centrally symmetric, continuous mass distribution μ on \mathbb{R}^3 , there exists a linear map $L \in GL(3, \mathbb{R})$ such that

$$\mu(L(C_1)) = \mu(L(C_2)) = \dots = \mu(L(C_k)).$$

Makeev actually shows in [Mak01] that L can be found in the set of all matrices of the form $a \cdot t$, where t is an upper triangular matrix and $a \in GL(3, \mathbb{R})$ is a matrix given in advance. In an earlier paper (see [Mak98]) he showed that a radial partition by a fan determined by the facets of a cube always exists for an arbitrary measure in \mathbb{R}^3 . Moreover, he shows in [Mak01] that a result analogous to Theorem 14.2.8 also holds for rhombic dodecahedra. Recall that the rhombic dodecahedron U_3 is the polytope bounded by twelve planes, each containing an edge of a cube and parallel to one of the great diagonal planes. A higher dimensional analogue of the rhombic dodecahedron is the polytope U_n in \mathbb{R}^n described as the dual of the difference body of a regular simplex.

PROBLEM 14.2.9

Let $T \subset \mathbb{R}^n$ be a regular simplex and $Q := T - T$ the associated “difference polytope.” Let $U_n := Q^\circ$ be the polytope polar to Q . Clearly U_n is a centrally symmetric polytope with $n^2 + n$ facets F_i , $i = 1, \dots, n^2 + n$. Let $\{K_i\}_{i=1}^{n^2+n}$ be the associated conical dissection of \mathbb{R}^n , where $K_i := \text{cone}(F_i)$. Is it true that for any continuous

mass distribution μ on \mathbb{R}^n there exists a nondegenerate affine map $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$\mu(A(K_1)) = \mu(A(K_2)) = \dots = \mu(A(K_{n^2+n}))?$$

The following result of Vrećica and Živaljević is an example of a radial partition result for a single measure in \mathbb{R}^n with ratios prescribed by a positive vector α .

THEOREM 14.2.10 [VZ01]

Let $\Delta \subset \mathbb{R}^n$ be a nondegenerate simplex with $O \in \text{int}(\Delta)$. Suppose that μ is a continuous mass distribution on \mathbb{R}^n , and let $\alpha = (\alpha_0, \dots, \alpha_n)$ be a given positive vector such that $\alpha_0 + \dots + \alpha_n = 1$. Then there exists a vector $v \in \mathbb{R}^n$ such that $\mu(v + K_i) = \alpha_i \mu(\mathbb{R}^n)$ for each $i = 0, \dots, n$, where $\mathcal{F} = \text{fan}(\Delta) = \{K_i\}_{i=0}^n$ is the radial fan associated to Δ .

14.2.6 EQUIPARTITIONS BY WEDGEGLIKE CONES

The center transversal theorem is a common generalization of the ham sandwich theorem and the center point theorem. There is another general statement extending the ham sandwich theorem that, as a special boundary case, includes the equipartition case of Theorem 14.2.10.

Definition: Let $\Delta := \text{conv}(\{a_i\}_{i=0}^m)$ be a regular simplex of dimension $m \leq d$ and let $P := \text{aff } \Delta$ be its affine hull. Then $\mathcal{D}(\Delta) = \{D_i\}_{i=0}^m$ represents the dissection of \mathbb{R}^d into $m+1$ wedgelike cones, where $D_i := P^\perp \oplus \text{cone}(\text{conv}(\{a_j\}_{j \neq i}))$.

CONJECTURE 14.2.11

Let μ_0, \dots, μ_k be a family of continuous mass distributions (measures), $0 \leq k \leq d-1$, defined on \mathbb{R}^d . Then there exists a $(d-k)$ -dimensional regular simplex Δ such that for the corresponding dissection, $\mathcal{D}(\Delta)$, for some $x \in \mathbb{R}^d$, and for all i, j ,

$$\mu_i(x + D_j) \geq \frac{\mu_i(\mathbb{R}^d)}{d - k + 1}.$$

This conjecture is denoted in [VZ92] by $B(d, k)$. Theorem 14.2.10 implies $B(d, 0)$, and the ham sandwich theorem is $B(d, d-1)$. The conjecture is also confirmed in the case $B(d, d-2)$ for all d . Moreover, there exists a natural topological conjecture implying $B(d, k)$ that is closely related to the analogous statement needed for the center transversal theorem. This statement, denoted in [VZ92] by $C(d, k)$, in the spirit of the CS/TM-scheme, essentially claims that there is no \mathbb{Z}_{k+1} -equivariant map from the Stiefel manifold $V_k(\mathbb{R}^n)$ to the unit sphere $S(V)$ in an appropriate \mathbb{Z}_{k+1} -representation V .

14.2.7 PARTITIONS BY CONVEX SETS

CONJECTURE 14.2.12

Let n and d be integers with $n, d \geq 2$. Assume that μ_1, \dots, μ_d are continuous mass distributions such that $\mu_1(\mathbb{R}^d) = \dots = \mu_d(\mathbb{R}^d) = n$. Then there exists a partition of

\mathbb{R}^d into n sets C_1, \dots, C_n such that the interiors $\text{int}(C_i)$ are convex sets and that $\mu_i(C_i) = 1$ for each $i = 1, \dots, n$.

This conjecture was formulated in [KK99] by A. Kaneko and M. Kano for the case $d = 2$. Kaneko and Kano originally formulated the conjecture for finite sets rather than for continuous mass distributions, but this is not essential. Note that the case $n = 2$ is true by the ham sandwich theorem. The case $d = 2$ was independently established by S. Bespamyatnikh, D. Kirkpatrick, and J. Snoeyink, by T. Sakai, and by H. Ito, H. Uehara, and M. Yokoyama; see [BM01] for additional information.

14.2.8 PARTITIONS BY k -FANS IN PRESCRIBED RATIOS

The conjecture of Kaneko and Kano (the case $d = 2, n = 3$) motivated I. Bárány and J. Matoušek in [BM01, BM02] to study general conical partitions of planar or spherical measures in prescribed ratios. We assume, in the following statements, that all measures are continuous mass distributions.

An arrangement of k semilines in the Euclidean (projective) plane or on the 2-sphere is called a k -fan if all semilines start from the same point. A k -fan is an α -partition for a probability measure μ if $\mu(\sigma_i) = \alpha_i$ for each $i = 1, \dots, k$, where $\{\sigma_i\}_{i=1}^k$ are conical sectors associated with the k -fan and $\alpha = (\alpha_1, \dots, \alpha_k)$ is a given vector. The set of all $\alpha = (\alpha_1, \dots, \alpha_m)$ such that for any collection of probability measures μ_1, \dots, μ_m there exists a common α -partition by a k -fan is denoted by $\mathcal{A}_{m,k}$. It was shown in [BM01] that the interesting cases of the problem of existence of α -partitions are $(k, m) = (2, 3), (3, 2), (4, 2)$.

CONJECTURE 14.2.13

Suppose that (k, m) is equal to $(2, 3), (3, 2)$ or $(4, 2)$. Then $\alpha \in \mathcal{A}_{k,m}$ if and only if

$$\alpha_1 + \dots + \alpha_m = 1 \quad \text{and} \quad \alpha_i > 0 \quad \text{for each } i = 1, \dots, m.$$

The only known elements in $\mathcal{A}_{4,2}$ are, up to a permutation of coordinates, $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ and $(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{2}{5})$. They were discovered by Bárány and Matoušek by an ingenious application of the CS/TM scheme [BM01, BM02]. From this Bárány and Matoušek deduced that $\{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), (\frac{1}{2}, \frac{1}{4}, \frac{1}{4})\} \cup \{(\frac{p}{5}, \frac{q}{5}, \frac{r}{5}) \mid p, q, r \in N^+, p + q + r = 5\} \subset \mathcal{A}_{3,2}$.

14.2.9 OTHER EQUIPARTITIONS

There are other types of partitions of mass distributions. A “cobweb partition theorem” of Schulman [Sch93] guarantees an equipartition of a plane mass distribution into 8 pieces by an arrangement of lines resembling a cobweb.

A result of Paterson (see [Mat03]) is an interesting example of a ham-sandwich-type theorem that deals with partitions of lines rather than of points. It says that for every set of lines in \mathbb{R}^3 , there exist 3 mutually perpendicular planes such that the interior of each of the resulting octants is intersected by no more than half of the lines.

14.3 THE PROBLEMS OF BORSUK AND KNASTER

The topological methods used in proofs of measure partition results are actually applicable to a much wider class of combinatorial and geometric problems. In fact quite different problems, which on the surface have very little in common (say one of them may be discrete and the other not), may actually lead to the same or closely related configuration spaces and test maps. This in turn implies that such problems both follow from the same general topological principle and that they could, despite appearances, be classified as “relatives.”

14.3.1 BORSUK'S PROBLEM

Borsuk's well-known problem [Bor33] about covering sets in \mathbb{R}^n with sets of smaller diameter was solved in the negative by J. Kahn and G. Kalai [KK93] who proved that the size of a minimal cover is exponential in n ; see Chapters 1 and 2 of this Handbook. This, however, gave a new impetus to the study of “Borsuk numbers” after the old exponential upper bounds suddenly became more plausible. This may be one of the reasons why results about “universal covers,” originally used for these estimates, have received new attention in the last few years.

The following result was proved originally by V. Makeev; see also [HMS02, Kup99]. Recall the rhombic dodecahedron U_3 , the polytope bounded by twelve rhombic facets, which appeared in Section 14.2.5.

THEOREM 14.3.1 [Mak98]

A rhombic dodecahedron of width 1 is a universal cover for all sets $S \subset \mathbb{R}^3$ of diameter 1. In other words, each set of diameter 1 in 3-space can be covered by a rhombic dodecahedron whose opposite faces are 1 unit apart.

Let $\Sigma \subset \mathbb{R}^n$ be a regular simplex of edge-length 1, with vertices v_1, \dots, v_{n+1} . Then the intersection of $n(n+1)/2$ parallel strips S_{ij} of width 1, where S_{ij} is bounded by the $(n-1)$ -planes orthogonal to the segment $[v_i, v_j]$ passing through the vertices v_i and v_j ($i < j$), is a higher dimensional analog of the rhombic dodecahedron. It is easy to see that this is just another description of the polytope U_n that we encountered in Problem 14.2.9.

CONJECTURE 14.3.2 Makeev's conjecture [Mak94]

The polytope U_n is a universal cover in \mathbb{R}^n . In other words, for each set $S \subset \mathbb{R}^n$ of diameter 1, there exists an isometry $I : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that $S \subset I(U_n)$.

The relevance of the Makeev conjecture for the general Borsuk problem is obvious since in low dimensions, $d = 2$ and $d = 3$, the solutions were based on the construction of suitable universal covers. (Note that the case $d = 4$ of the Borsuk partition problem is still open!) The following stronger conjecture is yet another example of a topological statement with potentially interesting consequences in discrete and computational geometry.

CONJECTURE 14.3.3 [HMS02]

Let $f : S^{n-1} \rightarrow \mathbb{R}$ be an odd function, and let $\Sigma_n \subset \mathbb{R}^n$ be a regular simplex of

edge-length 1, with vertices v_1, \dots, v_{n+1} . Then there exists an orthogonal linear map $A \in SO(n)$ such that the $n(n+1)/2$ hyperplanes H_{ij} , $1 \leq i < j \leq n+1$, are concurrent, where

$$H_{ij} := \{x \in \mathbb{R}^n \mid \langle x, A(v_j - v_i) \rangle = f(A(v_j - v_i))\}.$$

G. Kuperberg showed in [Kup99] that, unlike the cases $n = 2$ and $n = 3$, for $n \geq 4$ there is homologically an even number of isometries $I : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that $S \subset I(U_n)$ for a given set S of constant width. Kuperberg showed that the Makeev conjecture can be reduced (essentially in the spirit of the CS/TM-scheme) to the question of the existence of a Γ -equivariant map $f : SO(n) \rightarrow V \setminus \{0\}$, where Γ is a group of symmetries of the root system of type A_n and the test space V is an $n(n-1)/2$ -dimensional representation of Γ . The fact that such a map exists if and only if $n \geq 4$ may be an indication that the Makeev conjecture is false in higher dimensions.

14.3.2 KNASTER'S PROBLEM

Knaster's problem is one of the old conjectures of discrete geometry with a distinct topological flavor. The conjecture is now known to be false in general, but the problem remains open in many interesting special cases.

PROBLEM 14.3.4 *Knaster's problem* [Kna47]

Given a finite subset $S = \{s_1, \dots, s_k\} \subset S^n$ of the n -sphere, determine the conditions on k and n so that for each continuous map $f : S^n \rightarrow \mathbb{R}^m$ there will exist an isometry $O \in SO(n+1)$ with

$$f(O(s_1)) = f(O(s_2)) = \dots = f(O(s_k)).$$

Knaster originally conjectured that such an isometry O always exists if $k \leq n - m + 2$. Just as in the case of the Borsuk problem, the first counterexamples took a long time to appear. V. Makeev [Mak86, Mak90], and somewhat later K. Babenko and S. Bogatyj [BB89], showed that the condition $k \leq n - m + 2$ is not sufficient if the original set S is sufficiently “flat.” In [Che98], W. Chen constructed new counterexamples confirming that the (original) Knaster conjecture is false for all $n > m > 2$.

The fact that Knaster's conjecture is false in general does not rule out the possibility that for some special configurations $S \subset S^n$ the answer is still positive. The case where S is the set of vertices of a “big” regular simplex in S^n is of special interest since it directly generalizes the Borsuk-Ulam theorem.

Questions closely related to Knaster's conjecture are the problems of inscribing or circumscribing polyhedra to convex bodies in \mathbb{R}^n ; see [HMS02, Kup99]. G. Kuperberg observed that both the circumscription problem for constant-width bodies and Knaster's problem are special cases of the following problem.

PROBLEM 14.3.5 [Kup99]

Given a finite set T of points on S^{d-1} and a linear subspace L of the space of all functions from T to \mathbb{R}^n , decide if, for each continuous function $f : S^{d-1} \rightarrow \mathbb{R}^n$, there is an isometry O such that the restriction of $f \circ O$ to T is an element of L .

14.4 TVERBERG-TYPE THEOREMS AND THEIR APPLICATIONS

Every collection of seven points in the plane can be partitioned into three nonempty, disjoint subsets so that the corresponding convex hulls have a nonempty intersection. If we add two more points and color all the points with three colors so that each color is equally represented, then there exists a partition of this set of nine colored points into three multicolored three-point sets such that the corresponding multicolored triangles have a nonempty intersection. Something similar is possible in 3-space, but this time we need five points of each color in order to guarantee a partition of this kind. In short, given a constellation of five blue, five red, and five yellow stars in space, it is always possible to form three vertex-disjoint multicolored triangles with nonempty intersection. These are the simplest nontrivial cases of the Tverberg-type theorems, which, together with their consequences and most important applications, are shown in Figure 14.4.1.

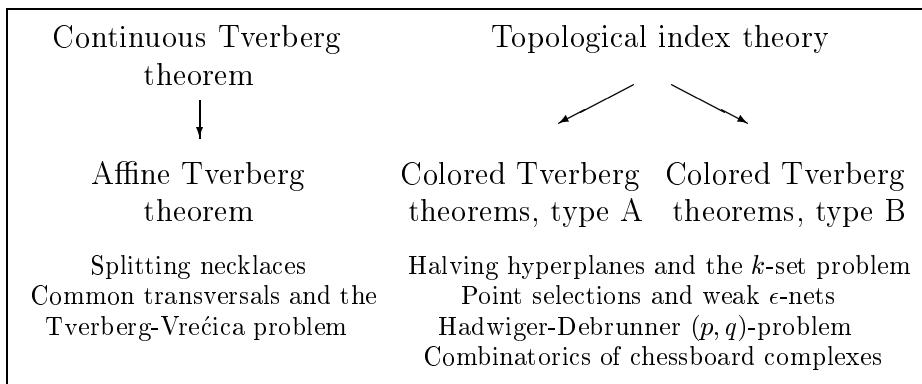


FIGURE 14.4.1
Tverberg-type theorems.

GLOSSARY

Tverberg-type problem: A problem in which a finite set $A \subset \mathbb{R}^d$ is to be partitioned into nonempty, disjoint pieces A_1, \dots, A_p , possibly subject to some constraints, so that the corresponding convex hulls $\{\text{conv}(A_i)\}_{i=1}^p$ intersect.

Colors: A set of $k+1$ colors is a collection $\mathcal{C} = \{C_0, \dots, C_k\}$ of disjoint subsets of \mathbb{R}^d , $d \geq k$. A set $B \subset \mathbb{R}^d$ is **multicolored** if it contains a point from each of the sets C_i ; in this case $\text{conv } B$ is called a **rainbow simplex** (possibly degenerate).

Type A and **Type B**: Colored Tverberg problems are of type A or type B depending on whether $k = d$ or $k < d$ (resp.), where $k+1$ is the number of colors.

Tverberg numbers $T(r, d)$, $T(r, k, d)$: $T(r, k, d)$ is the minimal size of each of the colors C_i , $i = 0, \dots, k$, that guarantees that there always exist r intersecting rainbow simplices. $T(r, d) := T(r, d, d)$.

14.4.1 MONOCHROMATIC TVERBERG THEOREMS

THEOREM 14.4.1 *Affine Tverberg Theorem [Tve66]*

Every set $K = \{a_j\}_{j=0}^{(q-1)(d+1)} \subset \mathbb{R}^d$ with $(d+1)(q-1)+1$ elements can be partitioned into q nonempty, disjoint subsets K_1, \dots, K_q so that the corresponding convex hulls have nonempty intersection:

$$\bigcap_{i=1}^q \text{conv}(K_i) \neq \emptyset.$$

(The special case $q = 2$ is Radon's theorem; see [Chapter 4](#).)

THEOREM 14.4.2 *Continuous Tverberg Theorem [BSS81]*

Let Δ^m be an m -dimensional simplex and assume that q is a prime integer. Then for every continuous map $f : \Delta^{(q-1)(d+1)} \rightarrow \mathbb{R}^d$ there exist vertex-disjoint faces $\Delta^{t_1}, \dots, \Delta^{t_q} \subset \Delta^{(q-1)(d+1)}$ such that $\bigcap_{i=1}^q f(\Delta^{t_i}) \neq \emptyset$.

APPLICATIONS AND RELATED RESULTS

The affine Tverberg theorem was proved by Helge Tverberg in 1966. The continuous Tverberg theorem, proved by Bárány, Shlosman, and Szűcs, reduces to the affine version if f is an affine (simplicial) map. It is not known if this result remains true for arbitrary q , although several authors have independently confirmed this if q is a prime power: see [[Živ98](#)] for a historical account. Some of the relevant references for these two theorems and their applications are [[Bár93](#), [Bjö95](#), [Sar92](#), [Eck93](#), [Vol96](#), [Živ98](#), [Mat02](#), [Mat03](#)].

The following “necklace-splitting theorem” of Noga Alon is a very nice application of the continuous Tverberg theorem.

THEOREM 14.4.3 [\[Alo87\]](#)

Assume that an open necklace has $k a_i$ beads of color i , $1 \leq i \leq t$, $k \geq 2$. Then it is possible to cut this necklace at $t(k-1)$ places and assemble the resulting intervals into k collections, each containing exactly a_i beads of color i .

REMARK 14.4.4

The proof of the necklace-splitting theorem provides a very nice example of an application of the CS/TM scheme (Section 14.1). A continuous model of a necklace is an interval $[0, 1]$ together with k measurable subsets A_1, \dots, A_k representing “beads” of different colors. It is well known that the configuration space of all sequences $0 \leq x_1 \leq \dots \leq x_m \leq 1$ is the m -dimensional simplex, hence the totality of all m -cuts of a necklace is identified with an m -dimensional simplex Σ . Given a cut $c \in \Sigma$, the assembling of the resulting subintervals $I_0(c), \dots, I_m(c)$ of $[0, 1]$ into k collections is determined by a function $f : [m+1] \rightarrow [k]$. Hence, a configuration space associated to the necklace-splitting problem is obtained by gluing together m -simplices Σ_f , one for each function $f \in \text{Fun}([m+1], [k])$. The complex $\mathcal{C}_{m,k}$ obtained by this construction turns out, in fact, to be a very important example

of a complex obtained from a simplex by a *deleted join operation*. The reader is referred to [Mat03] and [Živ98] for details about the role of (deleted) joins in combinatorics.

An interesting connection has emerged recently between ham-sandwich- and Tverberg-type problems. An example of this is the so-called Tverberg-Vrećica conjecture, which incorporates both the center transversal theorem (Theorem 14.2.5) and the (affine) Tverberg theorem in a single general statement.

CONJECTURE 14.4.5 [TV93]

Assume that $0 \leq k \leq d - 1$ and let S_0, S_1, \dots, S_k be a collection of finite sets in \mathbb{R}^d of given cardinalities $|S_i| = (r_i - 1)(d - k + 1) + 1$, $i = 0, 1, \dots, k$. Then S_i can be split into r_i nonempty sets, $S_i^1, \dots, S_i^{r_i}$, so that for some k -dimensional affine subspace $D \subset \mathbb{R}^d$, $D \cap \text{conv}(S_i^j) \neq \emptyset$ for all i and j , $0 \leq i \leq k$, $1 \leq j \leq r_i$.

This conjecture was confirmed in [Živ99] for the case where both d and k are odd integers and $r_i = q$ for each i , where q is an odd prime number. Recently S. Vrećica confirmed this conjecture also in the case $r_1 = \dots = r_k = 2$ [Vre03].

The expository article [Kal01] is recommended as a source of additional information about Tverberg-type theorems not covered here. From among Kalai's deep conjectures, beautiful visions, and unexpected possible connections (e.g., with the 4-color theorem), we select the following conjecture.

CONJECTURE 14.4.6 G. Kalai (1974)

Given a set $A \subset \mathbb{R}^d$, let $T_r(A)$ be the set of all points in \mathbb{R}^d that belong to the convex hull of r pairwise disjoint subsets of A . By convention let $\dim(\emptyset) = -1$. Then

$$\sum_{r=1}^{|A|} \dim(T_r(A)) \geq 0.$$

14.4.2 COLORED TVERBERG THEOREMS

Let $T(r, k, d)$ be the minimal number t so that for every collection of colors $\mathcal{C} = \{C_0, \dots, C_k\}$ with the property $|C_i| \geq t$ for all $i = 0, \dots, k$, there exist r multicolored sets $A_i = \{a_j^i\}_{j=0}^k$, $i = 1, \dots, r$, that are pairwise disjoint but where the corresponding rainbow simplices $\sigma_i := \text{conv } A_i$ have a nonempty intersection, $\bigcap_{i=1}^r \sigma_i \neq \emptyset$.

The colored Tverberg problem is to establish the existence of, and then to evaluate or estimate, the integer $T = T(r, k, d)$. The cases $k = d$ and $k < d$ are related, but there is also an essential difference. In the case $k = d$, provided t is large enough, the number of intersecting rainbow simplices can be arbitrarily large. In the case $k < d$, for dimension reasons, one cannot expect more than $r \leq d/(d - k)$ intersecting k -dimensional rainbow simplices. This is the reason why colored Tverberg theorems are classified as type A or type B, depending on whether $k = d$ or $k < d$.

In the type A case, where $T(r, d, d)$ is abbreviated simply as $T(r, d)$, it is easy to see that a lower bound for this function is r . It is conjectured that this lower bound is attained:

CONJECTURE 14.4.7 (Type A) [BL92]

$T(r, d) = r$ for all r and d .

This conjecture has been confirmed for $r = 2$ and for $d \leq 2$ [BL92].

The colored Tverberg problem (type A) was originally conjectured and designed as a tool for solving important problems of computational geometry (see [Section 14.4.3](#)). The weak form of the conjecture, $T(r, d) < +\infty$ [BFL90], is already far from obvious.

The following theorem of Živaljević and Vrećica (see [Bár93, Mat03, Živ98]) provides the best bounds known in the general case. It implies that $T(r, d) \leq 4r - 3$ for all r and d .

THEOREM 14.4.8 (Type A) [ZV92]

For every integer r and every collection of $d+1$ disjoint sets (“colors”) C_0, C_1, \dots, C_d in \mathbb{R}^d , each of cardinality at least $4r - 3$, there exist r disjoint, multicolored subsets $S_i \subset \bigcup_{i=0}^d C_i$ such that

$$\bigcap_{i=1}^r \text{conv } S_i \neq \emptyset.$$

If r is a power of a prime number then it suffices to assume that the size of each of the colors is at least $2r - 1$. In other words, $T(r, d) \leq 2r - 1$ if r is a prime power and $T(r, d) \leq 4r - 3$ in the general case.

In the type B case, let us assume that $r \leq d/(d - k)$, which is a necessary condition for a colored Tverberg theorem of type B.

CONJECTURE 14.4.9 (Type B)

$T(r, k, d) = 2r - 1$.

There exist examples showing that $T(r, k, d) \geq 2r - 1$.

The following theorem [VZ94, Živ98] confirms Conjecture 14.4.9 above for the case of a prime power r .

THEOREM 14.4.10 (Type B)

Let C_0, \dots, C_k be a collection of $k + 1$ disjoint finite sets (“colors”) in \mathbb{R}^d . Let r be a prime power such that $r \leq d/(d - k)$ and let $|C_i| = t \geq 2r - 1$. Then there exist r multicolored k -dimensional simplices S_i , $i = 1, \dots, r$, that are pairwise vertex-disjoint such that

$$\bigcap_{i=1}^r \text{conv } S_i \neq \emptyset.$$

The usual price for using topological (equivariant) methods is the extra assumption that r is a prime or a power of a prime number. On the other hand, the results obtained by these methods hold in greater generality and include nonlinear versions of Theorems 14.4.8 and 14.4.10; see [Živ98] for details and examples.

EXAMPLE 14.4.11

The simplest instance of Theorem 14.4.10 is the case $d = 2$, $k = 1$, and $r = 2$. Then, in the nonlinear version of this theorem, we recognize the well-known fact that the complete bipartite graph $K_{3,3}$ is not planar. This is one of the earliest results in topology, already known to Euler, who formulated it as a problem about three houses and three wells.

14.4.3 APPLICATIONS OF COLORED TVERBERG THEOREMS

Theorem 14.4.8 provided a general bound of the form $T(d+1, d) \leq 4d+1$, which opened the possibility of proving many interesting results in discrete and computational geometry.

HALVING HYPERPLANES AND THE k -SET PROBLEM

The number $h_d(n)$ of halving hyperplanes of a set of size n in \mathbb{R}^d , i.e., the number of essentially distinct placements of a hyperplane that split the set in half, according to Bárány, Füredi, and Lovász [BFL90], satisfies

$$h_d(n) = O(n^{d-\epsilon_d}), \quad \text{where } \epsilon_d = T(d+1, d)^{-(d+1)}.$$

POINT SELECTIONS AND WEAK ϵ -NETS

The equivalence of the following statements was established in [ABFK92] before Theorem 14.4.8 was proved. Considerable progress has since been made in this area [Mat02], and different combinatorial techniques for proving these statements have emerged in the meantime.

- Weak colored Tverberg theorem: $T(d+1, d)$ is finite.
- Point selection theorem: There exists a constant $s = s_d$, whose value depends on the bound for $T(d+1, d)$, such that any family \mathcal{H} of $(d+1)$ -element subsets of a set $X \subset \mathbb{R}^d$ of size $|\mathcal{H}| = p\binom{|X|}{d+1}$ contains a pierceable subfamily \mathcal{H}' such that $|\mathcal{H}'| \gg p^s \binom{|X|}{d+1}$. (\mathcal{H}' is **pierceable** if $\bigcap_{S \in \mathcal{H}'} \text{conv } S \neq \emptyset$. $A \gg_d B$ if $A \geq c_1(d)B + c_2(d)$, where $c_1(d) > 0$ and $c_2(d)$ are constants depending only on the dimension d .)
- Weak ϵ -net theorem: For any $X \subset \mathbb{R}^d$ there exists a weak ϵ -net F for convex sets with $|F| \ll_d \epsilon^{(d+1)(1-1/s)}$, where $s = s_d$ is as above. (See Chapter 36 for the notion of ϵ -net; a *weak* ϵ -net is similar, except that it need not be part of X .)
- Hitting set theorem: For every $\eta > 0$ and every $X \subset \mathbb{R}^d$ there exists a set $E \subset \mathbb{R}^d$ that misses at most $\eta \binom{|X|}{d+1}$ simplices of X and has size $|E| \ll_d \eta^{1-s_d}$, where s_d is as above.

OTHER RELATED RESULTS

A topological configuration space that arises via the CS/TM-scheme in proofs of Theorems 14.4.8 and 14.4.10 is the so-called **chessboard complex** $\Delta_{r,t}$, which owes its name to the fact that it can be described as the complex of all nontaking rook placements on an $r \times t$ chessboard. This is an interesting combinatorial object that arises independently as the coset complex of the symmetric group, as the complex of partial matchings in a complete bipartite graph, and as the complex of all partial injective functions. In light of the fact that the high connectivity of a configuration space is a property of central importance for applications (cf. Theorem 14.5.1), chessboard complexes have been studied from this point of view in numerous papers; see [Ath] and [Wac] for recent advances and references.

14.5 TOOLS FROM EQUIVARIANT TOPOLOGY

The method of equivariant maps is a versatile tool for proving results in discrete geometry and combinatorics. For many results these are the only proofs available. Equivariant maps are typically encountered at the final stage of application of the CS/TM-scheme (Section 14.1).

GLOSSARY

G -space X , G -action: A group G acts on a space X if each element of G is a continuous transformation of X and multiplication in G corresponds to composition of transformations. Formally, a G -action α is a continuous map $\alpha : G \times X \rightarrow X$ such that $\alpha(g, \alpha(h, x)) = \alpha(gh, x)$. Then X is called a G -space and $\alpha(g, x)$ is often abbreviated as $g \cdot x$ or gx .

Free G -action: An action is free if $g \cdot x = x$ for some $x \in X$ implies $g = e$, where e is the unit element in G .

G -equivariant map: A map $f : X \rightarrow Y$ of two G -spaces X and Y is equivariant if for all $g \in G$ and $x \in X$, $f(g \cdot x) = g \cdot f(x)$.

Borsuk-Ulam-type theorem: Any theorem establishing the nonexistence of a G -equivariant map between two G -spaces X and Y .

n -connected space: A path-connected and simply connected space with trivial homology in dimensions $1, 2, \dots, n$. A path-connected space X is simply connected or 1-connected if every closed loop $\omega : S^1 \rightarrow X$ can be deformed to a point.

The following generalization of the Borsuk-Ulam theorem is the key result used in proofs of many Tverberg-type statements. Note that if $X = S^n$, $Y = S^{n-1}$, and $G = \mathbb{Z}_2$, it specializes to the “odd” form of the Borsuk-Ulam theorem given in Section 14.2 (following Theorem 14.2.2).

THEOREM 14.5.1 [Dol83]

Suppose X and Y are simplicial (more generally CW) complexes equipped with the free action of a finite group G , and that X is m -connected, where $m = \dim Y$. Then there does not exist a G -equivariant map $f : X \rightarrow Y$.

Theorem 14.5.1 is strong enough for the majority of applications. Nevertheless, in some cases more sophisticated tools are needed. A *topological index theory* is a complexity theory for G -spaces that allows us to conclude that there does not exist a G -equivariant map $f : X \rightarrow Y$ if the G -space Y is of *larger complexity* than the G -space X . A measure of complexity of a given G -space is the so-called *equivariant index* $\text{Ind}_G(X)$. In general, an index function is defined on a class of G -spaces, say all finite G -CW complexes, and takes values in a suitable partially ordered set Ω . For example, if $G = \mathbb{Z}_2$, an index function $\text{Ind}_{\mathbb{Z}_2}(X)$ is defined as the minimum integer n such that there exists a \mathbb{Z}_2 -equivariant map $f : X \rightarrow S^n$. In this case $\Omega := \mathbb{N}$ is the poset of nonnegative integers. Note that the Borsuk-Ulam theorem simply states that $\text{Ind}_{\mathbb{Z}_2}(S^n) = n$.

PROPOSITION 14.5.2 [Mat03, Živ98]

For each nontrivial finite group G , there exists an integer-valued index function $\text{Ind}_G(\cdot)$ defined on the class of finite, G -simplicial complexes such that

- (i) If $\text{Ind}_G(Y) > \text{Ind}_G(X)$, then a G -equivariant map $f : X \rightarrow Y$ does not exist.
- (ii) If X is $(n-1)$ -connected then $\text{Ind}_G(X) \geq n$.
- (iii) If X is an n -dimensional, free G -complex then $\text{Ind}_G(X) \leq n$.
- (iv) $\text{Ind}_G(X * Y) \leq \text{Ind}_G(X) + \text{Ind}_G(Y) + 1$, where $X * Y$ is the join of spaces.

It is clear that the computation or good estimates of the complexity indices $\text{Ind}_G(X)$ are essential for applications. Occasionally this can be done even if the details of construction of the index function are not known. Such a tool for finding the lower bounds for an index function described in Proposition 14.5.2 is provided by the following inequality.

PROPOSITION 14.5.3 *Sarkaria inequality* [Mat03, Živ98]

Let L be a free G -complex and $L_0 \subset L$ a G -invariant, simplicial subcomplex. Let $\Delta(L \setminus L_0)$ be the order complex (cf. [Chapter 21](#)) of the complementary poset $L \setminus L_0$. Then

$$\text{Ind}_G(L_0) \geq \text{Ind}_G(L) - \text{Ind}_G(\Delta(L \setminus L_0)) - 1.$$

In some applications it is more natural, and sometimes essential, to use more sophisticated partially ordered sets of G -degrees of complexity. A notable example is the *ideal valued index theory* of S. Husseini and E. Fadell [FH88], which proved useful in establishing the existence of equilibrium points in incomplete markets (mathematical economics).

14.6 SOURCES AND RELATED MATERIAL

FURTHER READING

The reader will find additional information about applications of topological methods in discrete geometry and combinatorics, as well as a more comprehensive bibliography, in the survey papers [Alo88, Bár93, Bjö95, Eck93, Ste85, Živ98] as well as in the books [Mat02, Mat03].

The reader interested in broader aspects of the topology/computer science interaction is directed to the following sources:

- (1) Both [BEA+99] and [DEG99], surveys of existing applications, may also be seen as programs offering an insight into future research in computational topology, identifying some of the most important general research themes in this field.

- (2) The Web page of the *BioGeometry project*, [BioG], also includes information (α -shapes, topological persistence, etc.) about the topological aspects of the problem of designing computational techniques and paradigms for representing, storing, searching, simulating, analyzing, and visualizing biological structures.
 - (3) The *CompuTop.org Software Archive* (maintained by Nathan Dunfield) is focused on software for low-dimensional topological computations [Dun].
 - (4) The Lisp computer program *Kenzo* [Ser] exemplifies the powerful computational techniques now available in *effective algebraic topology*.
 - (5) For general information about algebraic topology the reader may find the Web site [WD] of the Hopf Topology Archive and the associated Topology discussion group (C. Wilkerson, D. Davis) extremely useful.
-

RELATED CHAPTERS

- [Chapter 1: Finite point configurations](#)
[Chapter 4: Helly-type theorems and geometric transversals](#)
[Chapter 32: Computational topology](#)
[Chapter 63: Biological applications of computational topology](#)
-

REFERENCES

- [AA89] J. Akiyama and N. Alon. Disjoint simplices and geometric hypergraphs. In G.S. Blum, R.L. Graham, and J. Malkevitch, editors, *Combinatorial Mathematics; Proceedings of the Third International Conference (New York 1985)*, vol. 555, pages 1–3. *Ann. New York Acad. Sci.*, 1989.
- [Alo87] N. Alon. Splitting necklaces. *Adv. Math.*, 63:247–253, 1987.
- [Alo88] N. Alon. Some recent combinatorial applications of Borsuk-type theorems. In M.M. Deza, P. Frankl, and D.G. Rosenberg, editors, *Algebraic, Extremal, and Metric Combinatorics*, pages 1–12. Cambridge University Press, 1988.
- [ABFK92] N. Alon, I. Bárány, Z. Füredi, and D. Kleitman. Point selections and weak ϵ -nets for convex hulls. *Combin. Probab. Comput.*, 1:189–200, 1992.
- [AET00] N. Amenta, D. Eppstein, and S-H. Teng. Regression depth and center points. *Discrete Comput. Geom.*, 23:305–329, 2000.
- [Ath] C. Athanasiadis. Decompositions and connectivity of matching and chessboard complexes. *Discrete Comput. Geom.*, to appear.
- [BB89] I.K. Babenko and S.A. Bogatyi. On the mapping of a sphere into Euclidean space (Russian). *Mat. Zametki*, 46:3–8, 1989; translated in *Math. Notes*, 46:683–686, 1989.
- [Bár93] I. Bárány. Geometric and combinatorial applications of Borsuk’s theorem. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, Volume 10 of *Algorithms Combin.* Springer-Verlag, Berlin, 1993.
- [BFL90] I. Bárány, Z. Füredi, and L. Lovász. On the number of halving planes. *Combinatorica*, 10:175–183, 1990.

- [BL92] I. Bárány and D.G. Larman. A colored version of Tverberg's theorem. *J. London Math. Soc.*, 45:314–320, 1992.
- [BM01] I. Bárány and J. Matoušek. Simultaneous partitions of measures by k -fans, *Discrete Comput. Geom.*, 25:317–334, 2001.
- [BM02] I. Bárány and J. Matoušek. Equipartitions of two measures by a 4-fan. *Discrete Comput. Geom.*, 27:293–301, 2002.
- [BSS81] I. Bárány, S.B. Shlosman, and A. Szűcs. On a topological generalization of a theorem of Tverberg. *J. London Math. Soc.*, 23:158–164, 1981.
- [BioG] BioGeometry project. <http://biogeometry.duke.edu>.
- [Bjö95] A. Björner. Topological methods. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 1819–1872. North-Holland, Amsterdam, 1995.
- [BEA+99] M. Bern et al. Emerging challenges in computational topology. *ACM Computing Research Repository*. arXiv:cs.CG/9909001.
- [Bor33] K. Borsuk. Drei Sätze über die n -dimensionale euklidische Sphäre. *Fund. Math.*, 20:177–190, 1933.
- [Bre93] G.E. Bredon. *Topology and Geometry*. Volume 139 of *Graduate Texts in Math.* Springer-Verlag, New York, 1993.
- [Bro75] L.E.J. Brouwer. *Collected Works*. North Holland, Amsterdam, 1975, 1976.
- [BB49] R.C. Buck and E.F. Buck. Equipartition of convex sets. *Math. Mag.* 22:195–198, 1949.
- [Car03] G. Carlsson, editor. *Proceedings of the Conference on Algebraic Topological Methods in Computer Science, 2001. Homology Homotopy Appl.*, 5(2), 2003.
- [Che98] W. Chen. Counterexamples to Knaster's conjecture. *Topology*, 37:401–405, 1998.
- [DEG99] T.K. Dey, H. Edelsbrunner, and S. Guha. Computational Topology. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*. Volume 223 of *Contemp. Math.*, pages 109–143. Amer. Math. Soc., Providence, 1999.
- [DEGN99] T.K. Dey, H. Edelsbrunner, S. Guha, and D.V. Nekhayev. Topology preserving edge contraction. *Publ. Inst. Math. (Beograd) (N.S.)*, 66:23–45, 1999.
- [Die89] J. Dieudonné. *A History of Algebraic and Differential Topology*. Birkhäuser, Boston, 1989.
- [Dol83] A. Dold. Simple proofs of some Borsuk-Ulam results. *Contemp. Math.*, 19:65–69, 1983.
- [Dol'93] V.L. Dol'nikov. Transversals of families of sets in \mathbb{R}^n and a relationship between Helly and Borsuk theorems. *Mat. Sb.*, 184:111–131, 1993.
- [Dun] CompuTop Software Archive. <http://www.math.harvard.edu/~nathand/computop/>.
- [Eck93] J. Eckhoff. Helly, Radon, and Carathéodory type theorems. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. North-Holland, Amsterdam, 1993.
- [FH88] E. Fadell and S. Husseini. An ideal-valued cohomological index theory with applications to Borsuk-Ulam and Bourgin-Yang theorems. *Ergodic Theory Dynam. Systems*, 8*:73–85, 1988.
- [HMS02] T. Hausel, E. Makai, Jr., and A. Szűcs. Inscribing cubes and covering by rhombic dodecahedra via equivariant topology. *Mathematika*, 47:371–397, 2002.
- [HR95] M. Herlihy and S. Rajsbaum. Algebraic topology and distributed computing—a primer. In *Computer Science Today*, Volume 1000 of *Lecture Notes in Comput. Sci.*, pages 203–217. Springer-Verlag, Berlin, 1995.

- [HR00] M. Herlihy and E. Ruppert. On the existence of booster types. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, 2000, pages 653–663.
- [KK93] J. Kahn and G. Kalai. A counterexample to Borsuk’s conjecture. *Bull. Amer. Math. Soc.*, 29:60–62, 1993.
- [Kak41] S. Kakutani. A generalization of Brouwer’s fixed point theorem. *Duke Math. J.*, 8:457–459, 1941.
- [Kal01] G. Kalai. Combinatorics with a geometric flavor. In N. Alon, J. Bourgain, A. Connes, M. Gromov, and V. Milman, editors, *Visions in Mathematics. Towards 2000. Geom. Funct. Anal. 2000*, Special Volume, Part II, pages 742–791. Birkhäuser, Basel, 2001.
- [KK99] A. Kaneko and M. Kano. Balanced partitions of two sets of points in the plane. *Comput. Geom. Theor. Appl.*, 13:253–261, 1999.
- [Kna47] B. Knaster. Problem 4. *Colloq. Math.*, 1:30, 1947.
- [Knu] D.E. Knuth. Generating all n -tuples, Chapter 7.2.1.1, prefascicle 2A of *The Art of Computer Programming*, vol. 4, released September 2001, <http://www-cs-faculty.stanford.edu/~knuth/fasc2a.ps.gz>.
- [Kup99] G. Kuperberg. Circumscribing constant-width bodies with polytopes. *New York J. Math.*, 5:91–100, 1999.
- [MVZ01] E. Makai, S. Vrećica, and R. Živaljević. Plane sections of convex bodies of maximal volume. *Discrete Comput. Geom.*, 25:33–49, 2001.
- [Mak86] V.V. Makeev. Some properties of continuous mappings of spheres and problems in combinatorial geometry. In L.D. Ivanov, editor, *Geometric Questions in the Theory of Functions and Sets* (Russian), Kalinin State Univ., 1986, pages 75–85.
- [Mak90] V.V. Makeev. The Knaster problem on the continuous mappings from a sphere to a Euclidean space. *J. Soviet Math.*, 52:2854–2860, 1990.
- [Mak94] V.V. Makeev. Inscribed and circumscribed polygons of a convex body. *Mat. Zametki*, 55:128–130, 1994; translated in *Math. Notes*, 55:423–425, 1994.
- [Mak98] V.V. Makeev. Some special configurations of planes that are associated with convex compacta (Russian). *Zap. Nauchn. Sem. S.-Petersburg* (POMI), 252:165–174, 1998.
- [Mak01] V.V. Makeev. Equipartition of a mass continuously distributed on a sphere and in space (Russian). *Zap. Nauchn. Sem. S.-Petersburg* (POMI), 279:187–196, 2001.
- [Mat] J. Matoušek. A combinatorial proof of Kneser’s conjecture. *Combinatorica*, to appear.
- [Mat02] J. Matoušek. *Lectures on Discrete Geometry*. Volume 212 of *Graduate Texts in Math.* Springer-Verlag, New York, 2002.
- [Mat03] J. Matoušek. *Using the Borsuk-Ulam Theorem. Lectures on Topological Methods in Combinatorics and Geometry*. Springer-Verlag, Berlin, 2003.
- [MTTV97] G.L. Miller, S.-H. Teng, W. Thurston, and S. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. Assoc. Comput. Mach.*, 44:1–29, 1997.
- [MTTV98] G.L. Miller, S.-H. Teng, W. Thurston, S.A. Vavasis. Geometric separators for finite-element meshes. *SIAM J. Sci. Comput.*, 19:364–386, 1998.
- [Mun84] J.R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, 1984.
- [Rad46] R. Rado. Theorem on general measure. *J. London Math. Soc.*, 21:291–300, 1946.
- [Ram96] E. Ramos. Equipartitions of mass distributions by hyperplanes. *Discrete Comput. Geom.*, 15:147–167, 1996.
- [Sar92] K.S. Sarkaria. Tverberg’s theorem via number fields. *Israel J. Math.*, 79:317–320, 1992.

- [Sar00] K.S. Sarkaria. Tverberg partitions and Borsuk-Ulam theorems. *Pacific J. Math.*, 196:231–241, 2000.
- [Sch93] L.J. Schulman. An equipartition of planar sets. *Discrete Comput. Geom.*, 9:257–266, 1993.
- [Ser] F. Sergeraert. “Kenzo”, a computer program for machine computations of homotopy/homology groups. <http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/>.
- [Soi02] Y. Soibelman. Topological Borsuk problem, arXiv:math.CO/0208221.
- [Ste85] H. Steinlein. Borsuk’s antipodal theorem and its generalizations and applications: a survey. In *Topological Methods in Nonlinear Analysis*, volume 95 of *Sém. Math. Sup.*, pages 166–235. Presses de l’Université de Montréal, 1985.
- [Tve66] H. Tverberg. A generalization of Radon’s theorem. *J. London Math. Soc.*, 41:123–128, 1966.
- [TV93] H. Tverberg and S. Vrećica. On generalizations of Radon’s theorem and the ham sandwich theorem. *European J. Combin.*, 14:259–264, 1993.
- [Vol96] A.Yu. Volovikov. On a topological generalization of the Tverberg theorem. *Math. Notes*, 59:324–32, 1996.
- [Vre03] S. Vrećica. Tverberg’s conjecture. *Discrete Comput. Geom.*, 29:505–510, 2003.
- [VZ92] S. Vrećica and R. Živaljević. The ham sandwich theorem revisited. *Israel J. Math.*, 78:21–32, 1992.
- [VZ94] S. Vrećica and R. Živaljević. New cases of the colored Tverberg theorem. In H. Barcelo and G. Kalai, editors, *Jerusalem Combinatorics ’93*, pages 325–334. Volume 178 of *Contemp. Math.*, Amer. Math. Soc., Providence, 1994.
- [VZ01] S. Vrećica and R. Živaljević. Conical equipartitions of mass distributions. *Discrete Comput. Geom.*, 25:335–350, 2001.
- [Wac] M.L. Wachs. Topology of matching, chessboard, and general bounded degree graph complexes. *Algebra Universalis* (Gian-Carlo Rota memorial issue), to appear.
- [WD] Hopf Topology Archive. <http://hopf.math.psu.edu/pub/hopf.html> .
- [Zie02] G.M. Ziegler. Generalized Kneser coloring theorems with combinatorial proofs. *Invent. Math.*, 147:671–691, 2002.
- [Živ98] R. Živaljević. User’s guide to equivariant methods in combinatorics, I and II. *Publ. Inst. Math. (Beograd) (N.S.)*, (I) 59(73):114–130, 1996 and (II) 64(78):107–132, 1998.
- [Živ99] R. Živaljević. The Tverberg-Vrećica problem and the combinatorial geometry on vector bundles. *Israel J. Math.*, 111:53–76, 1999.
- [ZV90] R. Živaljević and S. Vrećica. An extension of the ham sandwich theorem. *Bull. London Math. Soc.*, 22:183–186, 1990.
- [ZV92] R. T. Živaljević and S.T. Vrećica. The colored Tverberg’s problem and complexes of injective functions. *J. Combin. Theory Ser. A*, 61:309–318, 1992.

15 POLYOMINOES

Solomon W. Golomb and David A. Klarner¹

INTRODUCTION

A *polyomino* is a finite, connected subgraph of the square-grid graph consisting of infinitely many unit cells matched edge-to-edge, with pairs of adjacent cells forming edges of the graph. Polyominoes have a long history, going back to the start of the 20th century, but they were popularized in the present era initially by Solomon Golomb, then by Martin Gardner in his *Scientific American* columns “Mathematical Games.” They now constitute one of the most popular subjects in mathematical recreations, and have found interest among mathematicians, physicists, biologists, and computer scientists as well.

15.1 BASIC CONCEPTS

GLOSSARY

Cell: A unit square in the Cartesian plane with its sides parallel to the coordinate axes and with its center at an integer point (u, v) . This cell is denoted $[u, v]$ and identified with the corresponding member of \mathbb{Z}^2 .

Adjacent cells: Two cells, $[u, v]$ and $[r, s]$, with $|u - r| + |v - s| = 1$.

Square-grid graph: The graph with vertex set \mathbb{Z}^2 and an edge for each pair of adjacent cells.

Polyomino: A finite set S of cells such that the induced subgraph of the square-grid graph with vertex set S is connected. A polyomino with exactly n cells is called an *n-omino*. Polyominoes are also known as *animals*.

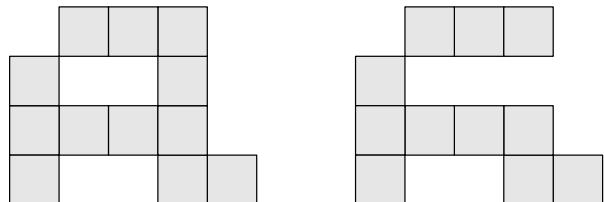


FIGURE 15.1.1

Two sets of cells: the set on the left is a polyomino, the one on the right is not.

¹This is a revision, by S.W. Golomb, of the chapter of the same title originally written for the first edition by the late D.A. Klarner.

15.2 EQUIVALENCE OF POLYOMINOES

Notions of equivalence for polyominoes are defined in terms of groups of affine maps that act on the set \mathbb{Z}^2 of cells in the plane.

GLOSSARY

Translation by (r, s) : The mapping from \mathbb{Z}^2 to itself that maps $[u, v]$ to $[u + r, v + s]$; it sends any subset $S \subset \mathbb{Z}^2$ to its *translate* $S + (r, s) = \{[u + r, v + s] : [u, v] \in S\}$.

Translation-equivalent: Sets S, S' of cells such that S' is a translate of S .

Fixed polyomino: A translation-equivalence class of polyominoes; $t(n)$ denotes the number of fixed n -ominoes.

Representatives of the six fixed 3-ominoes are shown in Figure 15.2.1.

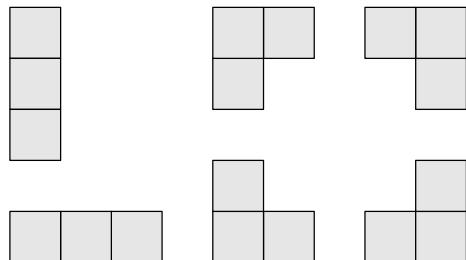


FIGURE 15.2.1

The six fixed 3-ominoes.

Lexicographically minimum cell: The unique member $[u, v]$ of a finite set $S \subset \mathbb{Z}^2$ with $v = \min\{v' : [u', v'] \in S\}$, $u = \min\{u' : [u', v] \in S\}$.

Standard position: The translate $S - (u, v)$ of S , where $[u, v]$ is the lexicographically minimum cell in S .

Rotation-translation group: The group \mathcal{R} of mappings of \mathbb{Z}^2 to itself of the form $[u, v] \mapsto [u, v] \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^k + (r, s)$. (The matrix $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, which is denoted by R , maps $[u, v]$ to $[v, -u]$ by right multiplication, hence represents a clockwise rotation of 90° .)

Rotationally equivalent: Sets S, S' of cells with $S' = \rho S$ for some $\rho \in \mathcal{R}$.

Chiral polyomino, or handed polyomino: A rotational-equivalence class of polyominoes; $r(n)$ denotes the number of chiral n -ominoes.

The top row of 5-ominoes in Figure 15.2.2 consists of the set of cells $F = \{[0, -1], [-1, 0], [0, 0], [0, 1], [1, 1]\}$, together with FR , FR^2 , and FR^3 . All four of these 5-ominoes are rotationally equivalent. The bottom row in Figure 15.2.2 shows these same four 5-ominoes reflected about the x -axis. These four 5-ominoes are rotationally equivalent as well, but none of them is rotationally equivalent to any of the 5-ominoes shown in the top row. Representatives of the seven chiral 4-ominoes are shown in Figure 15.2.3.

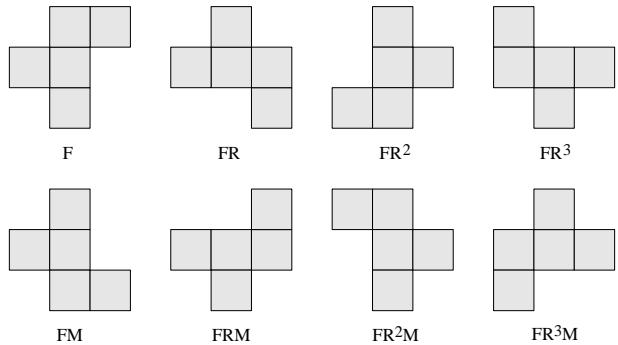


FIGURE 15.2.2

The 5-ominoes in the top row are rotationally equivalent, and so are their reflections in the bottom row, but the two sets are rotationally distinct.

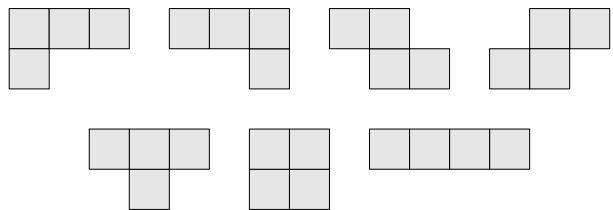


FIGURE 15.2.3

The seven chiral 4-ominoes.

Congruence group: The group \mathcal{S} of motions generated by the matrix $M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (reflection in the x -axis) and the rotation-translation group \mathcal{R} . (A typical element of \mathcal{S} has the form $[u, v] \mapsto [u, v]R^k M^i + (r, s)$, for some $k = 0, 1, 2$, or 3 , some $i = 0$ or 1 , and some $r, s \in \mathbb{Z}$.)

Congruent: Sets S, S' of cells such that $S' = \sigma(S)$ for some $\sigma \in \mathcal{S}$.

Free polyomino: A congruence class of polyominoes; $s(n)$ denotes the number of free n -ominoes.

The twelve free 5-ominoes are shown in Figure 15.2.4.

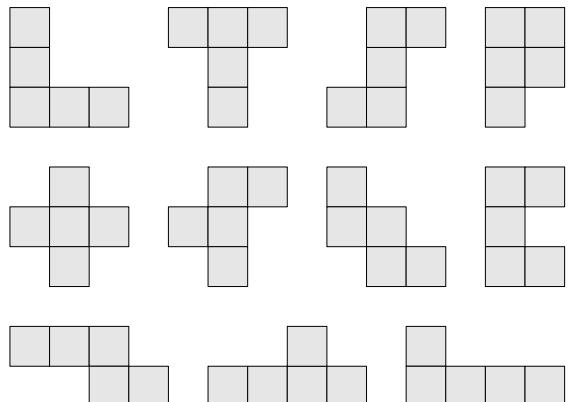


FIGURE 15.2.4

The twelve free 5-ominoes.

Standard position: A finite set $S \subset \mathbb{Z}^2$ is in standard position if and only if $[0, 0] \in S$, $0 \leq v$ for all $[u, v] \in S$, and $0 \leq u$ for all $[u, 0] \in S$.

THEOREM 15.2.1 *Embedding Theorem*

For each n , let U_n consist of the $n^2 - n + 1$ cells of the form $[u, v]$, where $\begin{cases} 0 \leq u \leq n, & \text{for } v = 0 \\ |u| + v \leq n, & \text{for } v > 0 \end{cases}$. (See Figure 15.2.5 for the case $n = 5$.) Then every n -omino in standard position is a subset of U_n .

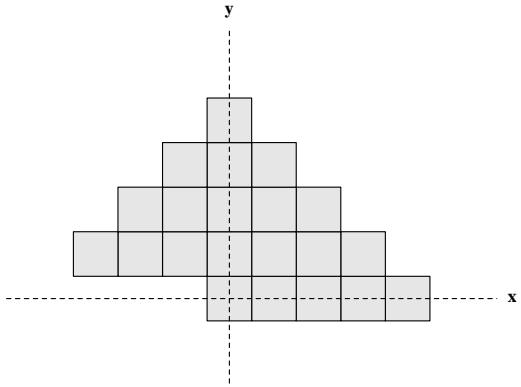


FIGURE 15.2.5
A set of $n^2 - n + 1$ cells that contains every n -omino in standard position.

COROLLARY 15.2.2

The number of fixed n -ominoes is finite for each n .

15.3 HOW MANY n -OMINOES ARE THERE?

Table 15.3.1, calculated by Redelmeier [Red81], indicates the values of $t(n)$, $r(n)$, and $s(n)$ for $n = 1, \dots, 24$.

The values seem to be growing exponentially, and indeed they have exponential bounds. It is easy to see that for each n ,

$$\frac{t(n)}{8} \leq s(n) \leq r(n) \leq t(n),$$

and results of Klarner and Rivest [KR73], and of Klarner and Satterfield [KS], using automata theory and building on earlier work of Eden, Klarner, and Read, have shown:

THEOREM 15.3.1

$\lim_{n \rightarrow \infty} (t(n))^{1/n} = \theta$ exists, and $3.9 < \theta < 4.65$.

Jensen and Guttmann [JG00], using an improved algorithm, have extended the enumeration of polyominoes to $n = 46$, but without publishing an extension of Table 15.3.1. They proved $\theta > 3.90318$, and obtained the estimate $\theta \approx 4.062570\dots$.

TABLE 15.3.1 The number of fixed, chiral, and free n -ominoes for $n \leq 24$.

n	$t(n)$	$r(n)$	$s(n)$
1	1	1	1
2	2	1	1
3	6	2	2
4	19	7	5
5	63	18	12
6	216	60	35
7	760	196	108
8	2725	704	369
9	9910	2500	1285
10	36446	9189	4655
11	135268	33896	17073
12	505861	126759	63600
13	1903890	476270	238591
14	7204874	1802312	901971
15	27394666	6849777	3426576
16	104592937	26152418	13079255
17	400795844	100203194	50107909
18	1540820542	385221143	192622052
19	5940738676	1485200848	742624232
20	22964779660	5741256764	2870671950
21	88983512783	22245940545	11123060678
22	345532572678	86383382827	43191857688
23	1344372335524	336093325058	168047007728
24	5239988770268	1309998125640	654999700403

A related, slightly earlier paper by Guttmann, Jensen, et al. [GJW00] describes a method for enumerating “punctured” polyominoes (i.e., those containing holes).

ALGORITHMS

Considerable effort has been expended to find a formula for the number of fixed n -ominoes (say), with no success. Redelmeier’s algorithm, which produced the entries in Table 15.3.1 (and took over ten months of computer time to run), generates the fixed n -ominoes one by one and counts them. Although the running time is (necessarily) exponential, the algorithm takes only $O(n)$ space. Improved algorithms have since been found [JG00], but none has subexponential running time.

UNSOLVED PROBLEMS

PROBLEM 15.3.2

Can $t(n)$ be computed by a polynomial-time algorithm?

A related problem concerns the constant θ defined above:

PROBLEM 15.3.3

Is there a polynomial algorithm to find, for each n , an approximation θ_n of θ satisfying

$$10^{-n} < |\theta_n - \theta| < 10^{-n+1} ?$$

The lower-bound method of [KS1] gives an algorithm for approximating θ from below that has exponential complexity; no such method is known for approximating θ from above.

PROBLEM 15.3.4

Define some decreasing sequence $\beta = (\beta_1, \beta_2, \dots)$ that tends to θ , and give an algorithm to compute β_n for every n .

It is known that $(t(n))^{1/n} \leq \theta$ for all n , and it seems that the ratios $\tau(n) = t(n+1)/t(n)$ increase for all n . If the latter is true, $\tau(n)$ would approach θ from below. This gives two more unsolved problems:

PROBLEM 15.3.5

Show that $(t(n))^{1/n} < (t(n+1))^{1/(n+1)}$ for all n .

PROBLEM 15.3.6

Show that $\tau(n) < \tau(n+1)$ for all n .

15.4 GENERATING POLYOMINOES

The algorithm we describe to generate all n -ominoes, which is essentially due to Redelmeier [Red81], also provides a way of encoding n -ominoes. Starting with all n -ominoes in standard position, with each cell and each neighboring cell numbered, it constructs all numbered $(n+1)$ -ominoes in standard position.

GLOSSARY

Border cell of an n -mino S : A cell $[u, v]$, with $v \geq 0$ or with $v = 0$ and $u \geq 0$, adjacent to some cell of S . The set of all border cells, which is denoted by $B(S)$, can be shown by induction to have no more than $2n$ elements.

The **lexicographic cell ordering** \prec on \mathbb{Z}^2 is defined by: $[r, s] \prec [u, v]$ if $s < v$, or if $s = v$ and $r < u$.

The algorithm, illustrated in [Figure 15.4.1](#) for $n = 1, 2$, and 3 , begins with cell 1 in position $[0, 0]$, with its border cells marked 2 and 3, and then adds these—one at a time—each time numbering *new* border cells in their lexicographic order. Whenever a number used for a border cell is not larger than the largest internal number, it is circled, and the corresponding cell is *not* added at the next stage.

[Figure 15.4.2](#) shows all the 4-ominoes produced in this way, with their border cells marked for the next step of the algorithm.

FIGURE 15.4.1

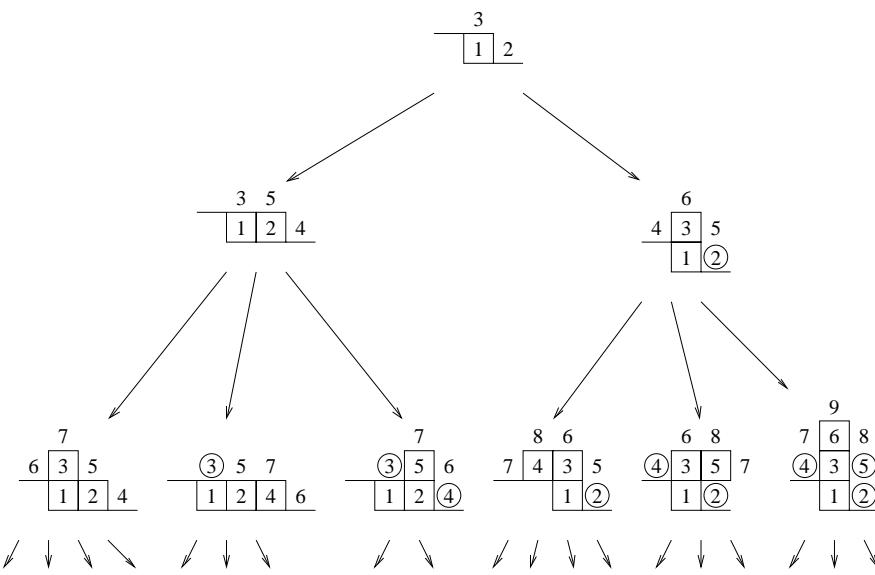
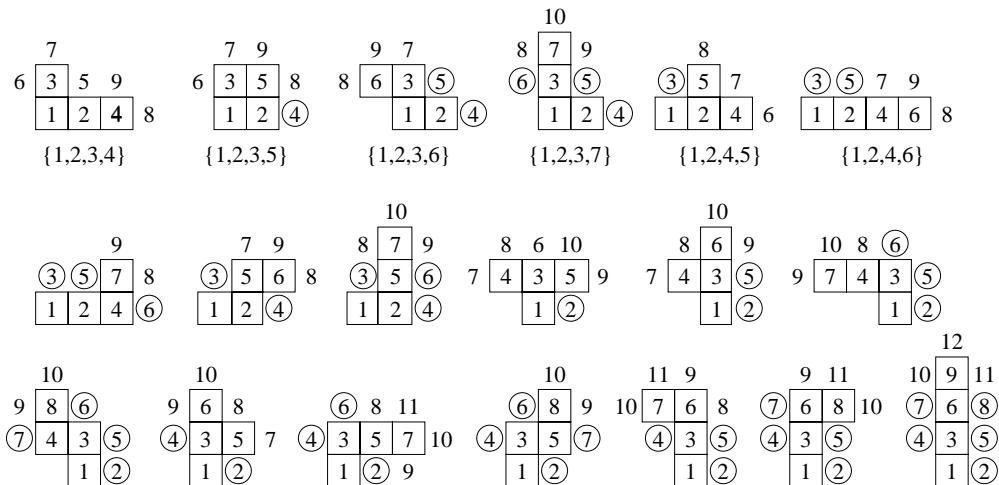


FIGURE 15.4.2



This process assigns a unique set of positive integers to each n -omino S , also illustrated in Figure 15.4.2. The *set character functions* for these integer sets, in turn, truncated after their final 1's, provide a binary codeword $\chi(S)$ for each n -omino S . For example, the code words for the first three 4-ominoes in Figure 15.4.2 would be 1111, 11101, and 111001.

PROBLEM 15.4.1

Which binary strings arise as codewords for n -ominoes?

The following is easy to see:

THEOREM 15.4.2

$t(n+1) = \sum n + |B(S)| - |\chi(S)|$, where the sum extends over all n -ominoes S in standard position, and $|\chi(S)|$ is the number of bits in the codeword of S .

PROBLEM 15.4.3

Is the generating function $T(z) = \sum_{n=1}^{\infty} t(n)z^n$ a rational function? Is $T(z)$ even algebraic?

15.5 SPECIAL TYPES OF POLYOMINOES

Particular kinds of polyominoes arise in various contexts. We will look at several of the most interesting ones.

GLOSSARY

A ***composition*** of n with k parts is an ordered k -tuple (p_1, \dots, p_k) of positive integers with $p_1 + \dots + p_k = n$.

A polyomino is called *row-convex* if every (horizontal) row consists of a single strip of cells. It is *row-column-convex* if this holds for every column as well.

Simply connected polyomino: A polyomino without holes. (Golomb calls these nonholey polyominoes **profane**.)

A ***width-k*** polyomino: One each of whose vertical cross sections fits in a $k \times 1$ strip of cells.

A **directed** polyomino is defined recursively as follows: Any single cell is a directed polyomino. An $(n+1)$ -omino is directed if it can be obtained by adding a new cell immediately above, or to the right of, a cell belonging to some directed n -omino.

COMPOSITIONS AND ROW-CONVEX POLYOMINOES

There is a natural 1-1 correspondence between compositions of n and a certain class of n -ominoes in standard position, as indicated in Figure 15.5.1 for the case $n = 4$.

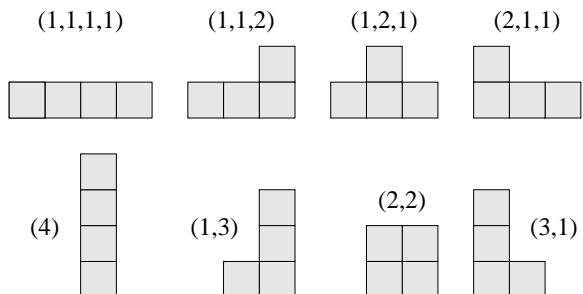


FIGURE 15.5.1
Compositions of 4 corresponding to certain 4-ominoes.

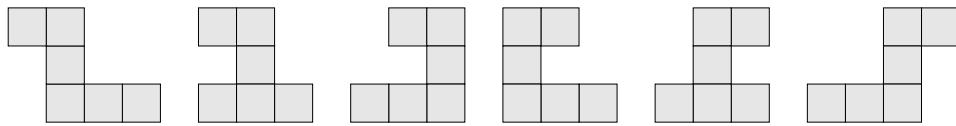
Let us, instead, assign to each composition (a_1, \dots, a_k) of n an n -omino with a *horizontal* strip of a_i cells in row i . This can be done in many ways, and the results are all the row-convex n -ominoes. Since there are $m + n - 1$ ways to form an $(m+n)$ -omino by placing a strip of n cells atop a strip of m cells, it follows that for each composition (a_1, \dots, a_k) of n into positive parts, there are

$$(a_1 + a_2 - 1)(a_2 + a_3 - 1) \cdots (a_{k-1} + a_k - 1)$$

n ominoes having a strip of a_i cells in the i th row for each i (see Figure 15.5.2 for an example arising from the composition $6 = 3 + 1 + 2$).

FIGURE 15.5.2

The 6 row-convex 6-ominoes corresponding to the composition $(3, 1, 2)$ of 6.



It follows that if $b(n)$ is the number of row-convex n -ominoes, then

$$b(n) = \sum (a_1 + a_2 - 1)(a_2 + a_3 - 1) \cdots (a_{k-1} + a_k - 1),$$

where the sum extends over all compositions (a_1, \dots, a_k) of n into k parts, for all k . $b(n)$, and the generating function $B(z) = \sum_{n=1}^{\infty} b(n)z^n$, are given by

THEOREM 15.5.1 [Kla67]

$$b(n+3) = 5b(n+2) - 7b(n+1) + 4b(n), \text{ and } B(z) = \frac{z(1-z)^3}{1-5z+7z^2-4z^3}.$$

COROLLARY 15.5.2

$\lim_{n \rightarrow \infty} (b(n))^{1/n} = \beta$, where β is the largest real root of $z^3 - 5z^2 + 7z - 4 = 0$;
 $3.20 < \beta < 3.21$.

ROW-COLUMN-CONVEX POLYOMINOES

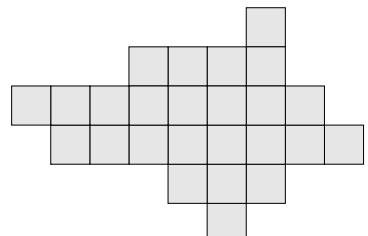


FIGURE 15.5.3

A typical row-column-convex polyomino.

The problem of finding the number, $c(n)$, of row-column-convex polyominoes was

first posed by D. Knuth [Knu72]. The existence of a generating function for $c(n)$ with special properties, proved in [KR74], enabled Bender to prove the following asymptotic formula:

THEOREM 15.5.3 [Ben74]

$c(n) \sim cg^n$, where $c = 2.67564\dots$ and $g = 2.30914\dots$

The following problem concerns polyominoes radically different from row-column-convex ones.

PROBLEM 15.5.4

Find the smallest natural number n such that there exists an n -omino with no row or column consisting of just a single strip of cells. (An example of a 21-omino with this property is shown in Figure 15.5.4.)

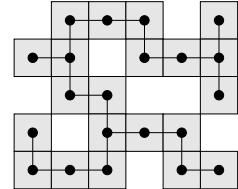


FIGURE 15.5.4

A 21-omino with no row or column a single strip of cells.

SIMPLY CONNECTED POLYOMINOES

Let $t^*(n)$, $s^*(n)$, and $r^*(n)$ denote the numbers of profane fixed, free, and chiral n -ominoes, respectively. Not much is known about their values.

PROBLEM 15.5.5

Compute $t^(n)$, $s^*(n)$, and $r^*(n)$ for as many values of n as possible.*

It is easy to see that $(t^*(n))^{1/n}$, $(s^*(n))^{1/n}$, and $(r^*(n))^{1/n}$ all approach the same limit, θ^* , as $n \rightarrow \infty$, and that $\theta^* \leq \theta$ ($= \lim_{n \rightarrow \infty} (t(n))^{1/n}$ as defined in Section 15.3).

PROBLEM 15.5.6 [Gol]

Does $\theta^ = \theta$?*

We conjecture that the answer is “no.”

WIDTH- k POLYOMINOES

A typical width-3 polyomino is shown in Figure 15.5.5.

THEOREM 15.5.7 [Rea62, KS]

Let $t(n, k)$ be the number of fixed width- k n -ominoes, and $T_k(z) = \sum_{n=1}^{\infty} t(n, k)z^n$. Then $T_k(z) = P_k(z)/Q_k(z)$ for some polynomials $P_k(z), Q_k(z)$ with integer coefficients, no common zeroes, and $Q_k(0) = 1$. Equivalently, the sequence $t(n, k)$, $n = 1, 2, \dots$, satisfies a linear, homogeneous difference equation with constant coef-

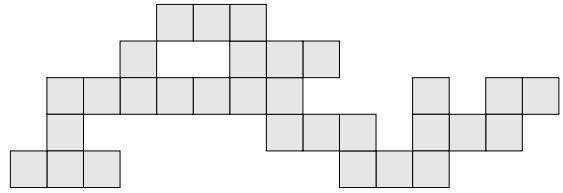


FIGURE 15.5.5
A width-3 polyomino.

ficients for each fixed k ; the order of the equation is roughly 3^k . Furthermore, the sequence $(t(n, k))^{1/n}$ converges to a limit τ_k as $n \rightarrow \infty$, and $\lim_{k \rightarrow \infty} \tau_k = \theta$ (see Section 15.3).

For example, for the fixed width-2 n -ominoes (shown in Figure 15.5.6 for small n), we have

$$T_2(z) = \frac{z}{1 - 2z - z^2} = z + 2z^2 + 5z^3 + 12z^4 + \dots,$$

and $t(n+2, 2) = 2t(n+1, 2) + t(n, 2)$ for $n \geq 1$.

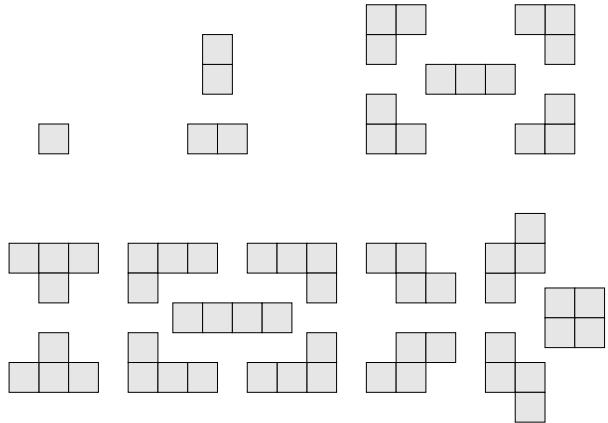


FIGURE 15.5.6
Width-2 n -ominoes for $n = 1, 2, 3, 4$.

DIRECTED POLYOMINOES

A portion of the family tree for directed polyominoes, constructed similarly to the one in Figure 15.4.1, is shown in Figure 15.5.7. As in Section 4, codewords can be defined for directed polyominoes, and converted into binary words. Let \mathcal{V} be the language formed by all of these.

PROBLEM 15.5.8

Characterize the words in \mathcal{V} . In particular, is \mathcal{V} an unambiguous context-free language?

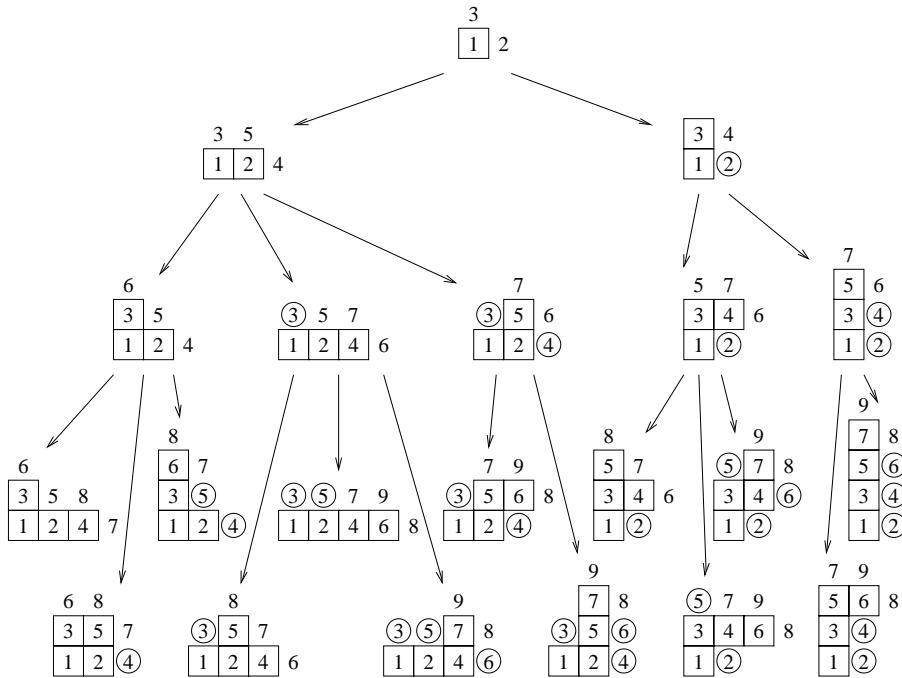
THEOREM 15.5.9 [Bou94]

If $d(n)$ is the number of directed n -ominoes in standard position, and $D(z) = \sum d(n)z^n$, then

$$D(z) = \frac{1}{2} \left(\sqrt{\frac{1+z}{1-3z}} - 1 \right).$$

FIGURE 15.5.7

A family tree for fixed directed polyominoes.



COROLLARY 15.5.10

$$d(n) = \frac{1}{2} \sum_{k=0}^n \binom{1/2}{k} \binom{-1/2}{n-k} (-3)^{n-k}$$

and $d(n)$ satisfies the recurrence relation

$$d(n) = 3^{n-1} - \sum_{k=1}^{n-1} d(k)d(n-k).$$

D. Kaiser [Kai95] used the corollary to generate [Table 15.5.1](#).

15.6 TILING WITH POLYOMINOES

We consider the special case of the tiling problem (see [Chapter 3](#)) in which the space we wish to tile is a set S of cells in the plane and the tiles are polyominoes. Usually S will be a rectangular set.

GLOSSARY

π -type: If S is a finite set of cells, \mathcal{C} a collection of subsets of S , $\pi = (S_1, \dots, S_k)$

TABLE 15.5.1 The first 30 values of $d(n)$, the number of directed n -ominoes in standard position.

n	$d(n)$	n	$d(n)$	n	$d(n)$
1	1	11	17303	21	741365049
2	2	12	49721	22	2173243128
3	5	13	143365	23	6377181825
4	13	14	414584	24	17830782252
5	35	15	1201917	25	55062568341
6	96	16	3492117	26	161995031226
7	267	17	10165779	27	476941691177
8	750	18	29643870	28	1405155255055
9	2123	19	86574831	29	4142457992363
10	6046	20	253188111	30	12219350698880

a partition (or cover) of S , and $T \subset S$, the π -type of T is defined as

$$\tau(\pi, T) = (|S_1 \cap T|, \dots, |S_k \cap T|).$$

Basis: If every rectangle in a set R can be tiled with translates of rectangles belonging to a finite subset $B \subset R$, and if B is minimal with this property, B is called a basis of R .

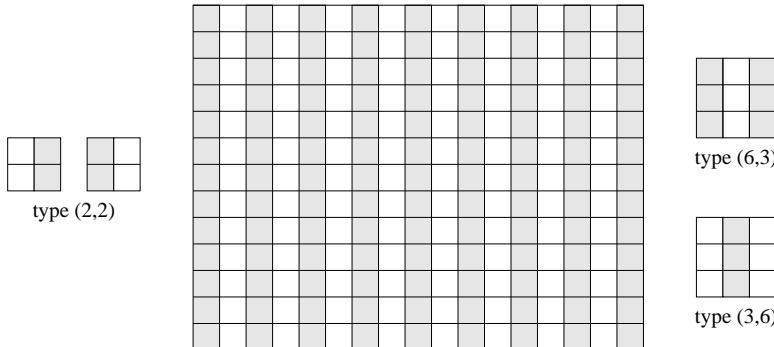
THEOREM 15.6.1 [Kla70]

Suppose S is a finite set and \mathcal{C} a collection of subsets of S . Then \mathcal{C} tiles S if and only if, for every partition (or cover) π of S , $\tau(\pi, S)$ is a nonnegative integer combination of the types $\tau(\pi, T)$ where T ranges over \mathcal{C} .

For example, one can use this to show that a 13×17 rectangular array of squares cannot be tiled with 2×2 and 3×3 squares: Let π be the partition of the 13×17 array S into “black” and “white” cells shown in Figure 15.6.1, and \mathcal{C} the set of all 2×2 and 3×3 squares in S .

FIGURE 15.6.1

A coloring of the 13×17 rectangle.



Then each 2×2 square in \mathcal{C} has type $(2, 2)$, while the 3×3 squares have types $(6, 3)$ and $(3, 6)$. If a tiling were possible, with x 2×2 squares, and with y_1 and y_2 3×3 squares of types $(6, 3)$ and $(3, 6)$ (respectively), then we would have

$$(9 \cdot 13, 8 \cdot 13) = x(2, 2) + y_1(6, 3) + y_2(3, 6),$$

which gives $13 = 3(y_1 - y_2)$, a contradiction.

THEOREM 15.6.2

Let \mathcal{C} be a finite union of translation classes of polyominoes, and let w be a fixed positive integer. Then one can construct a finite automaton that generates all \mathcal{C} -tilings of $w \times n$ rectangles for all possible values of n .

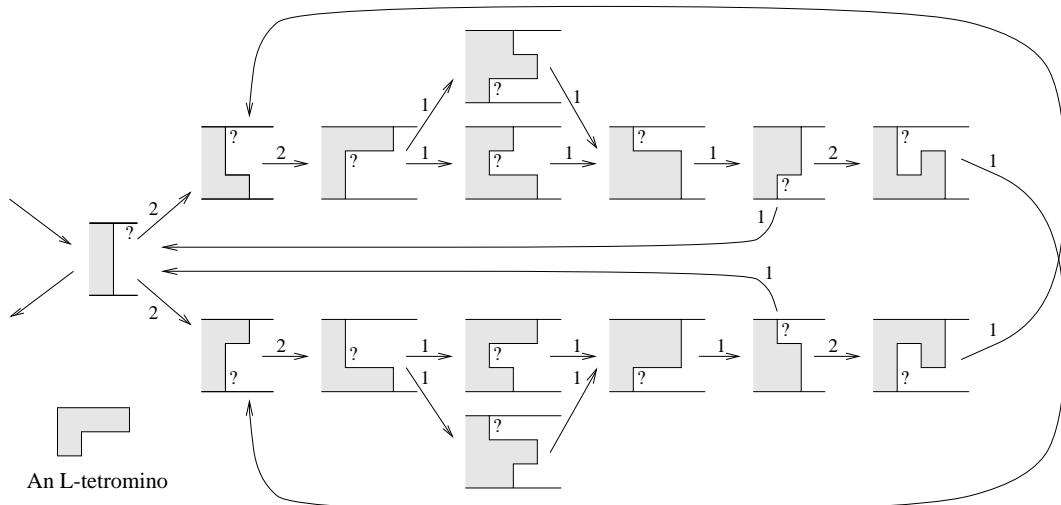
COROLLARY 15.6.3

If w is fixed and \mathcal{C} is given, then it is possible to decide whether there exists some n for which \mathcal{C} tiles a $w \times n$ rectangle.

For example, if we want to tile a $3 \times n$ rectangle with copies of the L -tetromino shown in Figure 15.6.2 in all eight possible orientations, the automaton of Figure 15.6.2 shows that it is necessary and sufficient for n to be a multiple of 8.

FIGURE 15.6.2

An automaton for tiling a $3 \times n$ rectangle with L -tetrominoes.



THEOREM 15.6.4 [KG69, dBK75]

Let R be an infinite set of oriented rectangles with integer dimensions. Then R has a finite basis.

(This theorem, which was originally conjectured by F. Göbel, extends to higher dimensions as well [dBK75].)

For example, let R be the set of all rectangles that can be tiled with the L -tetromino of Figure 15.6.2, and let $B = \{2 \times 4, 4 \times 2, 3 \times 8, 8 \times 3\} \subset R$. Then the following three facts are related:

- (a) R is the set of all $a \times b$ rectangles with $a, b > 1$ and $8|ab$;
- (b) B is a basis of R ;
- (c) Each member of B is tilable with the L -tetromino.

PROBLEM 15.6.5

The smallest rectangle that can be tiled with the Y -pentomino (shown in Figure 15.6.3) is 5×10 . Find a basis B for the set R of all rectangles that can be tiled with Y -pentominoes.

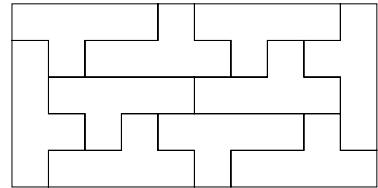


FIGURE 15.6.3

A 5×10 rectangle tiled with Y -pentominoes.

15.7 RECTANGLES OF POLYOMINOES

Here we consider the question of which polyomino shapes have the property that some finite number of copies, allowing all rotations and reflections, can be assembled to form a rectangle. No *a priori* limit can be established, given an arbitrary n -omino, on the minimum number of copies that *may* be required to form a rectangle. (See, e.g., [Ber66].) Fortunately, in any *specific* case, there is a high likelihood of answering the question. But there can be no procedure that can be routinely applied to indicate whether a given polyomino shape will tile some (possibly huge) rectangle.

In 1968, D.A. Klarner [Kla69] defined the **order** of a polyomino P as the minimum number of congruent copies of P that can be assembled (allowing translation, rotation, and reflection) to form a rectangle. For those polyominoes that will not tile any rectangle, the order is undefined. (A polyomino has order 1 if and only if it is itself a rectangle.)

A polyomino has order 2 if and only if it is “half a rectangle”, since two identical copies of it must form a rectangle. This necessarily means that the two copies will be 180° rotations of each other when forming a rectangle. Some examples are shown in Figure 15.7.1.

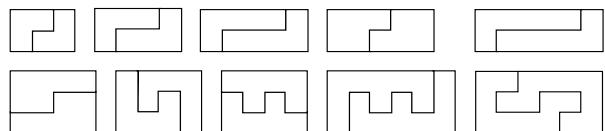


FIGURE 15.7.1

Some polyominoes of order 2.

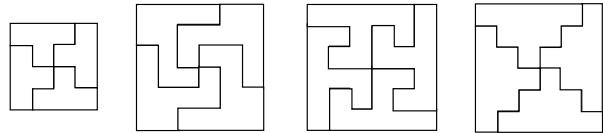
There are no polyominoes of order 3. (This was proved in [SW92] by Ian Stewart.) In fact, the only way any rectangle can be divided up into three identical copies of a “well-behaved” geometric figure is to partition it into three *rectangles* (see Figure 15.7.2), and by definition a rectangle has order 1.

FIGURE 15.7.2
How three identical rectangles can form a rectangle.



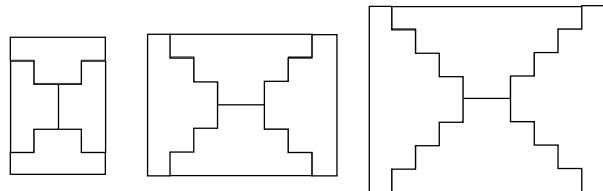
There are various ways in which four identical polyominoes can be combined to form a rectangle. One way, illustrated in Figure 15.7.3, is to have four 90° rotations of a single shape forming a square.

FIGURE 15.7.3
Polyominoes of order 4 under 90° rotation.



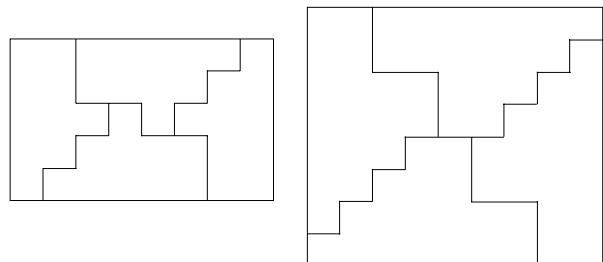
Another way to combine four identical shapes to form a rectangle uses the fourfold symmetry of the rectangle itself: left-right, up-down, and 180-rotational symmetry. Some examples of this appear in Figure 15.7.4.

FIGURE 15.7.4
Polyominoes of order 4 under rectangular symmetry.



There are also more complicated order-4 patterns that were found by Klärner [Kla69], two of which are illustrated in Figure 15.7.5.

FIGURE 15.7.5
Another order-4 construction by Klärner.



Beyond order 4, there is a systematic construction [Gol89] that gives examples of order $4s$ for every positive integer s ; numerous isolated examples of small polyominoes with orders including 10, 18, 24, 28, 50, 76, 92, 96, 138, 192, and 312 are also known.

Figure 15.7.6 shows the isolated examples of order 10 [Gol66] and orders 18, 24, and 28 [Kla69].

FIGURE 15.7.6

Four “sporadic” polyominoes, of orders 10, 18, 24, and 28, respectively.

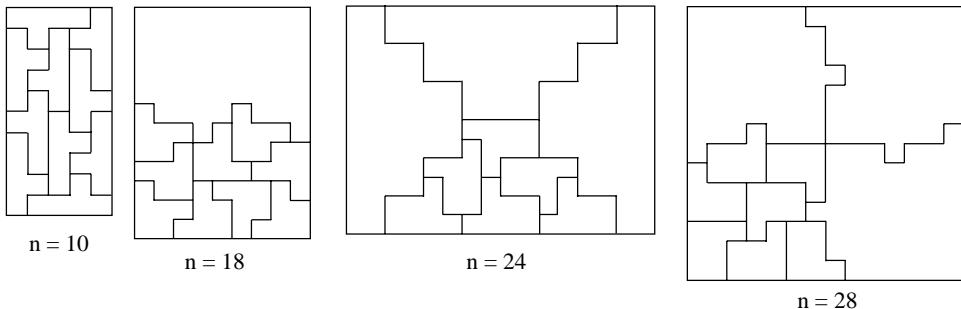


Figure 15.7.7 shows an example of order 50, found by William Rex Marshall of Dunedin, New Zealand, in 1990 [Mar90].

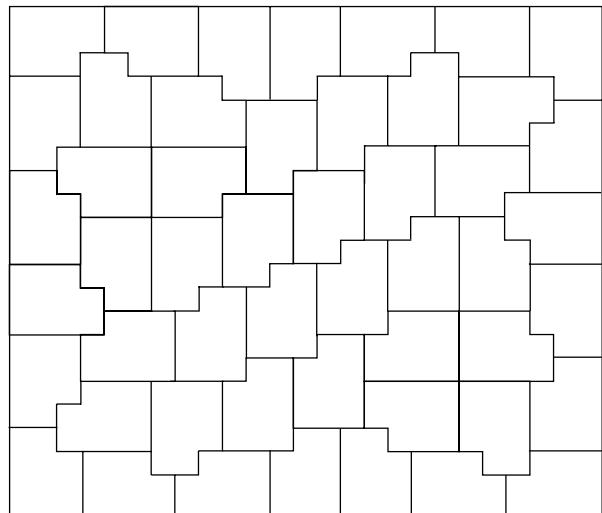


FIGURE 15.7.7

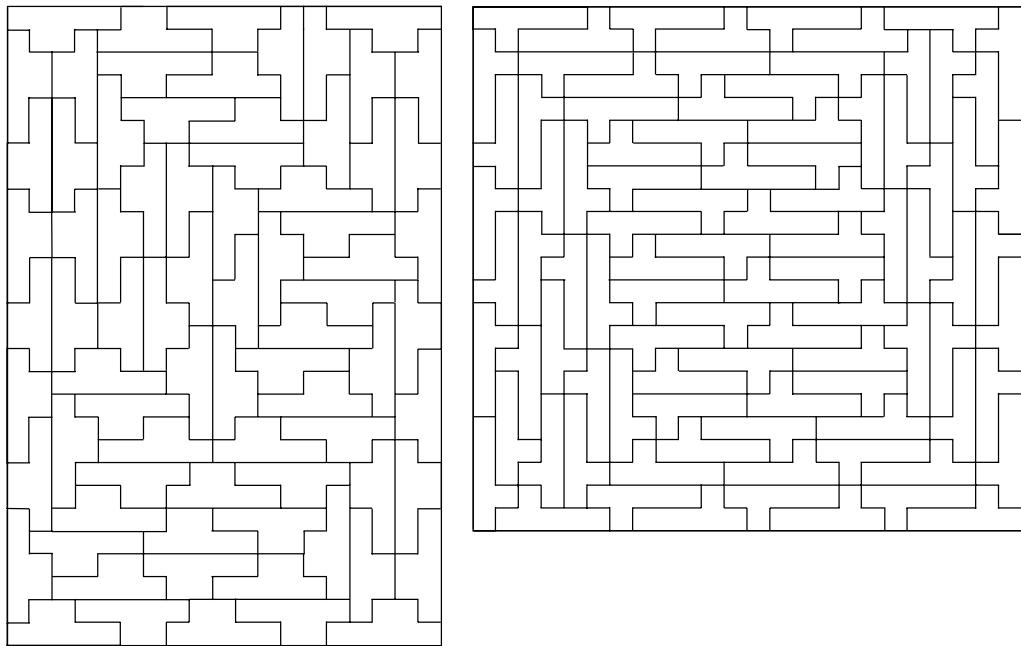
An 11-omino of order 50.

Figure 15.7.8 shows examples of orders 76 and 92, found by Karl A. Dahlke in 1987, but anticipated by T.W. Marlow in 1985.

The heptomino of order 76 in Figure 15.7.8 cannot tile its minimum rectangle with 180° rotational symmetry. This is also true of the dekomino in Figure 15.7.9 of order 96, whose minimum rectangle (the 30×32) was discovered by W.R. Marshall

FIGURE 15.7.8

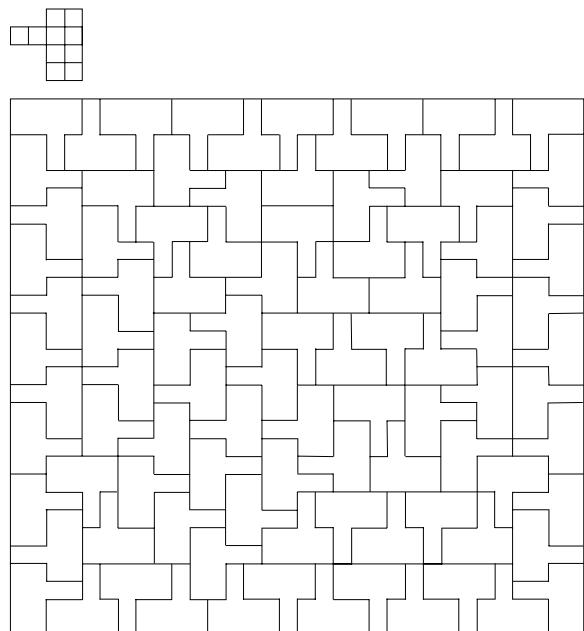
A heptomino of order 76 and a hexomino of order 92.



in 1991; in 1995, Marshall also found the minimum rectangles for the order 192 octomino (32×48) and the order 138 (30×46) dekomino [Mar95], later published in [Mar97].

FIGURE 15.7.9

A dekomino of order 96.

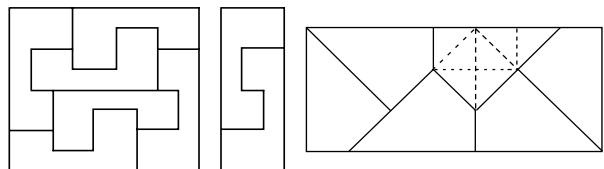


No polyomino whose order is an odd number greater than 1 has ever been found, but the possibility that such polyominoes exist (with orders greater than 3) has not been ruled out.

The known *even* orders of polyominoes are all the multiples of 4, as well as the numbers 2, 10, 18, 50, and 138. The smallest even order for which no example is known is 6. Figure 15.7.10 shows one way in which six copies of a polyomino can be fitted together to form a rectangle, but the polyomino in question (as shown) actually has order 2. Michael Reid found a *heptabolo* (a figure made of seven congruent isosceles right triangles) of order 6, also shown in Figure 15.7.10. (See also [Rei97].)

FIGURE 15.7.10

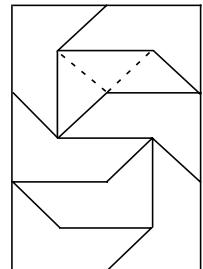
A 12-omino of order 2 that suggests an order-6 tiling, and Michael Reid's order-6 "heptabolo." (Is there any polyomino of order 6?)



The Golomb construction for polyominoes of order 4s gives its first new example, order 8, when $s = 2$. The underlying tiling concept of how to fit 8 congruent shapes together to form a rectangle is shown in Figure 15.7.11.

FIGURE 15.7.11

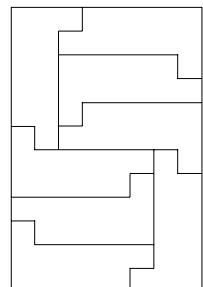
A rectangle formed from eight congruent pieces.



Although the shape used in Figure 15.7.11 is not a polyomino, the same concept can be realized using the 12-omino shown in Figure 15.7.12.

FIGURE 15.7.12

A polyomino of order 8.



PROBLEM 15.7.1

Given a polyomino, will it or won't it tile?

In recent years, whenever a specific polyomino whose ability to tile any rectangle had not yet been decided was publicized, someone with a good computer program has usually found a rectangle-tiling solution within a year. An infinite family of polyominoes, the first several of which are known to tile rectangles, as is every fourth one throughout the family, is illustrated in Figure 15.7.13. Two of the minimum rectangles, discovered in 1995 by Marshall (see [Mar97]) are shown in Figure 15.7.14. The general case is still open.

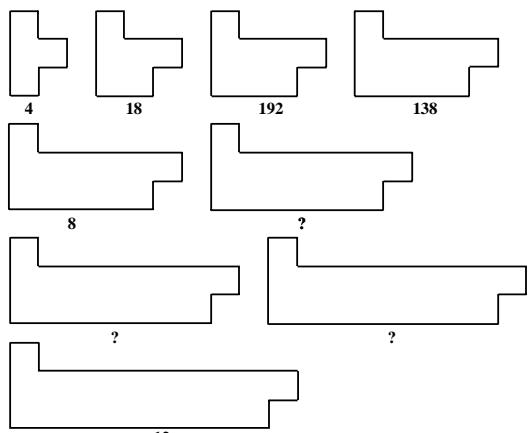
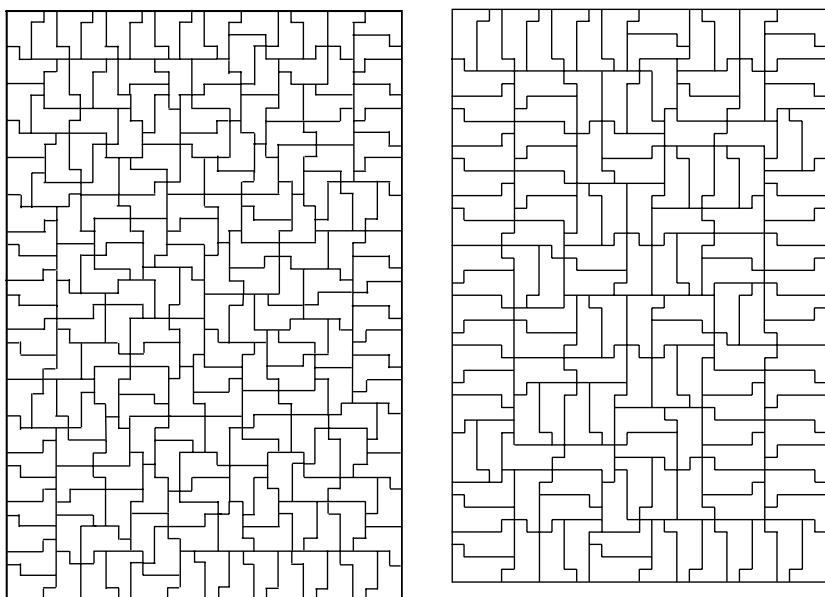


FIGURE 15.7.13

Infinite family of polyominoes. Does each one tile a rectangle? (The number below each figure is its order, if known.)

FIGURE 15.7.14

An octomino of order 192 and a dekomino of order 138.



15.8 SOURCES AND RELATED MATERIAL

FURTHER READING

A comprehensive recent survey of the subject, complete with an abundance of references, is [Gol94]. Another book on the subject is [Mar91]. Finally, there are a great many articles, puzzles, and problems concerning polyominoes to be found in the *Journal of Recreational Mathematics*.

RELATED CHAPTERS

[Chapter 3: Tilings](#)

REFERENCES

- [Ben74] E.A. Bender. Convex n -ominoes. *Discrete Math.*, 8:219–226, 1974.
- [Ber66] R. Berger. The undecidability of the domino problem. *Mem. Amer. Math. Soc.*, 66:1–72, 1966.
- [Bou94] M. Bousquet-Mélou. Polyominoes and polygons. *Contemp. Math.*, 178:55–70, 1994.
- [dBK75] N.G. de Bruijn and D.A. Klarner. A finite basis theorem for packing boxes with bricks. In *Papers Dedicated to C.J. Bouwkamp*, Philips Research Reports, 30:337–343, 1975.
- [Gol] S.W. Golomb. Personal communication.
- [Gol66] S.W. Golomb. Tiling with polyominoes. *J. Combin. Theory*, 1:280–296, 1966.
- [Gol89] S.W. Golomb. Polyominoes which tile rectangles. *J. Combin. Theory Ser. A*, 51:117–124, 1989.
- [Gol94] S.W. Golomb. *Polyominoes*, 2nd edition. Princeton Univ. Press, 1994.
- [Gol96] S.W. Golomb. Tiling rectangles with polyominoes. *Math. Intelligencer*, 18:38–47, 1996.
- [GJW00] A.J. Guttman, I. Jensen, L.H. Wong, and I.G. Enting. Punctured polygons and polyominoes on the square lattice. *J. Phys. A*, 33:1735–1764, 2000.
- [JG00] I. Jensen and A.J. Guttman. Statistics of lattice animals (polyominoes) and polygons. *J. Phys. A*, 33:L257–L263, 2000.
- [Kai95] D. Kaiser. Personal communication, 1995.
- [Kla67] D.A. Klarner. Cell growth problems. *Canad. J. Math.*, 19:851–863, 1967.
- [Kla69] D.A. Klarner. Packing a rectangle with congruent N -ominoes. *J. Combin. Theory*, 7:107–115, 1969.
- [Kla70] D.A. Klarner. A packing theory. *J. Combin. Theory*, 8:272–278, 1970.
- [KG69] D.A. Klarner and F. Göbel. Packing boxes with congruent figures. *Indag. Math.*, 31:465–472, 1969.
- [KR73] D.A. Klarner and R.L. Rivest. A procedure for improving the upper bound for the number of n -ominoes. *Canad. J. Math.*, 25:585–602, 1973.

- [KR74] D.A. Klarner and R.L. Rivest. Asymptotic bounds for the number of convex n -ominoes. *Discrete Math.*, 8:31–40, 1974.
- [KS] D.A. Klarner and W. Satterfield. The number of width- k n -ominoes. Unpublished.
- [Knu72] D.E. Knuth. Personal communication, 1972.
- [Mar90] W.R. Marshall. Personal communication, 1990.
- [Mar95] W.R. Marshall. Personal communications, 1995.
- [Mar97] W.R. Marshall. Packing rectangles with congruent polyominoes. *J. Combin. Theory Ser. A*, 77:181–192, 1997.
- [Mar91] G.E. Martin. *Polyominoes. A Guide to Puzzles and Problems in Tiling*. Math. Assoc. Amer., Washington, 1991.
- [Rea62] R.C. Read. Contributions to the cell growth problem. *Canad. J. Math.*, 14:1–20, 1962.
- [Red81] D.H. Redelmeier. Counting polyominoes: yet another attack. *Discrete Math.*, 36:191–203, 1981.
- [Rei97] M. Reid. Tiling rectangles and half strips with congruent polyominoes. *J. Combin. Theory Ser. A*, 80:106–123, 1997.
- [SW92] I. Stewart and A. Wormstein. Polyominoes of order 3 do not exist. *J. Combin. Theory Ser. A*, 61:130–136, 1992.

16 BASIC PROPERTIES OF CONVEX POLYTOPES

Martin Henk, Jürgen Richter-Gebert, and Günter M. Ziegler

INTRODUCTION

Convex polytopes are fundamental geometric objects that have been investigated since antiquity. The beauty of their theory is nowadays complemented by their importance for many other mathematical subjects, ranging from integration theory, algebraic topology, and algebraic geometry (toric varieties) to linear and combinatorial optimization.

In this chapter we try to give a short introduction, provide a sketch of “what polytopes look like” and “how they behave,” with many explicit examples, and briefly state some main results (where further details are in the subsequent chapters of this Handbook). We concentrate on two main topics:

- Combinatorial properties: faces (vertices, edges, . . . , facets) of polytopes and their relations, with special treatments of the classes of low-dimensional polytopes and polytopes with few vertices;
- Geometric properties: volume and surface area, mixed volumes, and quermassintegrals, including explicit formulas for the cases of the regular simplices, cubes, and cross-polytopes.

We refer to Grünbaum [Grü03] for a comprehensive view of polytope theory, and to Ziegler [Zie95] and Schneider [Sch93] for thorough treatments of the combinatorial (resp. convex geometric) aspects of polytope theory.

16.1 COMBINATORIAL STRUCTURE

GLOSSARY

\mathcal{V} -polytope: The convex hull of a finite set $X = \{x^1, \dots, x^n\}$ of points in \mathbb{R}^d :

$$P = \text{conv}(X) := \left\{ \sum_{i=1}^n \lambda_i x^i \mid \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1 \right\}.$$

\mathcal{H} -polytope: A bounded solution set of a finite system of linear inequalities:

$$P = P(A, b) := \{x \in \mathbb{R}^d \mid a_i^T x \leq b_i \text{ for } 1 \leq i \leq m\},$$

where $A \in \mathbb{R}^{m \times d}$ is a real matrix with rows a_i^T , and $b \in \mathbb{R}^m$ is a real vector with entries b_i . Here boundedness means that there is a constant N such that $\|x\| \leq N$ holds for all $x \in P$.

Polytope: A subset $P \subseteq \mathbb{R}^d$ that can be presented as a \mathcal{V} -polytope or (equivalently, by the main theorem below!) as an \mathcal{H} -polytope.

Dimension: The dimension of an arbitrary subset $S \subseteq \mathbb{R}^d$ is defined as the dimension of its affine hull: $\dim(S) := \dim(\text{aff}(S))$.

(Recall that $\text{aff}(S)$, the affine hull of a set S , is $\{\sum_{j=1}^p \lambda_j x^j \mid x^1, \dots, x^p \in S, \sum_{j=1}^p \lambda_j = 1\}$, the smallest affine subspace of \mathbb{R}^d containing S .)

d -polytope: A d -dimensional polytope. In what follows, a subscript in the name of a polytope usually denotes its dimension.

Interior and relative interior: The interior $\text{int}(P)$ is the set of all points $x \in P$ such that for some $\epsilon > 0$, the ϵ -ball $B_\epsilon(x)$ around x is contained in P .

Similarly, the relative interior $\text{relint}(P)$ is the set of all points $x \in P$ such that for some $\epsilon > 0$, the intersection $B_\epsilon(x) \cap \text{aff}(P)$ is contained in P .

Affine equivalence: For polytopes $P \subseteq \mathbb{R}^d$ and $Q \subseteq \mathbb{R}^e$, an affine map $\pi: \mathbb{R}^d \rightarrow \mathbb{R}^e$, $x \mapsto Ax + b$ mapping P bijectively to Q . π need not be injective or surjective. However, it has to restrict to a bijective map $\text{aff}(P) \rightarrow \text{aff}(Q)$. In particular, if P and Q are affinely equivalent, then they have the same dimension.

THEOREM 16.1.1 Main Theorem of Polytope Theory (cf. [Zie95, pp. 27])

The definitions of \mathcal{V} -polytopes and of \mathcal{H} -polytopes are equivalent. That is, every \mathcal{V} -polytope has a description by a finite system of inequalities, and every \mathcal{H} -polytope can be obtained as the convex hull of a finite set of points (its vertices).

Geometrically, a \mathcal{V} -polytope is the projection of an $(n-1)$ -dimensional simplex, while an \mathcal{H} -polytope is the bounded intersection of m closed halfspaces [Zie95, Lecture 1]. To see the main theorem at work, consider the following two statements: the first one is easy to see for \mathcal{V} -polytopes, but not for \mathcal{H} -polytopes, and for the second statement we have the opposite effect.

1. *Projections:* Every image of a polytope P under an affine map $\pi: x \mapsto Ax + b$ is a polytope.
2. *Intersections:* Any intersection of a polytope with an affine subspace is a polytope.

However, the computational step from one of the main theorem's descriptions of polytopes to the other—a “convex hull computation”—is far from trivial. Essentially, there are three types of algorithms available: inductive algorithms (inserting vertices, using a so-called beneath-beyond technique), projection resp. intersection algorithms (known as Fourier-Motzkin elimination resp. double description algorithms), and reverse search methods (as introduced by Avis and Fukuda). For explicit computations one can use public domain codes as integrated in the software package `polymake` [GJ00] that we use here; see also [Chapters 22](#) and [64](#).

In the following definitions of d -simplices, d -cubes, and d -cross-polytopes we give both a \mathcal{V} - and an \mathcal{H} -presentation in each case. From this one can see that the \mathcal{H} -presentation can have exponential “size” in terms of the size of the \mathcal{V} -presentation (e.g., for the d -cross-polytopes), and vice versa (for the d -cubes).

Definition: A (regular) d -dimensional **simplex** in \mathbb{R}^d is given by

$$T_d := \text{conv}\{e^1, e^2, \dots, e^d, \frac{1 - \sqrt{d+1}}{d}(e^1 + \dots + e^d)\}$$

$$= \left\{ x \in \mathbb{R}^d \mid \sum_{i=1}^d x_i \leq 1, -(1 + \sqrt{d+1}) + d)x_k + \sum_{i=1}^d x_i \leq 1 \text{ for } 1 \leq k \leq d \right\},$$

where e^1, \dots, e^d denotes the coordinate unit vectors in \mathbb{R}^d .

The simplices T_d are **regular polytopes** (with a symmetry group that is flag-transitive—see Chapter 19): the parameters have been chosen so that all edges of T_d have length $\sqrt{2}$. Furthermore, the origin $0 \in \mathbb{R}^d$ is in the interior of T_d : this is clear from the \mathcal{H} -presentation.

However, for the combinatorial theory one considers polytopes that differ only by a change of coordinates (an affine transformation) to be equivalent. Thus, we would refer to any d -polytope that can be presented as the convex hull of $d+1$ points as a **d -simplex**, since any two such polytopes are equivalent with respect to an affine map. Other standard choices include

$$\begin{aligned} \Delta_d &:= \text{conv}\{0, e^1, e^2, \dots, e^d\} \\ &= \left\{ x \in \mathbb{R}^d \mid \sum_{i=1}^d x_i \leq 1, x_k \geq 0 \text{ for } 1 \leq k \leq d \right\} \end{aligned}$$

and the $(d-1)$ -dimensional simplex in \mathbb{R}^d given by

$$\begin{aligned} \Delta'_{d-1} &:= \text{conv}\{e^1, e^2, \dots, e^d\} \\ &= \left\{ x \in \mathbb{R}^d \mid \sum_{i=1}^d x_i = 1, x_k \geq 0 \text{ for } 1 \leq k \leq d \right\}. \end{aligned}$$

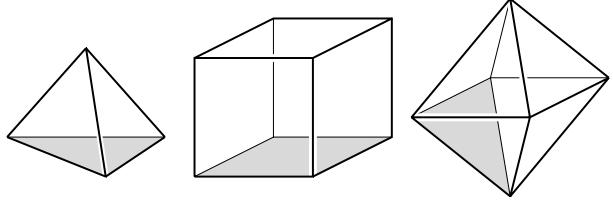


FIGURE 16.1.1
A 3-simplex, a 3-cube, and a 3-dimensional cross-polytope (octahedron).

Definition: A **d-cube** (a.k.a. the d -dimensional **hypercube**) is

$$\begin{aligned} C_d &:= \text{conv}\{\alpha_1 e^1 + \alpha_2 e^2 + \dots + \alpha_d e^d \mid \alpha_1, \dots, \alpha_d \in \{+1, -1\}\} \\ &= \left\{ x \in \mathbb{R}^d \mid -1 \leq x_k \leq 1 \text{ for } 1 \leq k \leq d \right\}, \end{aligned}$$

and a d -dimensional **cross-polytope** in \mathbb{R}^d (known as the **octahedron** for $d = 3$) is given by

$$C_d^\Delta := \text{conv}\{\pm e^1, \pm e^2, \dots, \pm e^d\} = \left\{ x \in \mathbb{R}^d \mid \sum_{i=1}^d |x_i| \leq 1 \right\}.$$

Again, there are other natural choices, among them

$$\begin{aligned} [0, 1]^d &= \text{conv}\left\{ \sum_{i \in S} e^i \mid S \subseteq \{1, 2, \dots, d\} \right\} \\ &= \left\{ x \in \mathbb{R}^d \mid 0 \leq x_k \leq 1 \text{ for } 1 \leq k \leq d \right\}, \end{aligned}$$

the d -dimensional ***unit cube***.

As another example to illustrate concepts and results we will occasionally use the unnamed polytope with six vertices shown in Figure 16.1.2.

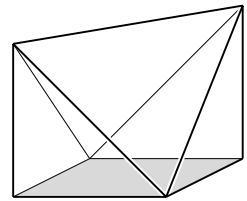


FIGURE 16.1.2

Our unnamed “typical” 3-polytope. It has 6 vertices, 11 edges, and 7 facets.

This polytope without a name can be presented as a \mathcal{V} -polytope by listing its six vertices. The following coordinates make it into a subpolytope of the 3-cube C_3 : the vertex set consists of all but two vertices of C_3 . Our list below (on the left) shows the vertices of our unnamed polytope in a format used as input for the **polymake** program, i.e., the vertices are given in homogeneous coordinates with an additional 1 as first entry. From these data the **polymake** program produces a description (on the right) of the polytope as an \mathcal{H} -polytope, i.e., it computes facets defining hyperplanes with respect to the homogeneous coordinates. For instance, the entries in the last row of the section FACETS describe the halfspace $1x_0 - 1x_1 + 1x_2 - 1x_3 \geq 0$ which corresponds to the facet-defining inequality $x_1 - x_2 + x_3 \leq 1$ of our 3-dimensional unnamed polytope.

POINTS	FACETS
1 1 1 1	1 0 -1 0
1 -1 -1 1	1 -1 0 0
1 1 1 -1	1 1 0 0
1 1 -1 -1	1 0 1 0
1 -1 1 -1	1 0 0 1
1 -1 -1 -1	1 1 -1 -1
	1 -1 1 -1

Unbounded polyhedra can, via projective transformations, be treated as polytopes with a distinguished facet (see [Zie95, p. 75]). In this respect, we do not lose anything on the combinatorial level if we restrict the following discussion to the setting of full-dimensional convex polytopes: d -polytopes embedded in \mathbb{R}^d .

16.1.1 FACES

GLOSSARY

Support function: Given a polytope $P \subseteq \mathbb{R}^d$, the function

$$h(P, \cdot): \mathbb{R}^d \rightarrow \mathbb{R}, \quad h(P, x) := \sup\{\langle x, y \rangle \mid y \in P\},$$

where $\langle x, y \rangle$ denotes the inner product on \mathbb{R}^d . (Since P is compact one may replace sup by max.)

For $v \in \mathbb{R}^d \setminus \{0\}$ the hyperplane

$$H(P, v) := \{x \in \mathbb{R}^d \mid \langle x, v \rangle = h(P, v)\}$$

is the **supporting hyperplane** of P with **outer normal vector** v . Note that $H(P, \mu v) = H(P, v)$ for $\mu \in \mathbb{R}$, $\mu > 0$. For a vector u of the $(d-1)$ -dimensional **unit sphere** S^{d-1} , $h(P, u)$ is the signed distance of the supporting plane $H(P, u)$ from the origin. (For $v = 0$ we set $H(P, 0) := \mathbb{R}^d$, which is not a hyperplane.)

The intersection of P with a supporting hyperplane $H(P, v)$ is called a (nontrivial) **face**, or more precisely a **k -face** if the dimension of $\text{aff}(P \cap H(P, v))$ is k . Each face is itself a polytope.

The set of all k -faces is denoted by $\mathcal{F}_k(P)$ and its cardinality by $f_k(P)$.

f -vector: The vector of face numbers $\mathbf{f}(P) = (f_0(P), f_1(P), \dots, f_{d-1}(P))$ associated with a d -polytope.

The empty set \emptyset and the polytope P itself are considered **trivial faces** of P , of dimensions -1 and $\dim(P)$, respectively. All faces other than P are **proper faces**.

The faces of dimension 0 and 1 are called **vertices** and **edges**, respectively. The $(\dim(P)-1)$ -faces of P are called **facets**.

Facet-vertex incidence matrix: The matrix $M \in \{0, 1\}^{f_{d-1}(P) \times f_0(P)}$ that has an entry $M(F, v) = 1$ if the facet F contains the vertex v , and $M(F, v) = 0$ otherwise.

Graded poset: A partially ordered set (P, \leq) with a unique minimal element $\hat{0}$, a unique maximal element $\hat{1}$, and a **rank function** $r: P \rightarrow \mathbb{N}_0$ that satisfies
(1) $r(\hat{0}) = 0$, and $p < p'$ implies $r(p) < r(p')$, and
(2) $p < p'$ and $r(p') - r(p) > 1$ implies that there is a $p'' \in P$ with $p < p'' < p'$.

Lattice L : A partially ordered set (P, \leq) in which every pair of elements $p, p' \in P$ has a unique maximal lower bound, called the **meet** $p \wedge p'$, and a unique minimal upper bound, called the **join** $p \vee p'$.

Atom, coatom: If L is a graded lattice, the minimal elements of $L \setminus \{\hat{0}\}$ (i.e., the elements of rank 1) are the atoms of L . Similarly, the maximal elements of $L \setminus \{\hat{1}\}$ (i.e., the elements of rank $r(\hat{1})-1$) are the coatoms of L . A graded lattice is **atomic** if every element is a join of a set of atoms, and it is **coatomic** if every element is a meet of a set of coatoms.

Face lattice $L(P)$: The set of all faces of P , partially ordered by inclusion.

Combinatorially isomorphic: Polytopes whose face lattices are isomorphic as abstract (unlabeled) partially ordered sets/lattices.

Equivalently, P and P' are combinatorially equivalent if their facet-vertex incidence matrices differ only by column and row permutations.

Combinatorial type: An equivalence class of polytopes under combinatorial equivalence.

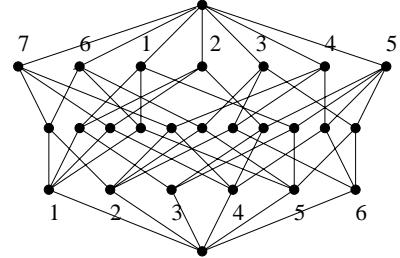
THEOREM 16.1.2 Face Lattices of Polytopes (cf. [Zie95, pp. 51])

The face lattices of convex polytopes are finite, graded, atomic, and coatomic lattices. The meet operation $G \wedge H$ is given by intersection, while the join $G \vee H$ is the intersection of all facets that contain both G and H . The rank function on $L(P)$ is given by $r(G) = \dim(G) + 1$.

The minimal nonempty faces of a polytope are its vertices: they correspond to atoms of the lattice $L(P)$. Every face is the join of its vertices, hence $L(P)$ is atomic. Similarly, the maximal proper faces of a polytope are its facets: they correspond to the coatoms of $L(P)$. Every face is the intersection of the facets it is contained in, hence face lattices of polytopes are coatomic.

FIGURE 16.1.3

The face lattice of our unnamed 3-polytope. The 7 coatoms (facets) and the 6 atoms (vertices) have been labeled in the order of their appearance in the lists on page 358. Thus, the downwards-path from the coatom “4” to the atom “2” represents the fact that the fourth facet contains the second vertex.



The face lattice is a complete encoding of the combinatorial structure of a polytope. However, in general the encoding by a facet-vertex incidence matrix is more efficient. The following matrix—also provided by **polymake**—represents our unnamed 3-polytope:

$$M = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 0 & 1 & 1 \\ 4 & 0 & 1 & 0 & 1 & 0 & 1 \\ 5 & 0 & 0 & 1 & 1 & 1 & 1 \\ 6 & 1 & 1 & 0 & 0 & 1 & 0 \\ 7 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

How do we decide whether a set of vertices $\{v^1, \dots, v^k\}$ is (the vertex set of) a face of P ? This is the case if and only if no other vertex v^0 is contained in all the facets that contain $\{v^1, \dots, v^k\}$. This criterion makes it possible, for example, to derive the edges of a polytope P from a facet-vertex matrix.

For low-dimensional polytopes, the criterion can be simplified: if $d \leq 4$, then two vertices are connected by an edge if and only if there are at least $d-1$ different facets that contain them both. However, the same is not true any longer for 5-dimensional polytopes, where vertices may be nonadjacent despite being contained in many common facets. (The best way to see this is by using polarity; see below.)

16.1.2 POLARITY

GLOSSARY

Polarity: If $P \subseteq \mathbb{R}^d$ is a d -polytope with the origin in its interior, then the *polar* of P is the d -polytope

$$P^\Delta := \{y \in \mathbb{R}^d \mid \langle y, x \rangle \leq 1 \text{ for all } x \in P\}.$$

Stellar subdivision: The stellar subdivision of a polytope P in a face F is the polytope $\text{conv}(P \cup x^F)$, where x^F is a point of the form $y^F - \epsilon(y^P - y^F)$, where y^P is in the interior of P , y^F is in the relative interior of F , and ϵ is small enough.

Vertex figure P/v : If v is a vertex of P , then $P/v := P \cap H$ is the polytope obtained by intersecting P with a hyperplane H that has v on one side and all the other vertices of P on the other side.

Cutting off a vertex: The polytope $P \cap H^-$ obtained by intersecting P with a closed halfspace H^- that does not contain the vertex v , but contains all other vertices of P in its interior. (In this situation, $P \cap H^+$ is a pyramid over the vertex figure P/v .)

Quotient of P : A polytope obtained from P by taking vertex figures (possibly several times).

Simplicial polytope: A polytope all of whose facets (equivalently, proper faces) are simplices.

Simple polytope: A polytope all of whose vertex figures (equivalently, proper quotients) are simplices.

Polarity is a fundamental construction in the theory of polytopes. One always has $P^{\Delta\Delta} = P$, under the assumption that P has the origin in its interior. This condition can always be obtained after a change of coordinates. In particular, we speak of (combinatorial) polarity between d -polytopes Q and R that are combinatorially isomorphic to P and P^Δ , respectively.

Any \mathcal{V} -presentation of P yields an \mathcal{H} -presentation of P^Δ , and conversely, via

$$P = \text{conv}\{v^1, \dots, v^n\} \iff P^\Delta = \{x \in \mathbb{R}^d \mid \langle v^i, x \rangle \leq 1 \text{ for } 1 \leq i \leq n\}.$$

There are basic relations between polytopes and polytopal constructions under polarity. For example, the fact that the d -cross-polytopes C_d^Δ are the polars of the d -cubes C_d is built into our notation. More generally, the polars of simple polytopes are simplicial, and conversely. This can be deduced from the fact that the facets F of a polytope P correspond to the vertex figures P^Δ/v of its polar P^Δ . In fact, F and P^Δ/v are combinatorially polar in this situation. More generally, one has a correspondence between faces and quotients under polarity.

At a combinatorial level, all this can be derived from the fact that the face lattices $L(P)$ and $L(P^\Delta)$ are anti-isomorphic: $L(P^\Delta)$ may be obtained from $L(P)$ by reversing the order relations. Thus, lower intervals in $L(P)$, corresponding to faces of P , translate under polarity into upper intervals of $L(P^\Delta)$, corresponding to quotients of P^Δ .

16.1.3 BASIC CONSTRUCTIONS

GLOSSARY

For the following constructions, let

$P \subseteq \mathbb{R}^d$ be a d -dimensional polytope with n vertices and m facets, and
 $P' \subseteq \mathbb{R}^{d'}$ a d' -dimensional polytope with n' vertices and m' facets.

Scalar multiple: For $\lambda \in \mathbb{R}$, the scalar multiple λP is defined by $\lambda P := \{\lambda x \mid x \in P\}$. P and λP are combinatorially (in fact, affinely) isomorphic for all $\lambda \neq 0$. In particular, $(-1)P = -P = \{-p \mid p \in P\}$, and $(+1)P = P$.

Minkowski sum: $P + P' := \{p + p' \mid p \in P, p' \in P'\}$.

It is also useful to define the difference as $P - P' = P + (-P')$. The polytopes $P + \lambda P'$ are combinatorially isomorphic for all $\lambda > 0$, and similarly for $\lambda < 0$.

If $P' = \{p'\}$ is one single point, then $P - \{p'\}$ is the image of P under the translation that takes p' to the origin.

Product: The $(d+d')$ -dimensional polytope $P \times P' := \{(p, p') \in \mathbb{R}^{d+d'} \mid p \in P, p' \in P'\}$. $P \times P'$ has $n \cdot n'$ vertices and $m + m'$ facets.

Join: The convex hull $P * P'$ of $P \cup P'$, after embedding P and P' in a space where their affine hulls are skew. For example,

$$P * P' := \text{conv}(\{(p, 0, 0) \in \mathbb{R}^{d+d'+1} \mid p \in P\} \cup \{(0, p', 1) \in \mathbb{R}^{d+d'+1} \mid p' \in P'\}).$$

$P * P'$ has dimension $d+d'+1$ and $n+n'$ vertices. Its k -faces are the joins of i -faces of P and $(k-i-1)$ -faces of P' , hence $f_k(P * P') = \sum_{i=-1}^k f_i(P) f_{k-i-1}(P')$.

Free sum: The free sum is the $(d+d')$ -dimensional polytope

$$P \oplus P' := \text{conv}(\{(p, 0) \in \mathbb{R}^{d+d'} \mid p \in P\} \cup \{(0, p') \in \mathbb{R}^{d+d'} \mid p' \in P'\}).$$

Thus the free sum $P \oplus P'$ is a projection of the join $P * P'$. If both P and P' have the origin in their interiors—this is the “usual” situation for creating free sums, then $P \oplus P'$ has $n + n'$ vertices and $m \cdot m'$ facets.

Pyramid: The join $\text{pyr}(P) := P * \{0\}$ of P with a point (a 0-dimensional polytope $P' = \{0\} \subseteq \mathbb{R}^0$). The pyramid $\text{pyr}(P)$ has $n + 1$ vertices and $m + 1$ facets.

Prism: The product $\text{prism}(P) := P \times I$, where I denotes the real interval $I = [-1, +1] \subseteq \mathbb{R}$.

Bipyramid: If P has the origin in its interior, then the bipyramid over P is the $(d+1)$ -dimensional polytope constructed as the free sum $\text{bipyr}(P) := P \oplus I$.

Lawrence extension: If $p \in \mathbb{R}^d$ is a point outside the polytope P , then the free sum $(P - \{p\}) \oplus [1, 2]$ is a *Lawrence extension of P at p* . (For $p \in P$ this is just a pyramid.)

Of course, the many constructions listed in the glossary above are not independent of each other. For instance, some of these constructions are related by polarity: for polytopes P and P' with the origin in their interiors, the product and the free sum constructions are related by polarity,

$$P \times P' = (P^\Delta \oplus P'^\Delta)^\Delta,$$

and this specializes to polarity relations among the pyramid, bipyramid, and prism constructions,

$$\text{pyr}(P) = (\text{pyr}(P^\Delta))^\Delta \quad \text{and} \quad \text{prism}(P) = (\text{bipyr}(P^\Delta))^\Delta.$$

Similarly, “cutting off a vertex” is polar to “stellar subdivision in a facet.”

It is interesting to study—and this has not been done systematically—how the basic polytope operations generate complicated convex polytopes from simpler ones. For example, starting from a one-dimensional polytope $I = C_1 = [-1, +1] \subseteq \mathbb{R}$, the

direct product construction generates the cubes C_d , while free sums generate the cross-polytopes C_d^Δ .

Even more complicated centrally symmetric polytopes, the *Hanner polytopes*, are obtained from copies of the interval I by using products and free sums. They are interesting since they achieve with equality the conjectured bound that all centrally symmetric d -polytopes have at least 3^d nonempty faces (Kalai [Kal89]).

Every polytope can be viewed as a region of a hyperplane arrangement: for this, take as \mathcal{A}_P the set of all hyperplanes of the form $\text{aff}(F)$, where F is a facet of P . For additional points, such as the points outside the polytope used for Lawrence extensions, or those used for stellar subdivisions, it is often important only in which region, or in which lower-dimensional region, of the arrangement \mathcal{A}_P they lie.

The Lawrence extension, by the way, may seem like quite a harmless little construction. However, it has the amazing property that it can encode the structure of a point *outside* a d -polytope into the boundary structure of a $(d+1)$ -polytope. This accounts for a large part of the “special” 4- and 5-polytopes in the literature, such as the 4-polytopes for which a facet, or even a 2-face, cannot be prescribed in shape [Ric96].

16.1.4 MORE EXAMPLES

There are many interesting classes of polytopes arising from diverse areas of mathematics (as well as physics, optimization, crystallography, etc.). Some of these are discussed below. You will find many more classes of examples discussed in other chapters of this Handbook. For example, regular and semiregular polytopes are discussed in [Chapter 19](#), while polytopes that arise as Voronoi cells of lattices appear in [Chapters 3, 7, and 62](#).

GLOSSARY

Graph of a polytope: The graph $G(P) = (V(P), E(P))$ with vertex set $V(P) = \mathcal{F}_0(P)$ and edge set $E(P) = \{\{v^1, v^2\} \subseteq \binom{V}{2} \mid \text{conv}\{v^1, v^2\} \in \mathcal{F}_1(P)\}$.

Zonotope: Any polytope Z that can be represented as the image of an n -dimensional cube C_n under an affine map; equivalently, any polytope that can be written as a Minkowski sum of n line segments (1-dimensional polytopes). The smallest n such that Z is an image of C_n is the *number of zones* of Z .

Moment curve: The curve γ in \mathbb{R}^d defined by $\gamma : \mathbb{R} \rightarrow \mathbb{R}^d$, $t \mapsto (t, t^2, \dots, t^d)^T$.

Cyclic polytope: The convex hull of a finite set of points on a moment curve, or any polytope combinatorially equivalent to it.

k -neighborly polytope: A polytope such that each subset of at most k vertices forms the vertex set of a face. Thus every polytope is 1-neighborly, and a polytope is 2-neighborly if and only if its graph is complete.

Neighborly polytope: A d -dimensional polytope that is $\lfloor d/2 \rfloor$ -neighborly.

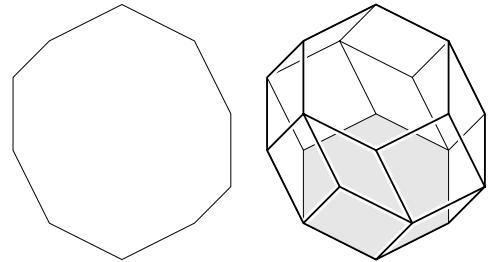
(0,1)-polytope: A polytope all of whose vertex coordinates are 0 or 1, that is, whose vertex set is a subset of the vertex set $\{0, 1\}^d$ of the unit cube.

ZONOTOPES

Zonotopes appear in quite different guises. They can equivalently be defined as the Minkowski sums of finite sets of line segments (1-dimensional polytopes), as the affine projections of d -cubes, or as polytopes all of whose faces (equivalently, all 2-faces) exhibit central symmetry. Thus a 2-dimensional polytope is a zonotope if and only if it is centrally symmetric.

FIGURE 16.1.4

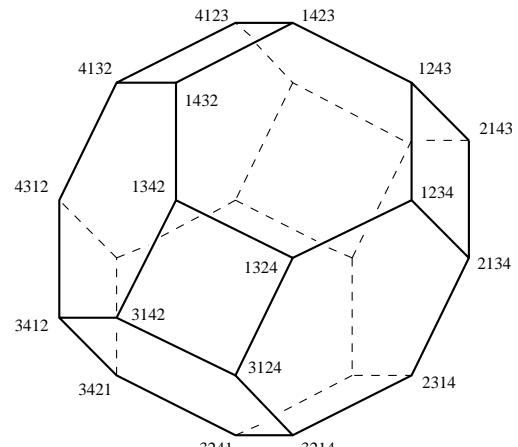
A 2-dimensional and a 3-dimensional zonotope, each with 5 zones. (The 2-dimensional one is a projection of the 3-dimensional one; note that every projection of a zonotope is a zonotope.)



Among the most prominent zonotopes are the permutohedra: The **permutohedron** Π_{d-1} is constructed by taking the convex hull of all d -vectors whose coordinates are $\{1, 2, \dots, d\}$, in any order. The permutohedron Π_{d-1} is a $(d-1)$ -dimensional polytope (contained in the hyperplane $\{x \in \mathbb{R}^d \mid \sum_{i=1}^d x_i = d(d+1)/2\}$) with $d!$ vertices and $2^d - 2$ facets.

FIGURE 16.1.5

The 3-dimensional permutohedron Π_3 . The vertices are labeled by the permutations that, when applied to the coordinate vector in \mathbb{R}^4 , yield $(1, 2, 3, 4)^T$.



One unusual feature of permutohedra is that they are simple zonotopes: these are rare in general, and the (unsolved) problem of classifying them is equivalent to the problem of classifying all simplicial arrangements of hyperplanes (see [Section 6.3.3](#)).

Zonotopes are important because their theory is equivalent to the theories of vector configurations (realizable oriented matroids) and of hyperplane arrange-

ments. In fact, the system of line segments that generates a zonotope can be considered as a vector configuration, and the hyperplanes that are orthogonal to the line segments provide the associated hyperplane arrangement. We refer to [BLS⁺99, Section 2.2] and [Zie95, Lecture 7].

Finally, we mention in passing a surprising bijective correspondence between the tilings of a zonotope with smaller zonotopes and oriented matroid liftings (realizable or not) of the oriented matroid of a zonotope. This correspondence is known as the *Bohne-Dress theorem*; we refer to Richter-Gebert and Ziegler [RZ94].

CYCLIC POLYTOPES

Cyclic polytopes can be constructed by taking the convex hull of $n > d$ points on the moment curve in \mathbb{R}^d . The “standard construction” is to define a cyclic polytope $C_d(n)$ as the convex hull of n integer points on this curve, such as

$$C_d(n) := \text{conv}\{\gamma(1), \gamma(2), \dots, \gamma(n)\}.$$

However, the combinatorial type of $C_d(n)$ is given by the—entirely combinatorial—**Gale evenness criterion**: If $C_d(n) = \text{conv}\{\gamma(t_1), \dots, \gamma(t_n)\}$, with $t_1 < \dots < t_n$, then $\gamma(t_{i_1}), \dots, \gamma(t_{i_d})$ determine a facet if and only if the number of indices in $\{i_1, \dots, i_d\}$ lying between any two indices *not* in that set is even. Thus, the combinatorial type does not depend on the specific choice of points on the moment curve [Zie95, Example 0.6; Theorem 0.7].

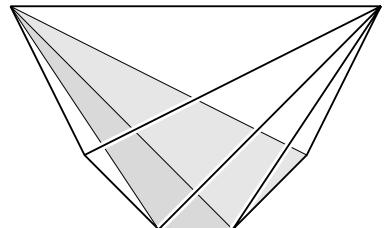


FIGURE 16.1.6

A 3-dimensional cyclic polytope $C_3(6)$ with 6 vertices. (In a projection of γ to the x_1x_2 -plane, the curve γ and hence the vertices of $C_3(6)$ lie on the parabola $x_2 = x_1^2$.)

The first property of cyclic polytopes to notice is that they are simplicial. The second, more surprising, property is that they are neighborly. This implies that among all d -polytopes P with n vertices, the cyclic polytopes maximize the number $f_i(P)$ of i -dimensional faces for $i < \lfloor d/2 \rfloor$. The same fact holds for all i : this is part of McMullen’s upper bound theorem (see below). In particular, cyclic polytopes have a very large number of facets,

$$f_{d-1}(C_d(n)) = \binom{n - \lceil \frac{d}{2} \rceil}{\lfloor \frac{d}{2} \rfloor} + \binom{n - 1 - \lceil \frac{d-1}{2} \rceil}{\lfloor \frac{d-1}{2} \rfloor}.$$

For example, we get that a cyclic 4-polytope $C_4(n)$ has $n(n - 3)/2$ facets. Thus $C_4(8)$ has 8 vertices, any two of them adjacent, and 20 facets. This is more than the 16 facets of the 4-dimensional cross-polytope, which also has 8 vertices!

NEIGHBORLY POLYTOPES

Here are a few observations about neighborly polytopes. For more information, see [BLS⁺99, Section 9.4] and the references quoted there.

The first observation is that if a polytope is k -neighborly for some $k > \lfloor d/2 \rfloor$, then it is a simplex. Thus, if one ignores the simplices, then $\lfloor d/2 \rfloor$ -neighborly polytopes form the extreme case, which motivates calling them simply “neighborly.” However, only in even dimensions $d = 2m$ do the neighborly polytopes have very special structure. For example, one can show that even-dimensional neighborly polytopes are necessarily simplicial, but this is not true in general. For the latter, note that, for example, all 3-dimensional polytopes are neighborly by definition, and that if P is a neighborly polytope of dimension $d = 2m$, then $\text{pyr}(P)$ is neighborly of dimension $2m+1$.

All simplicial neighborly d -polytopes with n vertices have the same number of facets (in fact, the same f -vector $(f_0, f_1, \dots, f_{d-1})$) as $C_d(n)$. They constitute the class of polytopes with the maximal number of i -faces for all i : this is the statement of McMullen’s upper bound theorem. We refer to [Chapter 18](#) for a thorough discussion of f -vector theory.

For $n \leq d+3$, every neighborly polytope is combinatorially isomorphic to a cyclic polytope. This covers, for instance, the polar of the product of two triangles, $(\Delta_2 \times \Delta_2)^\Delta$, which is easily seen to be a 4-dimensional neighborly polytope with 6 vertices; see [Figure 16.1.9](#). The first example of an even-dimensional neighborly polytope that is not cyclic appears for $d = 4$ and $n = 8$. It can easily be described in terms of its affine Gale diagram; see below.

Neighborly polytopes may at first glance seem to be very peculiar and rare objects, but there are several indications that they are not quite as unusual as they seem. In fact, the class of neighborly polytopes is believed to be very rich. Thus, Shemer [She82] has shown that for fixed even d the number of nonisomorphic neighborly d -polytopes with n vertices grows superexponentially with n . Also, many of the $(0,1)$ -polytopes studied in combinatorial optimization turn out to be at least 2-neighborly. Both these effects illustrate that “neighborliness” is not an isolated phenomenon.

OPEN PROBLEMS

1. Can every neighborly d -polytope $P \subseteq \mathbb{R}^d$ with n vertices be extended by a new vertex $v \in \mathbb{R}^d$ to a neighborly polytope $P' := \text{conv}(P \cup \{v\})$ with $n+1$ vertices? [She82, p. 314]
2. It is a classic problem of Perles whether every simplicial polytope is a quotient of a neighborly polytope. (For polytopes with at most $d+4$ vertices this was confirmed by Kortenkamp [Kor97].)
3. In some models of random polytopes it seems that
 - one obtains a neighborly polytope with high probability (which increases rapidly with the dimension of the space),
 - the most probable combinatorial type is a cyclic polytope,
 - but still this probability of a cyclic polytope tends to zero.

However, none of this has been proved. (See Bokowski and Sturmfels [BS89, p. 101], Bokowski, Richter-Gebert, and Schindler [BRS92], and Vershik and Sporyshev [VS92].)

(0,1)-POLYTOPES

There is a (0, 1)-polytope (given in terms of a \mathcal{V} -presentation) associated with every finite set system $\mathcal{S} \subseteq 2^E$ (where E is a finite set, and 2^E denotes the collection of all of its subsets), via

$$P[\mathcal{S}] := \text{conv}\left\{\sum_{i \in F} e^i \mid F \in \mathcal{S}\right\} \subseteq \mathbb{R}^E.$$

In combinatorial optimization, there is an extensive literature available on \mathcal{H} -presentations of special (0, 1)-polytopes, such as

- the ***traveling salesman polytopes*** T^n , where E is the edge set of a complete graph K_n , and \mathcal{F} is the set of all $(n-1)!$ Hamilton cycles (simple circuits through all the vertices) in E (see Grötschel and Padberg [GP85]);
- the ***cut and equicut polytopes***, where E is again the edge set of a complete graph, and \mathcal{S} represents, for example, the family of all cuts, or all equicuts, of the graph (see Deza and Laurent [DL97]).

Besides their importance for combinatorial optimization, there is a great deal of interesting polytope theory associated with such polytopes. For a striking example, see the equicut polytopes used by Kahn and Kalai [KK93] in their disproof of Borsuk's conjecture (see also [AZ01]).

Despite the detailed structure theory for the “special” (0, 1)-polytopes of combinatorial optimization, there is very little known about “general” (0, 1)-polytopes. For example, what is the “typical”, or the maximal, number of facets of a (0, 1)-polytope? Based on a random construction Bárány and Pór [BP01] proved the existence of d -dimensional (0, 1)-polytopes with $(cd/\log d)^{d/4}$ facets, where c is a universal constant. The best known upper bounds are of order $(d-2)!$. Another question, which is not only intrinsically interesting but might also provide new clues for basic questions of linear and combinatorial optimization, is: What is the maximal number of faces in a 2-dimensional projection of a (0, 1)-polytope? For a survey on (0, 1)-polytopes see [Zie00].

16.1.5 THREE-DIMENSIONAL POLYTOPES AND PLANAR GRAPHS

GLOSSARY

d-connected graph: A connected graph that remains connected if any $d-1$ vertices are deleted.

Drawing of a graph: A representation in the plane where the vertices are represented by distinct points, and simple Jordan arcs are drawn between the pairs of adjacent vertices.

Planar graph: A graph that can be drawn in the plane with Jordan arcs that are disjoint except for their endpoints.

Realization space: The set of all coordinatizations of a combinatorial structure, modulo affine coordinate transformations. (See [Section 6.3.2](#).)

Isotopy property: A combinatorial structure (such as a combinatorial type of polytope) has the isotopy property if any two realizations can be deformed into each other continuously, while maintaining the combinatorial type. Equivalently, the isotopy property holds for a combinatorial structure if and only if its realization space is connected.

THEOREM 16.1.3 Steinitz's Theorem [SR34]

For every 3-dimensional polytope P , the graph $G(P)$ is a planar, 3-connected graph. Conversely, for every planar 3-connected graph, there is a unique combinatorial type of 3-polytope P with $G(P) \cong G$.

Furthermore, the realization space $\mathcal{R}(P)$ of a combinatorial type of 3-polytope is homeomorphic to $\mathbb{R}^{f_1(P)-6}$, and contains rational points. In particular, 3-dimensional polytopes have the isotopy property, and they can be realized with integer vertex coordinates.

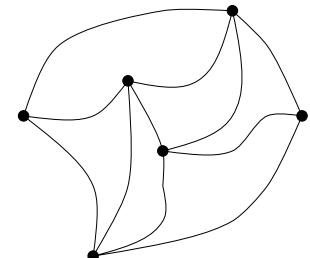


FIGURE 16.1.7

A (planar drawing of a) 3-connected, planar, unnamed graph. The formidable task of any proof of Steinitz's theorem is to construct a 3-polytope with this graph.

There are two essentially different ways known to prove Steinitz's theorem. The first one [SR34] provides a construction sequence for any type of 3-polytope, starting from a tetrahedron, and using only local operations such as cutting off vertices and polarity. The second type of proof realizes any combinatorial type by a global minimization argument, which as an intermediate step provides a special planar representation of the graph by a framework with a positive self-stress [McM94, OS94].

OPEN PROBLEMS

Because of Steinitz's theorem and its extensions and corollaries, the theory of 3-dimensional polytopes is quite complete and satisfactory. Nevertheless, some basic open problems remain.

1. It can be shown that every combinatorial type of 3-polytope with n vertices and a triangular facet can be realized with integer coordinates belonging to $\{1, 2, \dots, 37^n\}^3$ (J. Richter-Gebert and G. Stein, improving on Onn and Sturmfels [OS94]), but it is not clear whether the bound of 37^n can be replaced by a polynomial bound.
2. If P has a group G of symmetries, then it also has a symmetric realization.

However, it is not clear whether the space of all G -symmetric realizations $\mathcal{R}^G(P)$ is still homeomorphic to some \mathbb{R}^k . (It does not contain rational points in general, e.g., for the icosahedron!)

16.1.6 FOUR-DIMENSIONAL POLYTOPES AND SCHLEGEL DIAGRAMS

GLOSSARY

Schlegel diagram: A $(d-1)$ -dimensional representation $\mathcal{D}(P, F)$ of a d -dimensional polytope P , obtained as follows. Take a point of view very close to (an interior point of) the facet F , and let $\mathcal{D}(P, F)$ be the decomposition of F given by all the other facets of P , as seen from this point of view.

$(d-1)$ -diagram: A polytopal decomposition \mathcal{D} of a $(d-1)$ -polytope F such that

- (1) \mathcal{D} is a polytopal complex (i.e., a finite collection of polytopes closed under taking faces, such that any intersection of two polytopes in the complex is a face of each), and
- (2) the intersection of any polytope in \mathcal{D} with the boundary of F is a face of F (which may be empty).

Basic primary semialgebraic set defined over \mathbb{Z} : The solution set $S \subseteq \mathbb{R}^k$ of a finite set of equations and strict inequalities of the form $f_i(x) = 0$ resp. $g_j(x) > 0$, where the f_i and g_j are polynomials in k variables with integer coefficients.

Stable equivalence: Equivalence relation between semialgebraic sets generated by rational changes of coordinates and certain types of “stable” projections with contractible fibers. (See Richter-Gebert [Ric96, Section 2.5].)

In particular, if two sets are stably equivalent, then they have the same homotopy type, and they have the same arithmetic properties with respect to subfields of \mathbb{R} ; e.g., either both or neither of them contain a rational point.

The situation for 4-polytopes is fundamentally different from that for 3-dimensional polytopes. One reason is that there is no similar reduction of 4-polytope theory to a combinatorial (graph) problem.

The main results about graphs of d -polytopes are that they are d -connected (Balinski [Ba61]), and that each contains a subdivision of the complete graph on $d+1$ vertices, $K_{d+1} = G(T_d)$ (Grünbaum, [Grü03, pp. 200]). In particular, all graphs of 4-polytopes are 4-connected, and none of them is planar. (See also Chapter 20.)

Schlegel diagrams provide a reasonably efficient tool for visualization of 4-polytopes: we have a fighting chance to understand some of their theory in terms of the 3-dimensional (!) geometry of Schlegel diagrams.

A $(d-1)$ -diagram is a polytopal complex that “looks like” a Schlegel diagram, although there are diagrams (even 2-diagrams) that are not Schlegel diagrams. The situation is somewhat nicer for *simple* 4-polytopes. These are determined by their graphs (Blind and Mani-Levitska [BM87], and for a wonderful proof see Kalai [Kal88]), and they can be understood in terms of 3-diagrams: all simple 3-diagrams are projections of genuine 4-dimensional polytopes (Whiteley, see Rybníkov [Ryb99]).

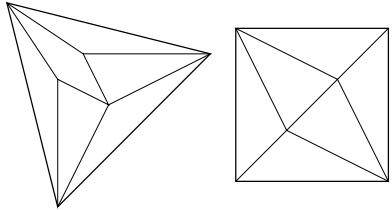


FIGURE 16.1.8

Two Schlegel diagrams of our unnamed 3-polytope, the first based on a triangle facet, the second on the “bottom square.”

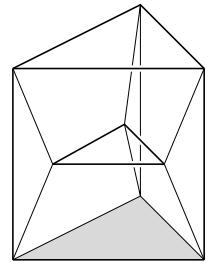


FIGURE 16.1.9

A Schlegel diagram of the product of two triangles. (This is a 4-dimensional polytope with 6 triangular prisms as facets, any two of them adjacent!)

The fundamental difference between the theories for polytopes in dimensions 3 and 4 is most apparent in the contrast between Steinitz’s theorem and the following result, which states simply that all the “nice” properties of 3-polytopes established in Steinitz’s theorem fail dramatically for 4-dimensional polytopes.

THEOREM 16.1.4 Richter-Gebert’s Universality Theorem for 4-Polytopes [Ric96]

The realization space of a 4-dimensional polytope can be “arbitrarily wild”: for every basic primary semialgebraic set S defined over \mathbb{Z} there is a 4-dimensional polytope $P[S]$ whose realization space $\mathcal{R}(P[S])$ is stably equivalent to S .

In particular, this implies the following.

- *The isotopy property fails for 4-dimensional polytopes.*
- *There are nonrational 4-polytopes: combinatorial types that cannot be realized with rational vertex coordinates.*
- *The coordinates needed to represent all combinatorial types of rational 4-polytopes with integer vertices grow doubly exponentially with $f_0(P)$.*

The complete proof of this universality theorem is given in [Ric96]. One key component of the proof corresponds to another failure of a 3-dimensional phenomenon in dimension 4: for any facet (2-face) F of a 3-dimensional polytope P , the shape of F can be arbitrarily prescribed; in other words, the canonical map of realization spaces $\mathcal{R}(P) \rightarrow \mathcal{R}(F)$ is always surjective. Richter-Gebert shows that a similar statement fails in dimension 4, even if F is a 2-dimensional pentagonal face: see Figure 16.1.10 for the case of a hexagon.

A problem that is left open is the structure of the realization spaces of simplicial 4-polytopes. All that is available now is a universality theorem for simplicial polytopes without a dimension bound (see Section 6.3.4), and a single example of a simplicial 4-polytope that violates the isotopy property, by Bokowski, Ewald, and Kleinschmidt [BEK84].

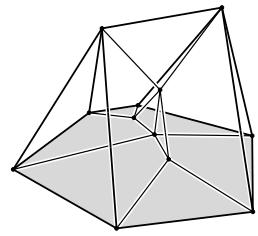


FIGURE 16.1.10

Schlegel diagram of a 4-dimensional polytope with 8 facets and 12 vertices, for which the shape of the base hexagon cannot be prescribed arbitrarily.

16.1.7 POLYTOPES WITH FEW VERTICES—GALE DIAGRAMS

GLOSSARY

Polytope with few vertices: A polytope that has only a few more vertices than its dimension; usually a d -polytope with at most $d+4$ vertices.

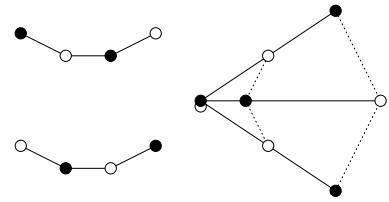
(Affine) Gale diagram: A configuration of n (positive and negative) points in affine space \mathbb{R}^{n-d-2} that encodes a d -polytope with n vertices uniquely up to projective transformations.

The computation of a Gale diagram involves only simple linear algebra. For this, let $V \in \mathbb{R}^{d \times n}$ be a matrix whose columns consist of coordinates for the vertices of a d -polytope. For simplicity, we assume that P is not a pyramid, and that the vertices $\{v^1, \dots, v^{d+1}\}$ affinely span \mathbb{R}^d . Let $\tilde{V} \in \mathbb{R}^{(d+1) \times n}$ be obtained from V by adding an extra (terminal) row of ones. The vector configuration given by the columns of \tilde{V} represents the *oriented matroid* of P ; see Chapter 6.

Now perform row operations on the matrix \tilde{V} to get it into the form $\tilde{V} \sim (I_{d+1}|A)$, where I_{d+1} denotes a unit matrix, and $A \in \mathbb{R}^{(d+1) \times (n-d-1)}$ is a real matrix. (The row operations do not change the oriented matroid.) The columns of the matrix $\tilde{V}^* := (-A^T|I_{n-d-1}) \in \mathbb{R}^{(n-d-1) \times n}$ then represent the dual oriented matroid. We find a vector $a \in \mathbb{R}^{n-d-1}$ that has nonzero scalar product with all the columns of \tilde{V}^* , divide each column w^* of \tilde{V}^* by the value $\langle a, w^* \rangle$, and delete from the resulting matrix any row that affinely depends on the others, thus obtaining a matrix $W \in \mathbb{R}^{(n-d-2) \times n}$. The columns of W give a colored point configuration in \mathbb{R}^{n-d-2} , where *black* points are used for the columns where $\langle a, w^* \rangle > 0$, and *white* points for the others. This colored point configuration represents an affine Gale diagram of P .

FIGURE 16.1.11

Two affine Gale diagrams of 4-dimensional polytopes: for a noncyclic neighborly polytope with 8 vertices, and for the polar (with 8 vertices) of the polytope with 8 facets from Figure 16.1.10, for which the shape of a hexagonal face cannot be prescribed arbitrarily.



It turns out that an affine configuration of colored points (consisting of n points that affinely span \mathbb{R}^e) represents a polytope (with n vertices, of dimension $n - e - 2$) if and only if the following criterion is met: For any hyperplane spanned by some of the points, and for each side of it, the number of black points on this side, plus the number of white points on the other side, is at least 2.

The final information one needs is how to read off properties of a polytope from its affine Gale diagram. Here the criterion is that a set of points represents a face if and only if the following condition is satisfied: the colored points *not* in the set support an affine dependency, with positive coefficients on the black points, and with negative coefficients on the white points. Equivalently, the convex hull of all the black points not in our set, and the convex hull of all the white points not in the set, intersect in their relative interiors.

Affine Gale diagrams have been *very* successfully used to study and classify polytopes with few vertices.

$d+1$ vertices: The only d -polytopes with $d+1$ vertices are the d -simplices.

$d+2$ vertices: There are exactly $\lfloor d^2/4 \rfloor$ combinatorial types of d -polytopes with $d+2$ vertices; among these, $\lfloor d/2 \rfloor$ types are simplicial. This corresponds to the situation of 0-dimensional affine Gale diagrams.

$d+3$ vertices: All d -polytopes with $d+3$ vertices are realizable with (small) integral coordinates and satisfy the isotopy property: all this can be easily analyzed in terms of 1-dimensional affine Gale diagrams.

$d+4$ vertices: Here anything can go wrong: the universality theorem for oriented matroids of rank 3 yields a universality theorem for simplicial d -polytopes with $d+4$ vertices. (See [Section 6.3.4](#).)

We refer to [Zie95, Lecture 6] for a detailed introduction to affine Gale diagrams.

16.2 METRIC PROPERTIES

The combinatorial data of a polytope—vertices, edges, . . . , facets—have their counterparts in genuine geometric data, such as face volumes, surface areas, quermass-integrals, and the like. In this second half of the chapter, we give a brief sketch of some key geometric concepts related to polytopes.

However, the topics of combinatorial and of geometric invariants are not disjoint at all: much of the beauty of the theory stems from the subtle interplay between the two sides. Thus, the computation of volumes inevitably leads to the construction of triangulations (explicitly or implicitly), mixed volumes lead to mixed subdivisions of Minkowski sums (one “hot topic” for current research in the area), quermass-integrals relate to face enumeration, and so on.

Furthermore, the study of polytopes yields a powerful approach to the theory of convex bodies: sometimes one can extend properties of polytopes to arbitrary convex bodies by approximation [Sch93]. However, there are also properties valid for polytopes that fail for convex bodies in general. This bug/feature is designed to keep the game interesting.

16.2.1 VOLUME AND SURFACE AREA

GLOSSARY

Volume of a d -simplex T : $V(T) = \left| \det \begin{pmatrix} v^0 & \cdots & v^d \\ 1 & \cdots & 1 \end{pmatrix} \right| / d!$, where $T = \text{conv}\{v^0, \dots, v^d\}$ with $v^0, \dots, v^d \in \mathbb{R}^d$.

Subdivision of a polytope P : A collection of polytopes $P_1, \dots, P_l \subseteq \mathbb{R}^d$ such that $P = \bigcup P_i$, and for $i \neq j$ we have that $P_i \cap P_j$ is a proper face of P_i and P_j (possibly empty). In this case we write $P = \uplus P_i$.

Triangulation of a polytope: A subdivision into simplices. (See [Chapter 17](#).)

Volume of a d -polytope: $\sum_{T \in \Delta(P)} V(T)$, where $\Delta(P)$ is a triangulation of P .

k -volume $V^k(P)$ of a k -polytope $P \subseteq \mathbb{R}^d$: The volume of P , computed with respect to the k -dimensional Euclidean measure induced on $\text{aff}(P)$.

Surface area of a d -polytope P : $\sum_{T \in \Delta(P), F \in \mathcal{F}_{d-1}(P)} V^{d-1}(T \cap F)$, where $\Delta(P)$ is a triangulation of P .

The volume $V(P)$ (i.e., the d -dimensional Lebesgue measure) and the surface area $F(P)$ of a d -polytope $P \subseteq \mathbb{R}^d$ can be derived from any triangulation of P , since volumes of simplices are easy to compute. The crux for this is in the (efficient?) generation of a triangulation, a topic on which Chapters 17 and [25](#) of this Handbook have more to say.

The following recursive approach only implicitly generates a triangulation, but derives explicit volume formulas. Let $P \subseteq \mathbb{R}^d$ ($P \neq \emptyset$) be a polytope. If $d = 0$ then we set $V(P) = 1$. Otherwise we set $\mathcal{S}_{d-1}(P) := \{u \in S^{d-1} \mid \dim(H(P, u) \cap P) = d-1\}$, and use this to define the volume of P as

$$V(P) := \frac{1}{d} \sum_{u \in \mathcal{S}_{d-1}(P)} h(P, u) \cdot V^{d-1}(H(P, u) \cap P).$$

Thus, for any d -polytope the volume is a sum of its facet volumes, each weighted by $1/d$ times its signed distance from the origin. Geometrically, this can be interpreted as follows. Assume for simplicity that the origin is in the interior of P . Then the collection $\{\text{conv}(F \cup \{0\}) \mid F \in \mathcal{F}_{d-1}(P)\}$ is a subdivision of P into d -dimensional pyramids, where the base of $\text{conv}(F \cup \{0\})$ has $(d-1)$ -dimensional volume $V^{d-1}(F)$ —to be computed recursively, the height of the pyramid is $h(P, u^F)$, and thus its volume is $\frac{1}{d}h(P, u^F) \cdot V^{d-1}(F)$; compare to Figure 16.2.1. The formula remains valid even if the origin is outside P or on its boundary.

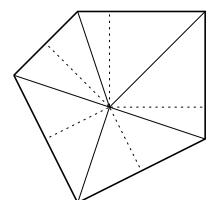


FIGURE 16.2.1

This pentagon, with the origin in its interior, is decomposed into five pyramids (triangles), each with one of the pentagon facets (edges) F_i as its base. For each pyramid, the height, of length $h(P, u^{F_i})$, is drawn as a dotted line.

Note that $V(P) \geq 0$. This holds with strict inequality if and only if the polytope P has full dimension d . The surface area $F(P)$ can also be expressed as

$$F(P) = \sum_{u \in \mathcal{S}_{d-1}(P)} V^{d-1}(H(P, u) \cap P).$$

Thus for a d -polytope the surface area is the sum of the $(d-1)$ -volumes of its facets. If $\dim(P) = d-1$, then $F(P)$ is twice the $(d-1)$ -volume of P . One has $F(P) = 0$ if and only if $\dim(P) < d-1$.

Both the volume and the surface area are continuous, monotone, and invariant with respect to rigid motions. $V(\cdot)$ is homogeneous of degree d , i.e., $V(\mu P) = \mu^d V(P)$ for $\mu \geq 0$, and $F(\cdot)$ is homogeneous of degree $d-1$. For further properties of the functionals $V(\cdot)$ and $F(\cdot)$ see [Had57] and [Sch93].

Table 16.2.1 gives the numbers of k -faces, the volume, and the surface area of the d -cube C_d (with edge length 2), of the cross-polytope C_d^Δ with edge length $\sqrt{2}$, and of the regular simplex T_d with edge length $\sqrt{2}$.

TABLE 16.2.1

POLYTOPE	$f_k(\cdot)$	VOLUME	SURFACE AREA
C_d	$2^{d-k} \binom{d}{k}$	2^d	$2d \cdot 2^{d-1}$
C_d^Δ	$2^{k+1} \binom{d}{k+1}$	$\frac{2^d}{d!}$	$2^d \frac{\sqrt{d}}{(d-1)!}$
T_d	$\binom{d+1}{k+1}$	$\frac{\sqrt{d+1}}{d!}$	$(d+1) \cdot \frac{\sqrt{d}}{(d-1)!}$

16.2.2 MIXED VOLUMES

GLOSSARY

Volume polynomial: The volume of the Minkowski sum $\lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_r P_r$, which is a homogeneous polynomial in $\lambda_1, \dots, \lambda_r$. (Here the P_i may be convex polytopes of any dimension, or more general (closed, bounded) convex sets.)

Mixed volumes: The coefficients of the volume polynomial of P_1, \dots, P_r .

Normal cone: The normal cone $N(F, P)$ of a face is the set of all vectors $v \in \mathbb{R}^d$ such that the supporting hyperplane $H(P, v)$ contains F , i.e.,

$$N(F, P) = \left\{ v \in \mathbb{R}^d \mid F \subseteq H(P, v) \cap P \right\}.$$

THEOREM 16.2.1 Mixed Volumes (cf. [Sch93, p. 270])

Let $P_1, \dots, P_r \subseteq \mathbb{R}^d$ be polytopes, $r \geq 1$, and $\lambda_1, \dots, \lambda_r \geq 0$. The volume of $\lambda_1 P_1 + \dots + \lambda_r P_r$ is a homogeneous polynomial in $\lambda_1, \dots, \lambda_r$ of degree d . Thus it can be written in the form

$$V(\lambda_1 P_1 + \dots + \lambda_r P_r) = \sum_{(i(1), \dots, i(d)) \in \{1, 2, \dots, r\}^d} \lambda_{i(1)} \cdots \lambda_{i(d)} \cdot V(P_{i(1)}, \dots, P_{i(d)}).$$

The coefficients in this expansion are symmetric in their indices. Furthermore, the coefficient $V(P_{i(1)}, \dots, P_{i(d)})$ depends only on $P_{i(1)}, \dots, P_{i(d)}$. It is called the **mixed volume** of the polytopes $P_{i(1)}, \dots, P_{i(d)}$.

With the abbreviation

$$V(P_1, k_1; \dots; P_r, k_r) := V(\underbrace{P_1, \dots, P_1}_{k_1 \text{ times}}, \dots, \underbrace{P_r, \dots, P_r}_{k_r \text{ times}}),$$

the polynomial becomes

$$V(\lambda_1 P_1 + \dots + \lambda_r P_r) = \sum_{\substack{k_1, \dots, k_r \geq 0 \\ k_1 + \dots + k_r = d}} \binom{d}{k_1, \dots, k_r} \lambda_1^{k_1} \cdots \lambda_r^{k_r} V(P_1, k_1; \dots; P_r, k_r).$$

In particular, the volume of the polytope P_i is given by the mixed volume $V(P_1, 0; \dots; P_i, d; \dots; P_r, 0)$. The theorem is also valid for arbitrary convex bodies: a good example where the general case can be derived from the polytope case by approximation. For more about the properties of mixed volumes from different points of view see Schneider [Sch93], Sangwine-Yager [San93], and McMullen [McM93].

The definition of the mixed volumes as coefficients of a polynomial is somewhat unsatisfactory. Schneider gave the following explicit rule, which generalizes an earlier result of Betke [Bet92] for the case $r = 2$. It uses information about the normal cones at certain faces. For this, note that $N(F, P)$ is a finitely generated cone, which can be written explicitly as the sum of the orthogonal complement of $\text{aff}(P)$ and the positive hull of those unit vectors u that are both parallel to $\text{aff}(P)$ and induce supporting hyperplanes $H(P, u)$ that contain a facet of P including F . Thus, for $P \subseteq \mathbb{R}^d$ the dimension of $N(F, P)$ is $d - \dim(F)$.

THEOREM 16.2.2 Schneider's Summation Formula [Sch94]

Let $P_1, \dots, P_r \subseteq \mathbb{R}^d$ be polytopes, $r \geq 2$. Let $x^1, \dots, x^r \in \mathbb{R}^d$ with $x^1 + \dots + x^r = 0$, $(x^1, \dots, x^r) \neq (0, \dots, 0)$, and

$$\bigcap_{i=1}^r (\text{relint } N(F_i, P_i) - x^i) = \emptyset$$

whenever F_i is a face of P_i and $\dim(F_1) + \dots + \dim(F_r) > d$. Then

$$\binom{d}{k_1, \dots, k_r} V(P_1, k_1; \dots; P_r, k_r) = \sum_{(F_1, \dots, F_r)} V(F_1 + \dots + F_r),$$

where the summation extends over the r -tuples (F_1, \dots, F_r) of k_i -faces F_i of P_i with $\dim(F_1 + \dots + F_r) = d$ and $\bigcap_{i=1}^r (N(F_i, P_i) - x^i) \neq \emptyset$.

The choice of the vectors x^1, \dots, x^r implies that the selected k_i -faces $F_i \subseteq P_i$ of a summand $F_1 + \dots + F_r$ are contained in complementary subspaces. Hence one may also write

$$\binom{d}{k_1, \dots, k_r} V(P_1, k_1; \dots; P_r, k_r) = \sum_{(F_1, \dots, F_r)} [F_1, \dots, F_r] \cdot V^{k_1}(F_1) \cdots V^{k_r}(F_r),$$

where $[F_1, \dots, F_r]$ denotes the volume of the parallelepiped that is the sum of unit cubes in the affine hulls of F_1, \dots, F_r .

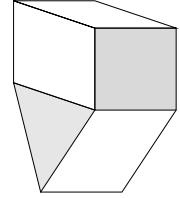
Finally, we remark that the selected sums of faces in the formula of the theorem form a subdivision of the polytope $P_1 + \dots + P_r$, i.e.,

$$P_1 + \dots + P_r = \biguplus_{(F_1, \dots, F_r)} (F_1 + \dots + F_r).$$

See Figure 16.2.2 for an example.

FIGURE 16.2.2

Here the Minkowski sum of a square P_1 and a triangle P_2 is decomposed into translates of P_1 and of P_2 (this corresponds to two summands with $F_1 = P_1$ resp. $F_2 = P_2$), together with three “mixed” faces that arise as sums $F_1 + F_2$, where F_1 and F_2 are faces of P_1 and P_2 (corresponding to summands with $\dim(F_1) = \dim(F_2) = 1$).



VOLUMES OF ZONOTOPES

If all summands in a Minkowski sum $Z = P_1 + \dots + P_r$ are line segments, say $P_i = p^i + [0, 1]z^i = \text{conv}\{p^i, p^i + z^i\}$ with $p^i, z^i \in \mathbb{R}^d$ for $1 \leq i \leq r$, then the resulting polytope Z is a zonotope. In this case the summation rule immediately gives $V(P_1, k_1; \dots; P_r, k_r) = 0$ if the vectors

$$\underbrace{z^1, \dots, z^1}_{k_1 \text{ times}}, \dots, \underbrace{z^r, \dots, z^r}_{k_r \text{ times}}$$

are linearly dependent. (This can also be seen directly from dimension considerations.) Otherwise, for $k_{i(1)} = k_{i(2)} = \dots = k_{i(d)} = 1$, say,

$$V(P_1, k_1; \dots; P_r, k_r) = \frac{1}{d!} \left| \det \left(z^{i(1)}, z^{i(2)}, \dots, z^{i(d)} \right) \right|.$$

Therefore, one obtains McMullen’s formula for the volume of the zonotope Z (cf. Shephard [Sh74]):

$$V(Z) = \sum_{1 \leq i(1) < i(2) < \dots < i(d) \leq r} \left| \det(z^{i(1)}, \dots, z^{i(d)}) \right|.$$

16.2.3 QUERMASSINTEGRALS AND INTRINSIC VOLUMES

GLOSSARY

i th quermassintegral $W_i(P)$: The mixed volume $V(P, d-i; B_d, i)$ of a polytope P and the d -dimensional unit ball B_d .

κ_d : The volume (Lebesgue measure) of B_d . (Hence $\kappa_0 = 1$, $\kappa_1 = 2$, $\kappa_2 = \pi$, etc.)

i th intrinsic volume $V_i(P)$: The $(d-i)$ th quermassintegral, scaled by the constant $\binom{d}{i}/\kappa_{d-i}$.

Outer parallel body of P at distance λ : The convex body $P + \lambda B_d$ for some $\lambda > 0$.

External angle $\gamma(F, P)$: The volume of $(\text{lin}(F - x^F) + N(F, P)) \cap B_d$ divided by κ_d , for $x^F \in \text{relint}(F)$. Thus $\gamma(F, P)$ is the “fraction of \mathbb{R}^d taken up by $\text{lin}(F - x^F) + N(F, P)$.” Equivalently, the external angle at a k -face F is the fraction of the spherical volume of S covered by $N(F, P) \cap S$, where S denotes the $(d-k-1)$ -dimensional unit sphere in $\text{lin}(N(F, P))$.

Internal angle $\beta(F, G)$ for faces $F \subseteq G$: The “fraction” of $\text{lin}\{G - x^F\}$ taken up by the cone $\text{pos}\{x - x^F \mid x \in G\}$, for $x^F \in \text{relint}(F)$. (A detailed discussion of relations between external and internal angles can be found in McMullen [McM75].)

The quermassintegrals are generalizations of both the volume and the surface area of P . In fact, they can also be seen as the continuous convex geometry analogs of face numbers.

For a polytope $P \subseteq \mathbb{R}^d$ and the d -dimensional unit ball B_d , the mixed volume formula, applied to the outer parallel body $P + \lambda B_d$, gives

$$V(P + \lambda B_d) = \sum_{i=0}^d \binom{d}{i} \lambda^i W_i(P),$$

with the convention $W_i(P) = V(P, d - i; B_d, i)$. This formula is known as the **Steiner polynomial**. The mixed volume $W_i(P)$, the i th quermassintegral of P , is an important quantity and of significant geometric interest [Had57, Sch93]. As special cases, $W_0(P) = V(P)$ is the volume, $dW_1(P) = F(P)$ is the surface area, and $W_d(P) = \kappa_d$.

For the geometric interpretation of $W_i(P)$ for polytopes, we use a normalization of the quermassintegrals due to McMullen [McM75]: For $0 \leq i \leq d$, the i th intrinsic volume of P is defined by

$$V_i(P) := \frac{\binom{d}{i}}{\kappa_{d-i}} W_{d-i}(P).$$

With this notation the Steiner polynomial can be written as

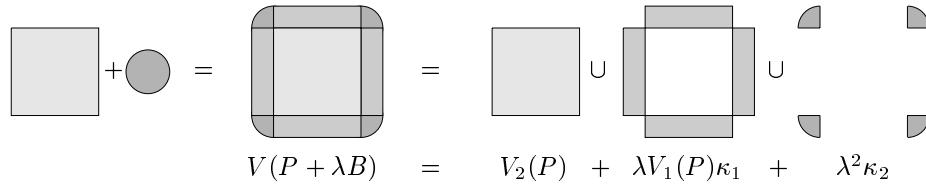
$$V(P + \lambda B_d) = \sum_{i=0}^d \lambda^{d-i} \kappa_{d-i} V_i(P).$$

(See Figure 16.2.3 for an example.) $V_d(P)$ is the volume of P , $V_{d-1}(P)$ is half the surface area, and $V_0(P) = 1$. One advantage of this normalization is that the intrinsic volumes are unchanged if P is embedded in some Euclidean space of different dimension. Thus, for $\dim(P) = k \leq d$, $V_k(P)$ is the ordinary k -volume of P with respect to the Euclidean structure induced in $\text{aff}(P)$.

For a $(\dim(P) - 2)$ -face F , the concept of external angle (see the glossary) reduces to the “usual” concept: then the external angle is given by $\frac{1}{2\pi} \arccos \langle u^{F_1}, u^{F_2} \rangle$ for unit normal vectors $u^{F_1}, u^{F_2} \in S^{d-1}$ to the facets F_1, F_2 with $F_1 \cap F_2 = F$. One

FIGURE 16.2.3

The Minkowski sum of a square P with a ball λB^2 yields the outer parallel body. This outer parallel body can be decomposed into pieces, whose volumes, $V(P)$, $\lambda V_1(P)\kappa_1$, and $\lambda^2\kappa_2$, correspond to the three terms in the Steiner polynomial.



has $\gamma(P, P) = 1$ for the polytope itself and $\gamma(F, P) = 1/2$ for each facet F . Using this concept, we get

$$V_k(P) = \sum_{F \in \mathcal{F}_k(P)} \gamma(F, P) \cdot V^k(F).$$

Internal and external angles are also useful tools in order to express combinatorial properties of polytopes (see the application below). One classical example is **Gram's equation** [Gra74]

$$\sum_{k=0}^{d-1} (-1)^k \sum_{F \in \mathcal{F}_k(P)} \beta(F, P) = (-1)^{d-1}.$$

This formula is quite similar to the Euler relation for the face numbers of a polytope (see [Chapter 18](#)). For a short and elegant probabilistic proof of Gram's equation, reducing it to Euler's relation, see [Wel94].

SOME COMPUTATIONS

In principle, one can use the external angle formula to determine the intrinsic volumes of a given polytope, but in general it is hard to calculate external angles. Indeed, for the computation of spherical volumes there are explicit formulas only in small dimensions.

In what follows, we give formulas for the intrinsic volumes of the polytopes C_d , C_d^Δ , and T_d . For this, we identify the k -faces of C_d with the k -cube C_k and the k -faces of C_d^Δ and of T_d with T_k , for $0 \leq k < d$.

The case of the cube C_d is rather trivial. Since $\gamma(C_k, C_d) = 2^{-(d-k)}$ one gets (see [Table 16.2.1](#))

$$V_k(C_d) = 2^k \binom{d}{k}.$$

For the regular simplex T_d we have

$$V_k(T_d) = \binom{d+1}{k+1} \cdot \frac{\sqrt{k+1}}{k!} \cdot \gamma(T_k, T_d).$$

An explicit formula for the external angles of a regular simplex by Ruben (see [Rub60] or [Had79]) is:

$$\gamma(T_k, T_d) = \sqrt{\frac{k+1}{\pi}} \int_{-\infty}^{\infty} e^{-(k+1)x^2} \left(\frac{1}{\sqrt{\pi}} \int_{-\infty}^x e^{-y^2} dy \right)^{d-k} dx.$$

For the regular cross-polytope we find for $k \leq d - 1$ that

$$V_k(C_d^\Delta) = 2^{k+1} \binom{d}{k+1} \cdot \frac{\sqrt{k+1}}{k!} \cdot \gamma(T_k, C_d^\Delta).$$

For this, the external angles of C_d^Δ were determined by Betke and Henk [BH93]:

$$\gamma(T_k, C_d^\Delta) = \sqrt{\frac{k+1}{\pi}} \int_0^\infty e^{-(k+1)x^2} \left(\frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy \right)^{d-k-1} dx.$$

AN APPLICATION

External angles and internal angles play a crucial role in work by Affentranger and Schneider [AS92] (see also [BV94]), who computed the expected number of k -faces of the orthogonal projection of a polytope $P \subseteq \mathbb{R}^d$ onto a randomly chosen isotropic subspace of dimension n . Let $E[f_k(P; n)]$ be that number. Then for $0 \leq k < n \leq d - 1$ it was shown that

$$E[f_k(P; n)] = 2 \sum_{m \geq 0} \sum_{F \in \mathcal{F}_k(P)} \sum_{\substack{G \in \mathcal{F}_{n-1-2m}(P) \\ F \subseteq G}} \beta(F, G) \gamma(G, P),$$

where $\beta(F, G)$ is the internal angle of the face F with respect to a face $G \supseteq F$.

In the sequel we apply the above formula to the polytopes C_d , C_d^Δ , and T_d . For the cubes one has $\beta(C_k, C_l) = (1/2)^{l-k}$, while the number of l -faces of C_d containing any given k -face is equal to $\binom{d-k}{l-k}$. Hence

$$E[f_k(C_d; n)] = 2 \binom{d}{k} \sum_{m \geq 0} \binom{d-k}{n-1-k-2m}.$$

In particular, $E[f_k(C_d; d-1)] = (2^{d-k} - 2) \binom{d}{k}$.

For the cross-polytope C_d^Δ the number of l -faces that contain a k -face is equal to $2^{l-k} \binom{d-k-1}{l-k}$. Thus

$$\begin{aligned} E[f_k(C_d^\Delta; n)] &= \\ &2 \binom{d}{k+1} \sum_{m \geq 0} 2^{n-2m} \binom{d-k-1}{n-1-k-2m} \beta(T_k, T_{n-1-2m}) \gamma(T_{n-1-2m}, C_d^\Delta). \end{aligned}$$

In the same way one obtains for T_d

$$\begin{aligned} E[f_k(T_d; n)] &= \\ &2 \binom{d+1}{k+1} \sum_{m \geq 0} \binom{d-k}{n-1-k-2m} \beta(T_k, T_{n-1-2m}) \gamma(T_{n-1-2m}, T_d). \end{aligned}$$

For the last two formulas one needs the internal angles $\beta(T_k, T_l)$ of the regular simplex T_d , for $0 \leq k \leq l \leq d$. For this, one has the following complex integral [BH99]:

$$\beta(T_k, T_l) = \frac{(k+1+l)^{1/2} (k+1)^{(l-1)/2}}{\pi^{(l+1)/2}} \int_{-\infty}^{\infty} e^{-w^2} \left(\int_0^{\infty} e^{-(k+1)y^2 + 2iw y} dy \right)^l dw.$$

Using this formula one can determine the asymptotic behavior of $E[f_k(C_d^\Delta; n)]$ and $E[f_k(T_d; n)]$ as n tends to infinity [BH99].

16.3 SOURCES AND RELATED MATERIAL

FURTHER READING

The classic account of the combinatorial theory of convex polytopes was given by Grünbaum in 1967 [Grü03]. It inspired and guided a great part of the subsequent research in the field. Besides the related chapters of this Handbook, we refer to [Zie95] and the handbook surveys by Klee and Kleinschmidt [KK95] and by Bayer and Lee [BL93] for further reading.

For the geometric theory of convex bodies we refer to the *Handbook of Convex Geometry* [GW93], to Schneider [Sch93] for an excellent monograph, and as an introduction to modern convex geometry we recommend [Bal97]. As for the algorithmic aspects of computing volumes, etc., we refer to Chapter 31 of this Handbook, on computational convexity, and to the additional references given there.

RELATED CHAPTERS

- Chapter 3: Tilings
- Chapter 6: Oriented matroids
- Chapter 7: Lattice points and lattice polytopes
- Chapter 12: Discrete aspects of stochastic geometry
- Chapter 17: Subdivisions and triangulations of polytopes
- Chapter 18: Face numbers of polytopes and complexes
- Chapter 19: Symmetry of polytopes and polyhedra
- Chapter 20: Polytope skeletons and paths
- Chapter 22: Convex hull computations
- Chapter 25: Triangulations and mesh generation
- Chapter 31: Computational convexity
- Chapter 62: Crystals and quasicrystals
- Chapter 64: Software

REFERENCES

- [AZ01] M. Aigner and G.M. Ziegler. *Proofs from THE BOOK*, 2nd Ed. Springer-Verlag, Berlin, 2001.
- [AS92] F. Affentranger and R. Schneider. Random projections of regular simplices. *Discrete Comput. Geom.*, 7:219–226, 1992.
- [Ba61] M.L. Balinski. On the graph structure of convex polyhedra in n -space. *Pacific J. Math.*, 11:431–434, 1961.
- [Bal97] K. Ball. An elementary introduction to modern convex geometry. In S. Levy, editor, *Flavors of Geometry*, MSRI Publications, volume 31, pages 1–58. Cambridge University Press, 1997.
- [BP01] I. Bárány and A. Pór. 0 – 1 polytopes with many facets. *Adv. Math.*, 161:209–228, 2001.

- [BV94] Y. Baryshnikov and R.A. Vitale. Regular simplices and Gaussian samples. *Discrete Comput. Geom.*, 11:141–147, 1994.
- [BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. North-Holland, Amsterdam, 1993.
- [Bet92] U. Betke. Mixed volumes of polytopes. *Arch. Math.*, 58:388–391, 1992.
- [BH93] U. Betke and M. Henk. Intrinsic volumes and lattice points of crosspolytopes. *Monatsh. Math.*, 115:27–33, 1993.
- [BLS⁺99] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*, 2nd Ed. Volume 46 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1999.
- [BM87] R. Blind and Peter Mani-Levitska. On puzzles and polytope isomorphisms. *Aequationes Math.*, 34:287–297, 1987.
- [BRS92] J. Bokowski, J. Richter-Gebert, and W. Schindler. On the distribution of order types. *Comput. Geom. Theory Appl.*, 1:127–142, 1992.
- [BEK84] J. Bokowski, G. Ewald, and P. Kleinschmidt. On combinatorial and affine automorphisms of polytopes. *Israel J. Math.*, 47:123–130, 1984.
- [BS89] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry*. Volume 1355 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1989.
- [BH99] K. Böröczky, Jr. and M. Henk. Random projections of regular polytopes. *Arch. Math.*, 73:465–473, 1999.
- [DL97] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Volume 15 of *Algorithms Combin.*, Springer-Verlag, Heidelberg, 1997.
- [GJ00] E. Gawrilow and M. Joswig. *polymake: a framework for analyzing convex polytopes*. In G. Kalai and G.M. Ziegler, editors, *Polytopes—Combinatorics and Computation*, 43–74. Birkhäuser, Basel, 2000. <http://www.math.tu-berlin.de/diskregeom/polymake>
- [Gra74] J. P. Gram. Om Rumvinklerne i et Polyeder. *Tidsskr. Math.*, 4:161–163, 1874.
- [GP85] M. Grötschel and M. Padberg. Polyhedral theory. In E.L. Lawler et al., editors, *The Traveling Salesman Problem*, pages 251–360. Wiley, Chichester, 1985.
- [GW93] P.M. Gruber and J.M. Wills, editors. *Handbook of Convex Geometry*, Volumes A and B. North-Holland, Amsterdam, 1993.
- [Grü03] B. Grünbaum. *Convex Polytopes*. Interscience, London, 1967; 2nd Ed., V. Kaibel, V. Klee, and G.M. Ziegler, editors, vol. 221 of *Graduate Texts in Math.*, Springer-Verlag, New York, 2003.
- [Had57] H. Hadwiger. *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie*. Springer-Verlag, Berlin, 1957.
- [Had79] H. Hadwiger. Gitterpunktanzahl im Simplex und Wills'sche Vermutung. *Math. Ann.*, 239:271–288, 1979.
- [KK93] J. Kahn and G. Kalai. A counterexample to Borsuk's conjecture. *Bull. Amer. Math. Soc.*, 29:60–62, 1993.
- [Kal88] G. Kalai. A simple way to tell a simple polytope from its graph. *J. Combin. Theory Ser. A*, 49:381–383, 1988.
- [Kal89] G. Kalai. The number of faces of centrally-symmetric polytopes (Research Problem). *Graphs Combin.*, 5:389–391, 1989.
- [KK95] V. Klee and P. Kleinschmidt. Polyhedral complexes and their relatives. In R. Graham,

- M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 875–917. North-Holland, Amsterdam, 1995.
- [Kor97] U. Kortenkamp. Every simplicial polytope with at most $d+4$ vertices is a quotient of a neighborly polytope. *Discrete Comput. Geom.*, 18:455–462, 1997.
- [McM75] P. McMullen. Non-linear angle-sum relations for polyhedral cones and polytopes. *Math. Proc. Comb. Phil. Soc.*, 78:247–261, 1975.
- [McM93] P. McMullen. Valuations and dissections. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume B, pages 933–988. North-Holland, Amsterdam, 1993.
- [McM94] P. McMullen. Duality, sections and projections of certain Euclidean tilings. *Geom. Dedicata*, 49:183–202, 1994.
- [OS94] S. Onn and B. Sturmfels. A quantitative Steinitz' theorem. *Beiträge Algebra Geom./Contrib. Algebra Geom.*, 35:125–129, 1994.
- [Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes*. Volume 1643 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1996.
- [RZ94] J. Richter-Gebert and G.M. Ziegler. Zonotopal tilings and the Bohne-Dress theorem. In H. Barcelo and G. Kalai, editors, *Jerusalem Combinatorics '93*, pages 211–232. Volume 178 of *Contemp. Math.*, Amer. Math. Soc., Providence, 1994.
- [Rub60] H. Ruben. On the geometrical moments of skew-regular simplices in hyperspherical space; with some applications in geometry and mathematical statistics. *Acta. Math.*, 103:1–23, 1960.
- [Ryb99] K. Rybníkov. Stresses and liftings of cell-complexes. *Discrete Comput. Geom.*, 21:481–517, 1999.
- [San93] J.R. Sangwine-Yager. Mixed volumes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 43–71. North-Holland, Amsterdam, 1993.
- [Sch93] R. Schneider. *Convex Bodies: The Brunn-Minkowski Theory*. Volume 44 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993.
- [Sch94] R. Schneider. Polytopes and the Brunn-Minkowski theory. In T. Bisztriczky, P. McMullen, R. Schneider, and A. Ivić Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 273–299. Kluwer, Dordrecht, 1994.
- [She82] I. Shemer. Neighborly polytopes. *Israel J. Math.*, 43:291–314, 1982.
- [Sh74] G.C. Shephard. Combinatorial properties of associated zonotopes. *Canad. J. Math.*, 26:302–321, 1974.
- [SR34] E. Steinitz and H. Rademacher. *Vorlesungen über die Theorie der Polyeder*. Springer-Verlag, Berlin, 1934; reprint, Springer-Verlag, Berlin, 1976.
- [VS92] A.M. Vershik and P.V. Sporyshev. Asymptotic behavior of the number of faces of random polyhedra and the neighborliness problem. *Selecta Math. Soviet.*, 11:181–201, 1992.
- [Wel94] E. Welzl. Gram's equation—a probabilistic proof. In *Results and Trends in Theoretical Computer Science (Graz, 1994)*, volume 812 of *Lecture Notes in Comput. Sci.*, pages 422–424. Springer-Verlag, Berlin, 1994.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995; revised ed., 1998.
[Updates, corrections, etc. available at <http://www.math.tu-berlin.de/~ziegler/>.]
- [Zie00] G.M. Ziegler. Lectures on 0/1-polytopes. In G. Kalai and G.M. Ziegler, editors, *Polytopes—Combinatorics and Computation*, volume 29 of *DMV Seminars*, pages 1–41. Birkhäuser, Basel, 2000.

17 SUBDIVISIONS AND TRIANGULATIONS OF POLYTOPES

Carl W. Lee

INTRODUCTION

Starting from a given finite set of points V in \mathbb{R}^d , we consider subdivisions of the convex hull of V into polytopes $\{P_1, \dots, P_m\}$. A subdivision is a triangulation if each P_i is a simplex. We start with definitions and properties, then turn to methods of constructing subdivisions and triangulations, face-counting results, some particular triangulations, and secondary and fiber polytopes. We confine ourselves to triangulations of convex structures for the most part.

17.1 BASIC CONCEPTS

GLOSSARY

Affine span: The affine span of a set V is the smallest affine space, or flat, containing V . It is denoted by $\text{aff}(V)$.

Convex hull: The convex hull of a set V is the smallest convex set containing V . It is denoted by $\text{conv}(V)$.

Polytope: A polytope P is the convex hull of a finite set of points. If it is d -dimensional, its boundary consists of faces of dimension -1 (the empty set), 0 (vertices), 1 (edges), $2, \dots$, and $d - 1$ (facets). Its set of vertices will be denoted by $\text{vert}(P)$.

Face of a set: Suppose S is a subset of a finite set T of points. We say S is a face of the set T if there is face F of the polytope $P = \text{conv}(T)$ for which $S = T \cap F$. Note that S may include points that are not vertices of F . If F is not P itself we say S is a face of the boundary of T . The dimension of S is taken to be the dimension of $\text{conv}(S)$, and faces of dimension $0, 1$, and $\dim(T) - 1$ are referred to as vertices, edges, and facets, respectively, of the set T .

Subdivision: Suppose V is a finite set of points such that $P = \text{conv}(V)$ is d -dimensional (a d -polytope). A subdivision of V is a finite collection $S = \{S_1, \dots, S_m\}$ of subsets of V such that:

- For each i , $1 \leq i \leq m$, $P_i = \text{conv}(S_i)$ is a d -polytope;
- P is the union of P_1, \dots, P_m ; and
- If $i \neq j$ then there is a common (possibly empty) face F of the boundaries of S_i and S_j such that $P_i \cap P_j = \text{conv}(F)$.

In this case we will also say that S is a subdivision of the polytope P . It is also usual to refer to the collection of polytopes $\{P_1, \dots, P_m\}$ rather than the subsets

$\{S_1, \dots, S_m\}$ as the subdivision, keeping in mind, though, that it may not be the case that $S_i = \text{vert}(P_i)$.

Trivial subdivision: The trivial subdivision of V is the subdivision $\{V\}$.

Simplex: A d -dimensional simplex is a d -polytope with exactly $d + 1$ vertices.

We will also refer to the set of vertices of a d -simplex as a d -simplex.

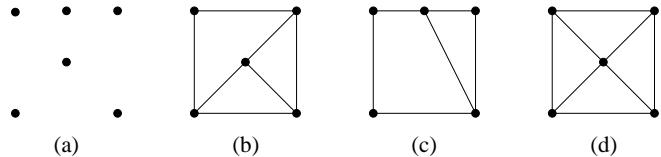
Triangulation: Suppose $\text{conv}(V)$ has dimension d . A subdivision $\{S_1, \dots, S_m\}$ of V is a triangulation if each S_i has cardinality $d + 1$, so that each P_i is a simplex.

Faces: The faces of a subdivision $\{S_1, \dots, S_m\}$ consist of S_1, \dots, S_m together with their faces. It is also usual to say that the faces of the subdivision are the polytopes P_1, \dots, P_m and their faces.

EXAMPLES

In Figure 17.1.1, (a) shows a set of points. The collection of three polygons in (b) is not a subdivision of that set since not every pair of polygons meets along a common edge or vertex; (c) shows a subdivision that is not a triangulation; and (d) gives a triangulation.

FIGURE 17.1.1
 (a) A set of points.
 (b) A nonsubdivision.
 (c) A subdivision.
 (d) A triangulation.



17.2 SEQUENTIAL CONSTRUCTION PROCEDURES

The convex hull of a finite set of points $V = \{v_1, \dots, v_n\}$ can be constructed sequentially by successively constructing $R_1 = \text{conv}(\{v_1\})$, $R_2 = \text{conv}(R_1 \cup \{v_2\})$, $R_3 = \text{conv}(R_2 \cup \{v_3\}), \dots, R_n = \text{conv}(R_{n-1} \cup \{v_n\})$. This inductive method for constructing the convex hull of V appears, for example, in [Grü67, Section 5.2]. See Chapter 22 of this Handbook for a discussion of its implementation. With little additional effort, a triangulation of each R_i can also be constructed, resulting finally in a triangulation of V . Another method of constructing a triangulation of V is to begin with the trivial subdivision of V , and then obtain a sequence of refinements. See [Lee91a].

GLOSSARY

Refinement of a subdivision: Suppose $S = \{S_1, \dots, S_l\}$ and $T = \{T_1, \dots, T_m\}$ are two subdivisions of V . Then T is a refinement of S if for each j , $1 \leq j \leq m$, there exists i , $1 \leq i \leq l$, such that $T_j \subseteq S_i$. In this case we will write $T \leq S$.

Visible facet: Suppose $P = \text{conv}(V)$ is a d -polytope in \mathbb{R}^d , F is a facet of P , and v is a point in \mathbb{R}^d . There is a unique hyperplane H (affine set of dimension

$d-1$) containing F . The polytope P is contained in exactly one of the two closed halfspaces determined by H . If v is contained in the opposite open halfspace, then F is said to be visible from v . We also say that the facet $V \cap F$ of the set V is visible from v . If P is a k -polytope in \mathbb{R}^d with $k < d$ and $v \in \text{aff}(P)$, then the above definition is modified in the obvious way so that everything is considered relative to the ambient space $\text{aff}(P)$.

Placing a vertex: Suppose $S = \{S_1, \dots, S_m\}$ is a subdivision of V and $v \notin V$. The subdivision T of $V \cup \{v\}$ that results from placing v is obtained as follows:

- If $v \notin \text{aff}(V)$, then for each $S_i \in S$, include $S_i \cup \{v\}$ in T .
- If $v \in \text{aff}(V)$, then for each $S_i \in S$, include S_i in T ; and if F is a facet of S_i that is contained in a facet of $\text{conv}(V)$ visible from v , then $F \cup \{v\} \in T$.
- Note: if $v \in \text{conv}(V)$, then $S = T$.

Pulling a vertex: Suppose $S = \{S_1, \dots, S_m\}$ is a subdivision of V and $v \in S_1 \cup \dots \cup S_m$. The result of pulling v is the subdivision T of V obtained by modifying each $S_i \in S$ as follows:

- If $v \notin S_i$, then $S_i \in T$.
- If $v \in S_i$, then +for every facet F of S_i not containing v , $F \cup \{v\} \in T$.

Note that T is a refinement of S . Pulling is described in Hudson [Hud69, Lemma 1.4].

Pushing a vertex: Suppose $S = \{S_1, \dots, S_m\}$ is a subdivision of V (where $\dim(\text{conv}(V)) = d$) and $v \in S_1 \cup \dots \cup S_m$. The result of pushing v is the subdivision T of V obtained by modifying each $S_i \in S$ as follows:

- If $v \notin S_i$, then $S_i \in T$.
- If $v \in S_i$ and $S_i \setminus \{v\}$ is $(d-1)$ -dimensional (i.e., $\text{conv}(S_i)$ is a pyramid with apex v), then $S_i \in T$.
- If $v \in S_i$ and $S_i \setminus \{v\}$ is d -dimensional, then $S_i \setminus \{v\} \in T$. Also, if F is any facet of $S_i \setminus \{v\}$ that is visible from v , then $F \cup \{v\} \in T$.

Note that T is a refinement of S .

Lexicographic subdivisions: If T is any subdivision of V constructed by starting with the trivial subdivision of V and then pushing and/or pulling some/all of the points in V in some order, then T is a lexicographic subdivision. Such triangulations were introduced by Sturmfels [Stu91]; see also [Stu96].

Diameter of a subdivision: Suppose $\{S_1, \dots, S_m\}$ is a subdivision, and $P_i = \text{conv}(S_i)$, $1 \leq i \leq m$. Polytopes $P_i \neq P_j$ are **adjacent** if they share a common facet. A sequence P_{i_0}, \dots, P_{i_k} is a **path** if P_{i_j} and $P_{i_{j+1}}$ are adjacent for each j , $1 \leq j \leq k$. The **length** of such a path is k . The **distance** between polytopes P_i and P_j is the length of the shortest path connecting them. The diameter of the subdivision is the maximum distance occurring between pairs of polytopes P_i, P_j .

MAIN RESULTS

Results (1) through (6) below are all discussed in [Lee91a].

1. If the points of V are ordered $\{v_1, \dots, v_n\}$ and T is the subdivision obtained by placing the points of V in that order, then
 - (a) T is a triangulation of V .
 - (b) The same triangulation is obtained by starting with the trivial subdivision of V and pushing the points of V in the opposite order v_n, \dots, v_1 .
 - (c) The diameter of T does not exceed $2(n-d-1)$, where $d = \dim(\text{conv}(V))$ [Lee91a].

Billera and Munson [BM84] defined triangulations by placing in the more general context of oriented matroids.

2. If S is any subdivision of $V = \{v_1, \dots, v_n\}$, then S can be refined to a triangulation by sequentially pushing and/or pulling all the points in some order.
3. For any specified point $v_k \in V = \{v_1, \dots, v_n\}$, there is a triangulation of V in which every simplex of maximum dimension contains v_k as a vertex—begin with the trivial subdivision $S = \{V\}$, pull v_k first, then pull the remaining points in any order.
4. For any specified simplex F with vertices in $V = \{v_1, \dots, v_n\}$, there is a triangulation of V in which F is a face—first place the vertices of F , then place the remaining vertices in any order.
5. If $\dim(\text{conv}(V)) = d$ and $\text{card}(V) \leq d+3$, then every triangulation of V can be obtained by placing the points of V in some order (equivalently, pushing the points of V in the opposite order) [Lee91a].
6. If V is the set of vertices of a convex n -gon in \mathbb{R}^2 , then every triangulation of V can be obtained by placing the points of V in some order (equivalently, pushing the points of V in the opposite order).
7. Suppose $V = \{v_1, \dots, v_n\}$ is the set of vertices of some d -polytope P . For a face F of P , define $v(F) = v_k$, where $k = \min\{i \mid v_i \in F\}$. A **full flag** of P is a chain C of faces $F_0 \subset F_1 \subset F_2 \subset \dots \subset F_{d-1} \subset F_d = P$ such that $\dim(F_i) = i$, $0 \leq i \leq d$, and $v(F_i) \neq v(F_{i-1})$, $1 \leq i \leq d$. For a full flag C , write $v(C) = \{v(F_0), \dots, v(F_d)\}$. Then the simplices of the triangulation of P determined by pulling the vertices of P in the order v_1, \dots, v_n are $\{v(C) \mid C \text{ is a full flag of } P\}$ [Sta80, Lee85].

EXAMPLES

[Figure 17.2.1](#) gives three triangulations of a set of seven points that can be obtained from the trivial subdivision by pulling and pushing [Lee91a]. The triangulation in (a) is obtained by pulling point 1, but cannot be obtained by pushing alone. The triangulation in (b) is obtained by pushing the points in the indicated order, or placing them in the opposite order, but cannot be obtained by pulling points alone. The lexicographic triangulation in (c) is obtained by pushing point 1 and then pulling point 2, but cannot be obtained by pulling points alone or by pushing points alone.

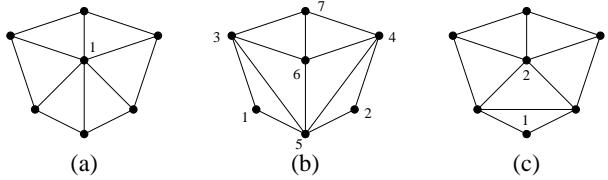


FIGURE 17.2.1

- (a) A pulling triangulation.
- (b) A pushing triangulation.
- (c) A lexicographic triangulation.

17.3 REGULAR TRIANGULATIONS AND SUBDIVISIONS

GLOSSARY

Regular subdivision: Any convex hull algorithm for points in \mathbb{R}^{d+1} (see Chapter 22 of this Handbook) can be used to compute subdivisions of sets of points $V = \{v_1, \dots, v_n\}$ in \mathbb{R}^d .

Such subdivisions are called regular and are obtained in the following way:

- (i) Regard V as sitting naturally in $(\mathbb{R}^d, 0)$.
- (ii) Choose arbitrary real numbers $\alpha_1, \dots, \alpha_n$.
- (iii) Determine $Q = \text{conv}(\{(v_1, \alpha_1), \dots, (v_n, \alpha_n)\})$.
- (iv) Project the lower facets of Q onto $(\mathbb{R}^d, 0)$.

Here, a lower facet is a facet of Q that is visible from the point $(0, -\alpha)$ for α sufficiently large. See [GKZ94, Lee91a, Zie95]. Some algorithmic aspects of computing regular triangulations can be found in [ES96].

Weakly regular subdivision: A subdivision S of a set V is weakly regular if there exists a set V' of the same dimension having a regular subdivision S' such that (V', S') is combinatorially isomorphic to (V, S) . That is, there is a one-to-one correspondence between the points of V and the points of V' such that for every subset $F \subseteq V$ and corresponding subset $F' \subseteq V'$, F is a face of S if and only if F' is a face of S' of the same dimension.

Polytopal complex: A polytopal complex is a finite, nonempty collection S of polytopes in \mathbb{R}^d that contains all the faces of its polytopes, and such that the intersection of any two of its polytopes is a common face of each of them (possibly empty). The dimension of S , $\dim(S)$, is the largest dimension of a polytope in S , and S is **pure** if every polytope in S is contained in one of dimension $\dim(S)$ [Zie95]. (Thus every subdivision $\{S_1, \dots, S_m\}$ has an associated pure polytopal complex consisting of the polytopes $\text{conv}(S_1), \dots, \text{conv}(S_m)$ and their faces.)

Shellable: A pure polytopal complex S is shellable if it is 0-dimensional (i.e., a nonempty finite set of points) or else $\dim(S) = k > 0$ and S has a **shelling**, i.e., an ordering of its maximal faces P_1, \dots, P_m such that for $2 \leq j \leq m$ the intersection of P_j with $P_1 \cup \dots \cup P_{j-1}$ is nonempty and is the beginning segment of a shelling of the $(k-1)$ -dimensional boundary complex of P_j [Zie95].

MAIN RESULTS

1. All regular subdivisions are shellable. This is a consequence of the fact that every polytope admits a line shelling [BM71, Zie95], and so in particular one

can begin shelling the polytope Q above by choosing a point in the interior of Q in sufficiently general position, moving in the direction $(O, -1)$, and listing the lower facets of Q in the order in which their supporting hyperplanes are crossed. On the other hand, there exist nonshellable subdivisions, starting in dimension 3. The first example was Rudin's nonshellable triangulation of a tetrahedron [Rud58]. For some additional discussion, including a nonshellable triangulation of the 3-cube, see Ziegler [Zie95].

2. All lexicographic triangulations are regular. In particular, if v_1, \dots, v_n are pushed/pulled in that order, then the corresponding triangulation is obtained by choosing $|\alpha_1| \gg |\alpha_2| \gg \dots \gg |\alpha_n| \gg 0$, where $\alpha_i > 0$ if v_i is pushed and $\alpha_i < 0$ if v_i is pulled [Lee91a].
3. If $\text{card}(V) = \dim(\text{conv}(V)) + 2$, then there are exactly two triangulations of V , and both are regular [Lee91a].
4. If $\text{card}(V) = \dim(\text{conv}(V)) + 3$, then all subdivisions of V are regular [Lee91a].
5. If V is the set of vertices of a convex n -gon in \mathbb{R}^2 , then all subdivisions of V are regular.
6. If $V \subset \mathbb{R}^2$, then all subdivisions of V are weakly regular as a consequence of Steinitz's Theorem (see [Grü67, Zie95] and Chapter 16 of this Handbook). However, there exists a set V of 6 points in \mathbb{R}^2 having a nonregular triangulation [Lee91a] (see Figure 17.3.2(b)).
7. There exists a set V of 7 points that are the vertices of a 3-polytope having a nonregular triangulation that is not even weakly regular [Lee91a] (see Figure 17.3.3(b)).
8. If V is the vertex set of $C_{4n-4}(4n)$, the cyclic polytope of dimension $4n - 4$ with $4n$ vertices, then V has at least 2^n triangulations, of which only $O(n^4)$ are regular [dHSS96]. (See Chapter 16 of this Handbook for the definition of the cyclic polytope.) If V is the vertex set of $C_{n-4}(n)$, then V has exactly $(n+4)2^{(n-4)/2} - n$ triangulations if n is even and $((3n+11)/2)2^{(n-5)/2} - n$ triangulations if n is odd. Of these, at most $6\binom{m}{4} + 3\binom{m}{3} + 4\binom{m}{2} - m + 2$ are regular if $n = 2m$ is even, and $6\binom{m}{4} + 5\binom{m}{2} - 4m + 5$ are regular if $n = 2m - 1$ is odd, and this number is tight for sufficiently generic coordinatizations of the polytope [AS02].
9. If $\alpha_i = \|v_i\|^2$, then the resulting subdivision is the Delaunay subdivision. If $\alpha_i = -\|v_i\|^2$, then the resulting subdivision is the “farthest site” Delaunay subdivision. (See Chapters 23 and 25 of this Handbook.)
10. Given a subdivision of V , one can test its regularity by using linear programming to check the existence of appropriate α_i , $1 \leq i \leq n$. On the other hand, checking weak regularity is quite hard, as difficult as determining solutions to systems of real polynomial inequalities (see comments on the Universality Theorem in Chapters 6 and 16 of this Handbook, and in [Zie95]).

EXAMPLES

Figure 17.3.1 shows the two triangulations (both regular) of the vertices of a 3-dimensional bipyramid over a triangle. In (a) there are two tetrahedra in the triangulation, sharing a common internal triangle; in (b) there are three, sharing a common internal edge.

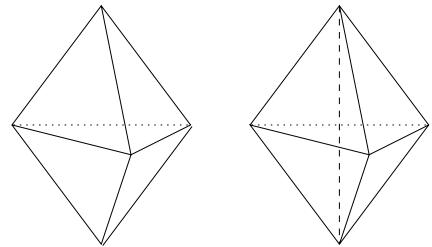


FIGURE 17.3.1

The two triangulations of a set of 5 points in \mathbb{R}^3 .

(a)

(b)

Figure 17.3.2 shows triangulations of two different sets of 6 points in \mathbb{R}^2 . The first triangulation is regular, the second is not. But by virtue of the first triangulation, the second is weakly regular.

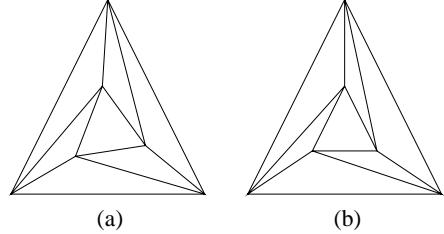


FIGURE 17.3.2

A regular and a nonregular (but weakly regular) triangulation.

Figure 17.3.3 shows two 3-polytopes, each with 7 vertices. The polytope in (a) is a “capped triangular prism” and its vertex set admits two nonregular triangulations. Denoting the simplices by their vertex sets, these are: $\{1257, 1457, 1236, 1267, 1345, 1346, 1467\}$ and $\{1245, 1247, 1237, 1367, 1356, 1456, 1467\}$. Both triangulations are, however, weakly regular. The polytope in (b) is obtained from the capped triangular prism by rotating the top triangle by a small amount. Its vertex set has one nonregular triangulation, which is not even weakly regular: $\{1245, 1247, 1237, 1367, 1356, 1456, 1467, 2457, 2367, 2345\}$. See [Lee91a].

17.3.1 TRIANGULATING REGIONS BETWEEN POLYTOPES

Suppose P and Q are two d -polytopes in \mathbb{R}^d with disjoint vertex sets V and W , respectively, and Q is contained in P . One can triangulate the region inside of P and outside of Q by the following procedure [GP88]:

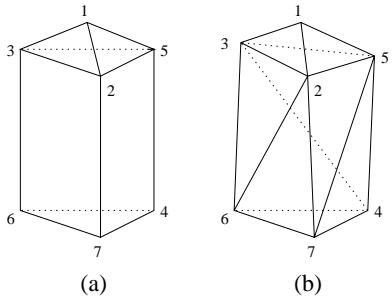


FIGURE 17.3.3

Two polytopes with nonregular triangulations.

(a)

(b)

1. Construct the regular subdivision of $V \cup W$ by setting $\alpha_i = 1$ for $v_i \in V$ and $\alpha_i = 0$ for $v_i \in W$.
2. Refine this subdivision to a triangulation by pushing and/or pulling each point in $V \cup W$.
3. Ignore the portion of the triangulation within Q .

Now suppose P and Q are two d -polytopes in \mathbb{R}^d with disjoint vertex sets V and W , respectively, and that there is a hyperplane H for which P and Q are contained in opposite open halfspaces. One can triangulate the region in $\text{conv}(P \cup Q)$ that is exterior to P and Q by the following procedure [GP88]:

1. Construct the regular subdivision of $V \cup W$ by setting α equal to the distance of v_i to H for each $v_i \in V \cup W$. For example, if $H = \{x \mid a \cdot x = \beta\}$, then α_i can be taken to equal $|a \cdot v_i - \beta|$. (It would also suffice to use these values of α_i for $v_i \in V$ and to set $\alpha_i = 0$ for $v_i \in W$.)
2. Refine this subdivision to a triangulation by pushing and/or pulling each point in $V \cup W$.
3. Ignore the portion of the triangulation within P or Q .

17.4 SUBDIVISIONS, TRIANGULATIONS, AND FACE VECTORS

Suppose $S = \{S_1, \dots, S_m\}$ is a subdivision of V such that $\dim(\text{conv}(V)) = d$. In this section we examine some of the properties of its face numbers. See [Bay93].

GLOSSARY

Boundary: Suppose S is as above. The boundary complex ∂S of S is the set of those faces of S given by $\{F \mid F \text{ is a face of } S, \text{ and } F \subseteq G \text{ for some face } G \text{ of dimension } d-1 \text{ contained in exactly one } S_i\}$. In particular, the empty set is a member of ∂S .

Interior: Suppose S is as above. The interior $\text{int } S$ is the set of those faces of S that are not in the boundary.

***f*-vector:** Suppose S is as above. Let $f_j(S)$ denote the number of j -dimensional faces of S , $-1 \leq j \leq d$. Note that $f_{-1}(S) = 1$ since the empty set is the unique face of S of dimension -1 . The f -vector of S is $f(S) = (f_0(S), \dots, f_d(S))$. In an analogous way we define $f(\partial S)$ and $f(\text{int } S)$. Note that $f_{-1}(\partial S) = 1$ and $f_{-1}(\text{int } S) = 0$.

Simplicial polytope: A simplicial polytope is one for which every facet (and hence every face) is a simplex.

17.4.1 h -VECTORS and g -VECTORS

Suppose S is any polytopal complex of dimension d . For example, S might be the boundary complex of a polytope of dimension $d+1$ or the polytopes corresponding to a subdivision of a finite set V such that $\text{conv}(V)$ has dimension d .

We define the h -vector $h(S) = (h_0(S), \dots, h_{d+1}(S))$ with generating function $h(S, x) = \sum_{i=0}^{d+1} h_i x^{d+1-i}$, and the g -vector $g(S) = (g_0(S), \dots, g_{\lfloor(d+1)/2\rfloor}(S))$ with generating function $g(S, x) = \sum_{i=0}^{\lfloor(d+1)/2\rfloor} g_i x^i$ in the following recursive way:

1. $g_0(S) = h_0(S)$.
2. $g_i(S) = h_i(S) - h_{i-1}(S)$, $1 \leq i \leq \lfloor(d+1)/2\rfloor$.
3. $g(\emptyset, x) = h(\emptyset, x) = 1$. (Here \emptyset denotes the empty polytopal complex, as distinguished from $\{\emptyset\}$, the polytopal complex consisting of a single set.)
4. $h(S, x) = \sum_{G \text{ face of } S} g(\partial G, x)(x-1)^{d-\dim(G)}$.

Take $h_i(S) = 0$ if $i < 0$ or $i > d+1$, and $g_i(S) = 0$ if $i < 0$ or $i > \lfloor(d+1)/2\rfloor$. For more information on f -vectors, g -vectors, and h -vectors, refer to [Chapter 18](#) of this Handbook. The formulas are simpler when all of the faces of S are simplices.

MAIN RESULTS

1. Assume that T is a triangulation of a d -polytope.
 - (a) The number of d -simplices in T equals the sum of the components of the h -vector of T .
 - (b) The h -vector of T is nonnegative [Sta96].
 - (c) The h -vector of ∂T is symmetric; i.e., $h_i(\partial T) = h_{d-i}(\partial T)$, $0 \leq i \leq d$. These are the Dehn-Sommerville equations; see [MS71, Sta96, Zie95] and Chapter 18 of this Handbook.
 - (d) The h -vectors of T , ∂T , and $\text{int } T$ are related in the following ways [MW71]:

$$h_i(T) - h_{d+1-i}(T) = h_i(\partial T) - h_{i-1}(\partial T), \quad 0 \leq i \leq d+1.$$

$$h_i(T) = h_{d+1-i}(\text{int } T), \quad 0 \leq i \leq d+1.$$

In particular, the h -vectors and the f -vectors of ∂T and $\text{int } T$ are completely determined by the h -vector (and hence the f -vector) of T .

- (e) Assume further that T is shellable and that P_1, \dots, P_m is a shelling order of the d -dimensional simplices in T . In particular, each P_j meets $\bigcup_{i=1}^{j-1} P_i$ in some positive number s_j of facets of P_j , $2 \leq j \leq m$. Define also $s_1 = 0$. Then $h_i(T)$ equals $\text{card}\{j \mid s_j = i\}$, $0 \leq i \leq d+1$ [McM70, MS71, Sta96].
- (f) Assume further that T is regular. Then for every integer k , $0 \leq k \leq d+2$, $(h_0(T) - h_{d+k+1}(T), h_1(T) - h_{d+k}(T), h_2(T) - h_{d+k-1}(T), \dots, h_{\lfloor(d+k+1)/2\rfloor}(T) - h_{\lfloor(d+k+2)/2\rfloor}(T))$ is an M -sequence [BL81]. (See [Chapter 18](#) of this Handbook for the definition of M -sequence.)
2. If S is the trivial subdivision of a convex d -polytope P consisting of P itself, then
- $$h_i(S) = \begin{cases} h_i(\partial P) - h_{i-1}(\partial P), & 1 \leq i \leq \lfloor d/2 \rfloor, \\ 0, & \lfloor d/2 \rfloor < i \leq d. \end{cases}$$
- See [Bay93].
3. Suppose V is a finite set of points with rational coordinates, S is a subdivision of V , and $P = \text{conv}(V)$. Then for all i , $h_i(S) \geq h_i(P)$ and $h_i(\partial S) \geq h_i(\partial P)$. Further, if P is simplicial and S is a triangulation, the result holds even without the rationality assumption. In either case, $f_d(S) \geq h_{\lfloor d/2 \rfloor}(\partial S) \geq h_{\lfloor d/2 \rfloor}(\partial P)$ [Bay93, Sta92].

EXAMPLES

In Table 17.4.1, we give the h -vectors and g -vectors of some polytopal complexes.

TABLE 17.4.1 h - and g -vectors of polytopal complexes.

S	h -vector	g -vector
$\{\emptyset\}$	(1)	(1)
Set of n points	$(1, n-1)$	(1)
Line segment	$(1, 0, 0)$	$(1, -1)$
Boundary of convex n -gon	$(1, n-2, 1)$	$(1, n-3)$
Trivial subdivision of convex n -gon	$(1, n-3, 0, 0)$	$(1, n-4)$
Boundary of tetrahedron	$(1, 1, 1, 1)$	$(1, 0)$
Trivial subdivision of tetrahedron	$(1, 0, 0, 0, 0)$	$(1, -1, 0)$
Boundary of cube	$(1, 5, 5, 1)$	$(1, 4)$
Trivial subdivision of cube	$(1, 4, 0, 0, 0)$	$(1, 3, -4)$
Triangulation of cube into 6 tetrahedra (See Figure 17.5.3(a))	$(1, 4, 1, 0, 0)$	$(1, 3, -3)$
Boundary of triangular prism	$(1, 3, 3, 1)$	$(1, 2)$
Trivial subdivision of triangular prism	$(1, 2, 0, 0, 0)$	$(1, 1, -2)$
Triangulation of triangular prism into 3 tetrahedra (See Figure 17.5.1)	$(1, 2, 0, 0, 0)$	$(1, 1, -2)$

17.4.2 SHALLOW TRIANGULATIONS

The concept of shallow triangulation is motivated by an attempt to understand the case of equality in the last main result mentioned above. See [Bay93, BL93].

GLOSSARY

The following definitions concern triangulations T of a finite set V of vertices of a convex d -polytope P .

Carrier: If F is a face of T , the carrier $C(F)$ of F is the smallest face of P containing F .

Shallow: If $\dim(C(F)) \leq 2\dim(F)$ for every face F of T , then T is a shallow triangulation.

Weakly neighborly: If all triangulations of V are shallow, then P is weakly neighborly.

Equidecomposable: If all triangulations of V have the same f -vector, then P is equidecomposable.

Stacked: If P has a triangulation in which there are no interior faces of dimension less than $d - 1$, then P is stacked.

k -stacked: If P is a simplicial d -polytope that has a triangulation in which there are no interior faces of dimension less than $d - k$, then P is k -stacked. In particular, a simplicial polytope is stacked if and only if it is 1-stacked.

MAIN RESULTS

1. A polytope P is weakly neighborly if and only if every set of $k + 1$ vertices is contained in a face of dimension at most $2k$ for all k [Bay93].
2. If P is weakly neighborly, then P is equidecomposable.
3. If T is a shallow triangulation of a polytope P , then $h(T) = h(P)$ and $h(\partial T) = h(\partial P)$ [Bay93].
4. If T is a triangulation of a polytope P with rational vertices and $h(T) = h(P)$, then T is shallow. Hence, if P is a rational polytope and $h(T) = h(P)$ for all triangulations T of P , then P is weakly neighborly [Bay93].
5. If P is a simplicial d -polytope, then it has a shallow triangulation if and only if it is k -stacked for some $1 \leq k \leq d/2$. In this case there is exactly one triangulation T of P having no interior faces of dimension less than $d - k$ (and this triangulation is the unique shallow one) [Bay93].
6. Suppose P is a d -polytope where $d > 3$. Then P is 1-stacked if and only if $g_2(\partial P) = 0$ [Bar71, Bar73].
7. Suppose P is a simplicial d -polytope such that $g_k(\partial P) = 0$ for some k with $3 \leq k \leq \lfloor d/2 \rfloor$. Then there is another simplicial d -polytope that has the same f -vector and is $(k-1)$ -stacked [KL84]. It is an open problem whether P itself

is always $(k-1)$ -stacked under this hypothesis [MW71]; this is known to be true if $f_0(P) \leq d+3$ or $k < f_0(P)/(f_0(P)-d)$ [Lee91b].

Some classes of weakly neighborly polytopes are given below [Bay93]:

- In dimension less than 3, all polytopes are weakly neighborly.
- In dimension 3, the only weakly neighborly polytopes are pyramids (over polygons) and the triangular prism.
- The product of two simplices of any dimensions is weakly neighborly. (See [Section 17.5.1](#) for the definition of product.)
- The only simplicial weakly neighborly polytopes are simplices and even-dimensional neighborly polytopes (those for which every subset of $d/2$ vertices determines a face of the polytope).
- Lawrence polytopes are weakly neighborly. (**Lawrence polytopes** are polytopes with centrally symmetric Gale diagrams; see [Chapters 6 and 16](#) of this Handbook for the definition of a Gale diagram. Equivalently, a Lawrence polytope is the result of executing a Lawrence extension on each vertex of a given arbitrary polytope; refer to Chapter 16 of this Handbook for the definition of a Lawrence extension. See also [Bay93, Zie95].)
- Pyramids over weakly neighborly polytopes are weakly neighborly.
- Subpolytopes of weakly neighborly polytopes are weakly neighborly.

17.4.3 RELATIONSHIPS TO COUNTING LATTICE POINTS

Triangulations of polytopes can be used to enumerate lattice points in polytopes [Sta96].

GLOSSARY

Integral: A polytope is integral if every vertex has integer coordinates.

$i(P, n)$: For an integral polytope P and a nonnegative integer n , $i(P, n)$ is the number of points $x \in P$ for which nx has integer coordinates. Equivalently, it is the number of integer points in nP .

Compressed ordering: An ordering of the vertices of an integral polytope P is compressed if every d -dimensional simplex of the triangulation obtained by pulling the vertices of P in that order has volume $1/d!$. P itself is **compressed** if every ordering of its vertices is compressed. (For example, the standard d -dimensional unit cube is compressed.)

MAIN RESULTS

Results (1) through (4) below are discussed in [Sta96].

1. $i(P, n)$ is a polynomial in n of degree d , called the Ehrhart polynomial of P (see Chapter 7 of this Handbook).
2. For integral d -polytope P write $J(P, t) = 1 + \sum_{n=1}^{\infty} i(P, n)t^n$. Then $J(P, t) = W(P, t)/(1 - t)^{d+1}$, where $W(P, t)$ is a polynomial of degree at most d with nonnegative integer coefficients.
3. If P is an integral d -polytope with compressed order σ , then

$$i(P, n) = \sum_{i=0}^d \binom{n-1}{i} f_i(T),$$

and $W(P, t) = h_0(T) + h_1(T)t + \cdots + h_d(T)t^d$, where T is the pulling triangulation induced by σ .

4. If P is a compressed integral d -polytope and σ is an ordering of its vertices, then the f -vector of the triangulation induced by σ depends only on P , not on σ .

For example, if P is the standard unit 3-cube, then any ordering σ produces a compressed triangulation T with h -vector $h(T) = (1, 4, 1, 0, 0)$. Thus $J(P, t) = (1 + 4t + t^2)/(1 - t)^4 = (1 + 4t + t^2)(1 + 4t + 10t^2 + 20t^3 + 35t^4 + \cdots) = 1 + 8t + 27t^2 + 64t^3 + 125t^4 + \cdots$.

17.5 SOME PARTICULAR TRIANGULATIONS

We gather together some information on some particular triangulations, including triangulations of the product of two simplices, the d -dimensional cube, the convex n -gon, and complete barycentric subdivisions.

17.5.1 PRODUCT OF TWO SIMPLICES

Consider the $(k+l)$ -polytope $P = \Delta_k \times \Delta_l$, the product of a k -dimensional simplex Δ_k and an l -dimensional simplex Δ_l . We consider triangulations of P using the points in its vertex set V . See [BCS88, deL96, GKZ94, Hai91].

GLOSSARY

Product: If P is a subset of \mathbb{R}^k and Q is a subset of \mathbb{R}^l , then the product of P and Q is the subset of \mathbb{R}^{k+l} given by $\{(v, w) \mid v \in P, w \in Q\}$.

MAIN RESULTS

1. As mentioned before, $P = \Delta_k \times \Delta_l$ is weakly neighborly, and so every triangulation has the same f -vector and h -vector. In particular, if T is a triangulation of P , then $f_{k+l}(T) = (k+l)!/(k!l!)$, and $h_i(T) = \binom{k}{i} \binom{l}{i}$ for $0 \leq i \leq k+l$ (with $h_i(T)$ taken to be zero if $i > \min\{k, l\}$) [BCS88].

2. Given a triangulation $\{P_1, \dots, P_s\}$ of a k -polytope P and a triangulation $\{Q_1, \dots, Q_t\}$ of an l -polytope Q , then there is a triangulation of $P \times Q$ using $s \cdot t \cdot (k+l)!/(k!l!)$ simplices of dimension $k+l$. To see this, observe that $\{P_i \times Q_j \mid 1 \leq i \leq s, 1 \leq j \leq t\}$ is a subdivision of $P \times Q$. Now refine this subdivision to a triangulation by, for example, pulling the vertices of $P \times Q$. Each $P_i \times Q_j$ will thereby be refined into $(k+l)!/(k!l!)$ simplices [Hai91].
3. All triangulations of $\Delta_2 \times \Delta_3$ and $\Delta_2 \times \Delta_4$ are regular. On the other hand, if $k, l \geq 3$, then there exist nonregular triangulations of $\Delta_k \times \Delta_l$ [deL96].

To describe one triangulation of $\Delta_k \times \Delta_l$ explicitly [BCS88, GKZ94], assume that Δ_k has vertex set $\{v_0, \dots, v_k\}$ and that Δ_l has vertex set $\{w_0, \dots, w_l\}$. Then $P = \Delta_k \times \Delta_l$ has vertex set $\{(v_i, w_j) \mid 0 \leq i \leq k, 0 \leq j \leq l\}$.

Consider paths from the vertex (v_0, w_0) to the vertex (v_k, w_l) in which each step involves increasing either the index of v or the index of w by one. Each such path selects a subset of $k+l+1$ vertices of P , which determines a $(k+l)$ -dimensional simplex. The collection of simplices associated with all such paths constitutes a triangulation of P . This is the same triangulation of P as the one obtained by starting with the trivial subdivision of P and pulling the vertices in the order

$$\begin{aligned} & (v_0, w_0), (v_0, w_1), \dots, (v_0, w_l), \\ & (v_1, w_0), (v_1, w_1), \dots, (v_1, w_l), \\ & \quad \vdots \\ & (v_k, w_0), (v_k, w_1), \dots, (v_k, w_l). \end{aligned}$$

Figure 17.5.1 shows this triangulation for $\Delta_2 \times \Delta_1$, a prism. The label ij on a vertex is an abbreviation for (v_i, w_j) .

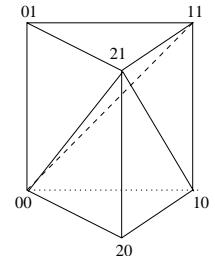


FIGURE 17.5.1
A triangulation of $\Delta_2 \times \Delta_1$.

17.5.2 d -CUBES

Here we consider triangulations of a d -dimensional cube using only the set V of its vertices. See [Hai91, OS03].

GLOSSARY

d -cube: The unit d -dimensional cube I^d is the d -fold product of the unit interval $I = [0, 1]$ with itself.

Index: A vertex of the d -dimensional cube is a point of the form $(a_1, \dots, a_d) \in \{0, 1\}^d$. Define the index of the vertex to be $\sum_{i=0}^{d-1} a_{i+1} 2^i$.

Size: The size of a triangulation T is the number $f_d(T)$ of d -simplices in T .

$\varphi(d)$: The size of the smallest triangulation of I^d . That is, $\varphi(d) = \min\{f_d(T) \mid T \text{ is a triangulation of } I^d\}$.

MAIN RESULTS

1. The maximum size of a triangulation of I^d is $d!$ (since the minimum volume of a d -simplex using the vertices of I^d is $1/d!$), and this is achievable for every d by pulling the vertices in any order.
2. $\varphi(d) \geq 2^d(d+1)^{-(d+1)/2}d!$. This bound is derived by observing that I^d can be inscribed in a sphere of diameter \sqrt{d} , and that the maximum volume of a simplex contained in this sphere is $(d+1)^{(d+1)/2}/(2^d d!)$ (the volume of a regular simplex) [Hai91].
3. There are precisely 74 triangulations of the 3-cube, and these fall into 6 classes of combinatorially different types [Big91, deL95]. All are regular. On the other hand, if $d \geq 4$, then not all triangulations of the d -cube are regular [deL96].
4. If I^d can be triangulated into $T(d)$ simplices, then I^{kd} can be triangulated into $[(kd)!/(d!)^k]T(d)^k = \rho^{kd}(kd)!$ simplices, where $\rho = (T(d)/d!)^{1/d}$. One measure of the efficiency of a triangulation is the number ρ . This result shows that any value of ρ achievable for one triangulation is achievable asymptotically. The smallest value of ρ obtainable from triangulations to date is $\rho \approx 0.816$ [OS03].
5. It is not known whether smaller size triangulations of I^d can be constructed if additional points other than vert(I^d) are allowed, but there are examples of other polytopes for which this happens [BBdR00].

Table 17.5.1 lists the known values of $\varphi(d)$ [Hug93, HA96].

TABLE 17.5.1 Minimal triangulations of d -cubes.

d	1	2	3	4	5	6	7
$\varphi(d)$	1	2	5	16	67	308	1493

It is also known that the smallest size of a triangulation of I^6 that slices off alternate vertices of I^6 is 324 [Hug93], and the smallest size of a triangulation of I^7 that slices off alternate vertices of I^7 is 1820 [HA96].

Figure 17.5.2 shows a triangulation of the 3-cube of size 5.

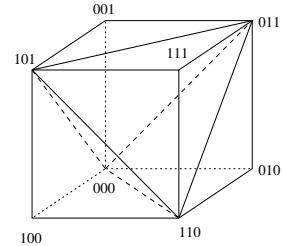


FIGURE 17.5.2

A minimum size triangulation of the 3-cube.

SOME SPECIFIC TRIANGULATIONS OF I^d

Pushing vertices: Start from the trivial subdivision of I^d and push the vertices in order of decreasing index (or place them in order of increasing index). The resulting triangulation will have $d!$ simplices [Big91].

Pulling vertices: Start from the trivial subdivision of I^d and pull the vertices in order of increasing index to obtain the triangulation T . Pulling the vertices in any order yields a triangulation with $d!$ simplices, so $f_d(T) = d!$. $h_d(T) = h_{d+1}(T) = 0$ and $h_i(T) = A(d, i)$, $0 \leq i \leq d - 1$, where $A(d, i)$ is the Eulerian number (it equals the number of permutations of $\{1, \dots, d\}$ having exactly i descents). There is a one-to-one correspondence between the simplices in T and the permutations of $\{1, \dots, d\}$, given in the following way: For a given permutation σ , the corresponding simplex has vertices $(0, \dots, 0) + e_{\sigma(1)} + e_{\sigma(2)} + \dots + e_{\sigma(k)}$, $0 \leq k \leq d$, where e_i denotes the standard i th unit vector. This is also known as Kuhn's triangulation [Big91, Tod76].

Sallee's corner slicing triangulation: Assume $d \geq 3$. For each vertex with an odd number of coordinates equaling 1, construct the simplex consisting of this vertex and its d neighbors (those joined to this vertex by an edge). These simplices, together with the central polytope remaining when these simplices are removed, constitute a subdivision of I^d . Refine this subdivision to a triangulation by pulling the vertices in order of increasing index. This triangulation has size $O(d!)$ [Hai91, Sal82].

Sallee's middle cut triangulation: Assume $d \geq 2$. Slice the cube into two polytopes by the hyperplane $x_1 + \dots + x_d = \lfloor d/2 \rfloor$. Refine this subdivision to a triangulation by pulling the vertices in order of increasing index. This triangulation has size $O(d!/d^2)$ [Sal84].

Haiman's triangulation: This triangulation method, which bootstraps a triangulation of I^d to a triangulation of I^{kd} as described in Section 17.5.1, Main Result 2, has size $O(\rho^d d!)$, where $\rho < 1$ [Hai91].

EXAMPLES

Figure 17.5.3 shows two triangulations of the 3-cube: (a) the one resulting from pulling the vertices in order of increasing index, and (b) the one resulting from pushing the vertices in order of decreasing index (equivalently, placing the vertices in order of increasing index).

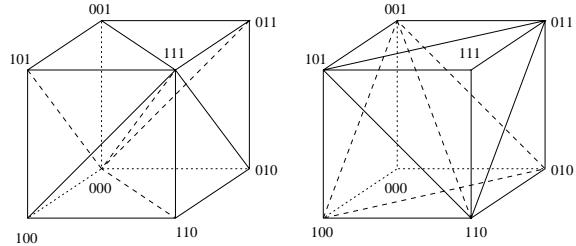


FIGURE 17.5.3

- (a) *The pulling triangulation of the 3-cube.*
- (b) *The pushing triangulation of the 3-cube.*

17.5.3 CONVEX n -GONS

There is no difficulty in finding subdivisions and triangulations of a convex n -gon using its set V of vertices. All subdivisions are regular, and all triangulations are constructible by pushing (or placing). Any subdivision is determined by a collection of mutually noncrossing internal diagonals. The set of all triangulations of the n -gon is isomorphic to many other combinatorial structures, including the set of all ways to parenthesize a string of $n - 1$ symbols and the set of all rooted binary trees with $n - 2$ nodes. See [Lee89, Zie95].

MAIN RESULTS

1. There are $\frac{1}{n-1} \binom{n-3}{j} \binom{n+j-1}{j+1}$ subdivisions of a convex n -gon having exactly j diagonals, $0 \leq j \leq n - 3$. In particular, the number of triangulations is the **Catalan number** $\frac{1}{n-1} \binom{2n-4}{n-2}$.
2. Two triangulations are **adjacent** if they share all but one diagonal. The **distance** between two triangulations T and T' is the length k of the shortest path $T = T_0, T_1, T_2, \dots, T_k = T'$ of triangulations in which T_i and T_{i-1} are adjacent for all $1 \leq i \leq k$. The distance between two triangulations of a convex n -gon does not exceed $2n - 6$ [Luc89]. This bound is achievable for infinitely many values of n [STT88].
3. The set of all triangulations of a convex n -gon is connected by a Hamiltonian cycle—a closed path $T_0, T_1, T_2, \dots, T_m = T_0$ containing each triangulation exactly once (except for T_0 , which starts and ends the path), in which T_i and T_{i-1} are adjacent for all i , $1 \leq i \leq m$ [Luc87].

17.5.4 COMPLETE BARYCENTRIC SUBDIVISIONS

For a given d -polytope P , let V be the collection of the centroids of the nonempty faces. Give the centroid of each k -dimensional face the label k , $0 \leq k \leq d$. Note that points labeled 0 are the vertices of P . Triangulate P by pulling the points of V in order of nonincreasing label. The resulting triangulation is the **complete barycentric subdivision** of P . The procedure can be extended in the obvious way to be applied to any polytopal complex. See [Bay88].

Figure 17.5.4 shows the complete barycentric subdivision of a 3-cube, a triangulation of size 48—there are eight pyramids into the center of the cube from each of the six original facets.

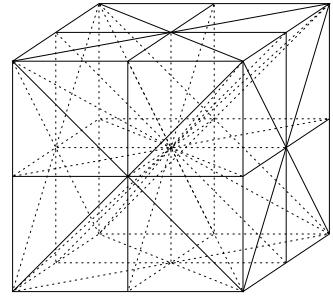


FIGURE 17.5.4

The complete barycentric subdivision of a 3-cube.

MAIN RESULTS

1. For every polytope P , there is a **dual polytope** (or polar polytope: see [Chapter 16](#) of this Handbook) P^* of the same dimension, whose face lattice is anti-isomorphic to that of P . The complete barycentric subdivisions T and T^* of P and P^* , respectively, are combinatorially isomorphic. That is to say, there is a bijection between the vertices of T and of T^* such that a subset of vertices of T determines a simplex in T precisely when the corresponding subset of vertices of T^* determines a simplex in T^* .
2. If T is the complete barycentric subdivision of a polytope P , then the combinatorial structure of the face lattice of P (up to lattice reversal by the previous result) can be recovered from the combinatorial structure of T , even if one is not given the specific geometric realization or the labels of the points [Bay88].
3. Suppose T is the complete barycentric subdivision of a d -dimensional simplex. Then $f_d(T) = d!$, $h_{d+1}(T) = 0$, and $h_i(T) = A(d+1, i)$, $0 \leq i \leq d$. These are the Eulerian numbers encountered in Kuhn's triangulation of I^{d+1} . In fact, Kuhn's triangulation is combinatorially isomorphic to the join of T to a new point (make a pyramid with this new point over every d -simplex in T) [Big91].
4. If T is the complete barycentric subdivision of I^d , then $f_d(T) = d!2^d$. Also, $h_{d+1}(T) = 0$, and $h_i(T)$ equals the number of signed permutations of $\{1, \dots, d\}$ with exactly i descents [Bre94].

17.6 SECONDARY AND FIBER POLYTOPES

This section concerns itself with the structure of the collection of all regular subdivisions of a given finite set of points $V = \{v_1, \dots, v_n\} \subset \mathbb{R}^d$. See [GKZ94, Lee91a, Zie95]. Assume that $\dim(\text{conv}(V)) = d$.

GLOSSARY

z -vector: Suppose T is a triangulation of V . Define the z -vector $z(T) = (z_1, \dots, z_n) \in \mathbb{R}^n$ by $z_i = \sum \text{vol}(F)$, where the sum is taken over all d -simplices F in T having v_i as a vertex.

Secondary polytope: The secondary polytope $\Sigma(V)$ is the convex hull of the z -vectors of all triangulations of V .

Link: If F is a face of a triangulation T , then the link of F is the set $\{G \mid G \text{ is a face of } T, F \cup G \text{ is a face of } T \text{ of dimension } \dim F + \dim G + 1, \text{ and } F \cap G = \emptyset\}$.

Adjacent triangulations: Suppose T is a triangulation of V . Suppose there is a subset W of $k+2$ points in V such that $\dim(\text{aff}(W)) = k$, T contains one of the (only) two triangulations of W , and the links with respect to T of all the k -dimensional faces F in the triangulation of W are identical. Then it is possible to interchange the triangulations of W , giving the new k -simplices the same links with respect to T , and thereby obtain a new triangulation of V . This operation is called a *flip*, and the resulting triangulation is said to be adjacent to T .

Connected: Two triangulations are said to be connected if one can be obtained from the other by a sequence of flips. The set of triangulations, under adjacency by flips, forms a graph.

The secondary polytope plays an important role in the study of Gröbner bases [Stu96] and generalized discriminants and determinants [GKZ94].

MAIN RESULTS

1. The collection of all regular subdivisions of the set V , partially ordered by refinement, is combinatorially equivalent to the boundary complex of the secondary polytope $\Sigma(V)$, which has dimension $n-d-1$ [GKZ94].
2. The vertices of $\Sigma(V)$ are precisely the z -vectors of the regular triangulations. In particular, no two regular triangulations have the same z -vector. The edges of $\Sigma(V)$ correspond to adjacent regular triangulations [GKZ94].
3. $\Sigma(V)$ can also be expressed as a discrete or continuous Minkowski sum of polytopes coming from a representation of V as a projection of the vertices of an $(n-1)$ -dimensional simplex. See [BFS90, BS92, Zie95].
4. Suppose $S = \{S_1, \dots, S_m\}$ is a regular triangulation of $V = \{v_1, \dots, v_n\} \subset \mathbb{R}^d$ determined by lifting numbers $\alpha_1, \dots, \alpha_n$. Let $f : \text{conv}(V) \rightarrow \mathbb{R}$ be the piecewise-linear convex function whose graph is given by the lower facets of $Q = \text{conv}(\{(v_1, \alpha_1), \dots, (v_n, \alpha_n)\})$. Define c_j to be the centroid of the polytope $P_j = \text{conv}(S_j)$, $1 \leq j \leq m$. Then the inequality

$$\sum_{i=1}^n \alpha_i z_i \geq (d+1) \sum_{j=1}^m \text{vol}(P_j) f(c_j)$$

is valid for the secondary polytope and holds with equality precisely for those subdivisions refining S . In particular, it describes a facet of $\Sigma(V)$ if S is a minimal nontrivial regular subdivision. These facet-defining inequalities, together with the set of $d+1$ equations

$$\sum_{i=1}^n z_i = (d+1)\text{vol}(P),$$

$$\sum_{i=1}^n z_i v_i = (d+1)\text{vol}(P)c,$$

where c is the centroid of $P = \text{conv}(V)$, fully describe $\Sigma(V)$ [C.W. Lee, unpublished].

5. As an immediate consequence of the existence of $\Sigma(V)$, every regular triangulation has at least $n - d - 1$ adjacent triangulations, and every pair of regular triangulations is connected.
6. In the special case that $n = d + 2$, there are precisely two nontrivial subdivisions of V (both regular), so $\Sigma(V)$ is a line segment.
7. In the special case that $n = d + 3$, all subdivisions are regular, and $\Sigma(V)$ is a convex polygon [Lee91a].
8. In the special case that V is the set of vertices of a convex n -gon, $\Sigma(V)$ is called the ***associahedron*** [Lee89]. Its dual is a simplicial polytope Q of dimension $n - 3$ having the following f -vector and h -vector:

$$f_{j-1}(Q) = \frac{1}{n-1} \binom{n-3}{j} \binom{n+j-1}{j+1}, \quad 0 \leq j \leq n-3,$$

$$h_i(Q) = \frac{1}{n-1} \binom{n-3}{i} \binom{n-1}{i+1}, \quad 0 \leq i \leq n-3.$$

From the discussion in Section 17.5.3, $f_{j-1}(Q)$ is the number of subdivisions of the n -gon having exactly j diagonals. There are various combinatorial interpretations of the h -vector. Explicit coordinates and inequalities for $\Sigma(V)$ can be found in [Zie95].

9. It is not necessarily the case that the collection of all triangulations, whether regular or not, is connected, though this is the case for triangulations of point sets in \mathbb{R}^2 , for triangulations of vertex sets of cyclic polytopes [Ram96], and for the special case that $n = d + 4$ (for which the graph of triangulations is 3-connected, and in particular every triangulation has at least 3 neighbors) [AS00]. The first example found of a triangulation of a point set with no adjacent triangulations is of a certain set of 324 points in \mathbb{R}^6 [San00]. There are now examples of point sets of size 50 and 26 that are vertex sets of 5-polytopes whose triangulation graphs are disconnected [San02].

Figure 17.6.1 shows the five regular triangulations of a set of 5 points in \mathbb{R}^2 , marking which pairs of triangulations are adjacent.

The secondary polytope of the product of two simplices is discussed, for example, in [deL96, GKZ94]. See [dHSS96] for properties of the polytope that is the convex hull of the $(0, 1)$ incidence vectors of all triangulations of V , and for the relationship of this polytope to $\Sigma(V)$. The special case when V is the set of vertices of a convex n -gon was first described in [DHH85].

17.6.1 FIBER POLYTOPES

A secondary polytope is a special case of a fiber polytope, which is associated with an affine map $\pi : P \rightarrow Q$ from a polytope P in \mathbb{R}^p onto a polytope Q in \mathbb{R}^q . Such

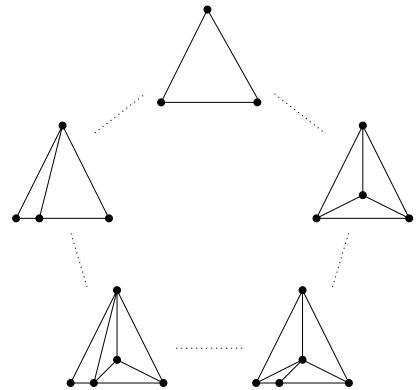


FIGURE 17.6.1
A polygon of regular triangulations.

a map induces certain regular subdivisions of Q (called π -coherent subdivisions). The fiber polytope $\Sigma(P, Q)$ has dimension $\dim(P) - \dim(Q)$, and its nonempty faces correspond to these π -coherent subdivisions.

A **section** is a continuous map $\gamma : Q \rightarrow P$ with $\pi(\gamma(x)) = x$ for all $x \in Q$. The **fiber polytope** is defined to be the set of all average values of the sections of π :

$$\Sigma(P, Q) = \left\{ \frac{1}{\text{vol}(Q)} \int_Q \gamma(x) dx \mid \gamma \text{ is a section of } \pi \right\}.$$

The associahedron and the permutohedron (see [Chapter 16](#) of this Handbook) are examples of fiber polytopes, and there are applications to zonotopal subdivisions and oriented matroids. For more details, see [BS92, RZ94, Zie95]. A subdivision of Q is π -induced if it corresponds to some subcomplex of P that π maps bijectively onto Q . This subdivision is proper if it is not the trivial subdivision of Q , and such subdivisions form a poset under refinement, called the **Baues poset**. A natural extension of the connectivity questions on regular triangulations is the

PROBLEM 17.6.1 Generalized Baues Problem [BKS94]

When is the Baues poset homotopy equivalent to a sphere of dimension $\dim P - \dim Q - 1$?

See [Rei99] for a survey of this problem.

17.7 SOURCES AND RELATED MATERIAL

FURTHER READING

[Chapter 25](#) discusses triangulations of more general (e.g., nonconvex) objects. [Chapter 23](#) provides details on Delaunay triangulations and Voronoi diagrams. Refer also to Chapter 16, on basic properties of convex polytopes.

A section on triangulations and subdivisions of convex polytopes can be found in the survey article [BL93]. The book [Zie95] and the article [Lee91a] contain information on regular subdivisions and triangulations; for their important role in generalized discriminants and determinants see the book [GKZ94], and for their

significance in computational algebra see the book [Stu96]. Additional references can be found in the above-mentioned sources, as well as the citations given in this chapter.

RELATED CHAPTERS

- [Chapter 3: Tilings](#)
 - [Chapter 6: Oriented matroids](#)
 - [Chapter 7: Lattice points and lattice polytopes](#)
 - [Chapter 16: Basic properties of convex polytopes](#)
 - [Chapter 18: Face numbers of polytopes and complexes](#)
 - [Chapter 22: Convex hull computations](#)
 - [Chapter 23: Voronoi diagrams and Delaunay triangulations](#)
 - [Chapter 25: Triangulations and mesh generation](#)
 - [Chapter 31: Computational convexity](#)
-

REFERENCES

- [AS00] M. Azaola and F. Santos. The graph of triangulations of a point configuration with $d+4$ vertices is 3-connected. *Discrete Comput. Geom.*, 23:489–536, 2000.
- [AS02] M. Azaola and F. Santos. The number of triangulations of the cyclic polytope $C(n, n-4)$. *Discrete Comput. Geom.*, 27:29–48, 2002.
- [Bar71] D.W. Barnette. The minimum number of vertices of a simple polytope. *Israel J. Math.*, 10:121–125, 1971.
- [Bar73] D.W. Barnette. A proof of the lower bound conjecture for convex polytopes. *Pacific J. Math.*, 46:349–354, 1973.
- [Bay88] M.M. Bayer. Barycentric subdivisions. *Pacific J. Math.*, 135:1–16, 1988.
- [Bay93] M.M. Bayer. Equidecomposable and weakly neighborly polytopes. *Israel J. Math.*, 81:301–320, 1993.
- [BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. Elsevier, Amsterdam, 1993.
- [BBdR00] A. Below, U. Brehm, J. de Loera, J. Richter-Gebert. Minimal simplicial dissections and triangulations of convex 3-polytopes. *Discrete Comput. Geom.*, 24:35–48, 2000.
- [Big91] F. Biggeli. *Regular Triangulations of Convex Polytopes and d-Cubes*. Ph.D. thesis, Univ. of Kentucky, Lexington, 1991.
- [BCS88] L.J. Billera, R. Cushman, and J.A. Sanders. The Stanley decomposition of the harmonic oscillator. *Nederl. Akad. Wetensch. Indag. Math.*, 50:375–393, 1988.
- [BFS90] L.J. Billera, P. Filliman, B. Sturmfels. Constructions and complexity of secondary polytopes. *Adv. Math.*, 83:155–179, 1990.
- [BKS94] L.J. Billera, M.M. Kapranov, and B. Sturmfels. Cellular strings on polytopes. *Proc. Amer. Math. Soc.*, 122:549–555, 1994.
- [BL81] L.J. Billera and C.W. Lee. The numbers of faces of polytope pairs and unbounded polyhedra. *European J. Combin.*, 2:307–322, 1981.
- [BM84] L.J. Billera and B.S. Munson. Triangulations of oriented matroids and convex polytopes. *SIAM J. Algebraic Discrete Methods*, 5:515–525, 1984.

- [BS92] L.J. Billera and B. Sturmfels. Fiber polytopes. *Ann. of Math.* (2), 135:527–549, 1992.
- [BM71] H. Bruggesser and P. Mani. Shellable decompositions of cells and spheres. *Math. Scand.*, 29:197–205, 1971.
- [Bre94] F. Brenti. q -Eulerian polynomials arising from Coxeter groups. *European J. Combin.*, 15:417–441, 1994.
- [DHH85] G.B. Dantzig, A.J. Hoffman, and T.C. Hu. Triangulations (tilings) and certain block triangular matrices. *Math. Programming*, 31:1–14, 1985.
- [deL95] J. de Loera. *Triangulations of Polytopes and Computational Algebra*. Ph.D. thesis, Cornell Univ., Ithaca, 1995.
- [deL96] J. de Loera. Nonregular triangulations of products of simplices. *Discrete Comput. Geom.*, 15:253–264, 1996.
- [dHSS96] J. de Loera, S. Hoşten, F. Santos, and B. Sturmfels. The polytope of all triangulations of a point configuration. *Doc. Math.*, 1:103–119, 1996.
- [ES96] H. Edelsbrunner and N.R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.
- [GKZ94] I.M. Gel'fand, M.M. Kapranov, and A.V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants*. Birkhäuser, Boston, 1994.
- [GP88] J.E. Goodman and J. Pach. Cell decomposition of polytopes by bending. *Israel J. Math.*, 64:129–138, 1988.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Wiley, New York, 1967.
- [Hai91] M. Haiman. A simple and relatively efficient triangulation of the n -cube. *Discrete Comput. Geom.*, 6:287–289, 1991.
- [Hud69] J.F.P. Hudson. *Piecewise Linear Topology*. Benjamin, New York, 1969.
- [Hug93] R.B. Hughes. Minimum-cardinality triangulations of the d -cube for $d = 5$ and $d = 6$. *Discrete Math.*, 118:75–118, 1993.
- [HA96] R.B. Hughes and M.R. Anderson. Simplexity of the cube. *Discrete Math.*, 158:99–150, 1996.
- [KL84] P. Kleinschmidt and C.W. Lee. On k -stacked polytopes. *Discrete Math.*, 48:125–127, 1984.
- [Lee89] C.W. Lee. The associahedron and triangulations of the n -gon. *European J. Combin.*, 10:551–560, 1989.
- [Lee85] C.W. Lee. Triangulating the d -cube. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440 of *Ann. New York Acad. Sci.*, pages 205–211, New York Acad. Sci., New York, 1985.
- [Lee91a] C.W. Lee. Regular triangulations of convex polytopes. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.*, pages 443–456. Amer. Math. Soc., Providence, 1991.
- [Lee91b] C.W. Lee. Winding numbers and the generalized lower-bound conjecture. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 209–219, Amer. Math. Soc., Providence, 1991.
- [Luc87] J.M. Lucas. The rotation graph of binary trees is Hamiltonian. *J. Algorithms*, 8:503–535, 1987.
- [Luc89] F. Luccio. On the upper bound on the rotation distance of binary trees. *Inform. Process. Lett.*, 31:57–60, 1989.

- [McM70] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.
- [MS71] P. McMullen and G.C. Shephard. *Convex Polytopes and the Upper Bound Conjecture*. Volume 3 of *London Math. Soc. Lecture Note Ser.*, Cambridge Univ. Press, 1971.
- [MW71] P. McMullen and D.W. Walkup. A generalized lower-bound conjecture for simplicial polytopes. *Mathematika*, 18:264–273, 1971.
- [OS03] D. Orden and F. Santos. Asymptotically efficient triangulations of the d -cube. *Discrete Comput. Geom.*, 30:509–518, 2003.
- [Ram96] J. Rambau. Triangulations of cyclic polytopes and higher Bruhat orders. *Mathematika*, 44:162–194, 1997.
- [Rei99] V. Reiner. The generalized Baues problem. In *New Perspectives in Algebraic Combinatorics (Berkeley, CA, 1996–97)*, volume 38 of *Math. Sci. Res. Inst. Publ.*, pages 293–336, Cambridge Univ. Press, 1999.
- [RZ94] J. Richter-Gebert and G.M. Ziegler. Zonotopal tilings and the Bohne-Dress theorem. In H. Barcelo and G. Kalai, editors, *Jerusalem Combinatorics, '93*, pages 211–232, Amer. Math. Soc., Providence, 1994.
- [Rud58] M.E. Rudin. An unshellable triangulation of a tetrahedron. *Bull. Amer. Math. Soc.*, 64:90–91, 1958.
- [Sal82] J.F. Sallee. A triangulation of the n -cube. *Discrete Math.*, 40:81–86, 1982.
- [Sal84] J.F. Sallee. The middle-cut triangulations of the n -cube. *SIAM J. Algebraic Discrete Methods*, 5:407–419, 1984.
- [San00] F. Santos. A point set whose space of triangulations is disconnected. *J. Amer. Math. Soc.*, 13:611–637, 2000.
- [San02] F. Santos. Non-connected toric Hilbert schemes. Preprint, 2002.
- [STT88] D.D. Sleator, R.E. Tarjan, and W.P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Amer. Math. Soc.*, 1:647–681, 1988.
- [Sta80] R.P. Stanley. Decompositions of rational convex polytopes. *Ann. Discrete Math.*, 6:333–342, 1980.
- [Sta92] R.P. Stanley. Subdivisions and local h -vectors. *J. Amer. Math. Soc.*, 5:805–851, 1992.
- [Sta96] R.P. Stanley. *Combinatorics and Commutative Algebra*. Second edition. Birkhäuser, Boston, 1996.
- [Stu91] B. Sturmfels. Gröbner bases of toric varieties. *Tohoku Math. J.*, 43:249–261, 1991.
- [Stu96] B. Sturmfels. *Gröbner Bases and Convex Polytopes*. Volume 8 of *Univ. Lecture Ser.*, Amer. Math. Soc., Providence, 1996.
- [Tod76] M.J. Todd. *The Computation of Fixed Points and Applications*. Volume 124 of *Lecture Notes in Econom. and Math. Systems*, Springer-Verlag, Berlin, 1976.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

18 FACE NUMBERS OF POLYTOPES AND COMPLEXES

Louis J. Billera and Anders Björner

INTRODUCTION

Geometric objects are often put together from simple pieces according to certain combinatorial rules. As such, they can be described as *complexes* with their constituent *cells*, which are usually polytopes and often simplices. Many constraints of a combinatorial and topological nature govern the incidence structure of cell complexes and are therefore relevant in the analysis of geometric objects. Since these incidence structures are in most cases too complicated to be well understood, it is worthwhile to focus on simpler invariants that still say something nontrivial about their combinatorial structure. The invariants to be discussed in this chapter are the *f-vectors* $f = (f_0, f_1, \dots)$, where f_i is the number of i -dimensional cells in the complex.

The theory of *f*-vectors can be discussed at two levels: (1) the numerical relations satisfied by the f_i numbers, and (2) the algebraic, combinatorial, and topological facts and constructions that give rise to and explain these relations. This chapter will summarize the main facts in the numerology of *f*-vectors (i.e., at level 1), with emphasis on cases of geometric interest.

The chapter is organized as follows. To begin with, we treat simplicial complexes, first the general case (Section 18.1), then complexes with various Betti number constraints (Section 18.2), and finally triangulations of spheres, polytope boundaries, and manifolds (Section 18.3). Then we move on to nonsimplicial complexes, discussing first the general case (Section 18.4) and then polytopes and spheres (Section 18.5).

18.1 SIMPLICIAL COMPLEXES

GLOSSARY

The convex hull of any set of $j + 1$ affinely independent points in \mathbb{R}^n is called a *j-simplex*. See [Chapter 16](#) for more about this definition, and for the notions of *faces* and *vertices* of a simplex.

A *geometric simplicial complex* Γ is a finite nonempty family of simplices in \mathbb{R}^n such that (i) $\sigma \in \Gamma$ implies that $\tau \in \Gamma$ for every face τ of σ , and (ii) if $\sigma, \tau \in \Gamma$ and $\sigma \cap \tau \neq \emptyset$ then $\sigma \cap \tau$ is a face of both σ and τ .

An *abstract simplicial complex* Δ is a finite nonempty family of subsets of some ground set V (the *vertex set*) such that if $A \in \Delta$ and $B \subseteq A$ then $B \in \Delta$. (Note that always $\emptyset \in \Delta$.) The elements $A \in \Delta$ are called *faces*.

Define the **dimension** of a face A and of Δ itself by $\dim A = |A| - 1$; $\dim \Delta = \max_{A \in \Delta} \dim A$. By a **d -complex** we mean a d -dimensional complex.

With every geometric simplicial complex Γ we associate an abstract simplicial complex by taking the family of vertex sets of its simplices. Conversely, every d -dimensional abstract simplicial complex Δ can be realized in \mathbb{R}^n for $n \geq 2d + 1$ (and sometimes less) by some geometric simplicial complex. The latter is unique up to homeomorphism, so it is correct to think of the realization map as a one-to-one correspondence between abstract and geometric simplicial complexes. We will therefore drop the adjectives “abstract” and “geometric” and speak only of a **simplicial complex**.

For a simplicial complex Δ , let $\Delta^i = \{i\text{-dimensional faces}\}$ and let $f_i = |\Delta^i|$. The integer sequence $f(\Delta) = (f_0, f_1, \dots)$ is called the **f -vector** of Δ . (The entry $f_{-1} = 1$ is usually suppressed.) The subcomplex $\Delta^{\leq i} = \bigcup_{j \leq i} \Delta^j$ is called the **i -skeleton** of Δ .

A simplicial complex Δ is called **pure** if all maximal faces are of equal dimension. It is called **r -colorable** if there exists a partition of the vertex set $V = V_1 \cup \dots \cup V_r$ such that $|A \cap V_i| \leq 1$ for all $A \in \Delta$ and $1 \leq i \leq r$. Equivalently, Δ is r -colorable if and only if its 1-skeleton $\Delta^{\leq 1}$ is r -colorable in the standard sense of graph theory. An $(r-1)$ -complex that is both pure and r -colorable is sometimes called **balanced**.

For integers $k, n \geq 1$ there is a unique way of writing

$$n = \binom{a_k}{k} + \binom{a_{k-1}}{k-1} + \dots + \binom{a_i}{i}$$

so that $a_k > a_{k-1} > \dots > a_i \geq i \geq 1$. Then define

$$\partial_k(n) = \binom{a_k}{k-1} + \binom{a_{k-1}}{k-2} + \dots + \binom{a_i}{i-1},$$

and

$$\delta^k(n) = \binom{a_k - 1}{k-1} + \binom{a_{k-1} - 1}{k-2} + \dots + \binom{a_i - 1}{i-1}.$$

Also let $\partial_k(0) = \delta^k(0) = 0$.

Let \mathbb{N}^∞ denote the set of sequences (n_0, n_1, \dots) of nonnegative integers, and $\mathbb{N}^{(\infty)}$ the subset of sequences such that $n_k = 0$ for all sufficiently large k . We call $n \in \mathbb{N}^{(\infty)}$ a **K -sequence** if

$$\partial_{k+1}(n_k) \leq n_{k-1} \quad \text{for all } k \geq 1.$$

We call $n \in \mathbb{N}^\infty$ an **M -sequence** if

$$n_0 = 1 \text{ and } \partial^k(n_k) \leq n_{k-1} \quad \text{for all } k \geq 2.$$

THE KRUSKAL-KATONA THEOREM AND SOME RELATIVES

The following basic result characterizes the f -vectors of simplicial complexes.

THEOREM 18.1.1 *Kruskal-Katona Theorem*

For $f = (f_0, f_1, \dots) \in \mathbb{N}^{(\infty)}$ the following are equivalent:

- (i) f is the f -vector of a simplicial complex;
- (ii) f is a K -sequence.

A simplicial complex is **connected** if its 1-skeleton is connected in the sense of graph theory.

THEOREM 18.1.2

For $f \in \mathbb{N}^{(\infty)}$ the following are equivalent:

- (i) f is the f -vector of a connected simplicial complex;
- (ii) f is a K -sequence and $\partial^3(f_2) \leq f_1 - f_0 + 1$.

Theorem 18.1.1 has a generalization to colored complexes, whose statement will require some additional definitions. Fix an integer $r > 0$. Then define $\binom{n}{k}_r$ as follows: partition $\{1, \dots, n\}$ into r subsets V_1, \dots, V_r as evenly as possible (so every subset V_i will have $\lfloor \frac{n}{r} \rfloor$ or $\lfloor \frac{n}{r} \rfloor + 1$ elements), and let $\binom{n}{k}_r$ be the number of k -subsets $F \subseteq \{1, \dots, n\}$ such that $|F \cap V_i| \leq 1$ for $1 \leq i \leq r$. For $k \leq r$ every positive integer n can be uniquely written

$$n = \binom{a_k}{k}_r + \binom{a_{k-1}}{k-1}_{r-1} + \dots + \binom{a_i}{i}_{r-k+i},$$

where $\frac{a_j}{a_{j-1}} > \frac{r-k+j}{r-k+j-1}$ for $j = k, k-1, \dots, i+1$, and $a_i \geq i \geq 1$. Then define

$$\partial_k^{(r)}(n) = \binom{a_k}{k-1}_r + \binom{a_{k-1}}{k-2}_{r-1} + \dots + \binom{a_i}{i-1}_{r-k+i},$$

and let $\partial_k^{(r)}(0) = 0$.

THEOREM 18.1.3

For $f = (f_0, \dots, f_{d-1})$, $d \leq r$, the following are equivalent:

- (i) f is the f -vector of an r -colorable simplicial complex;
- (ii) $\partial_{k+1}^{(r)}(f_k) \leq f_{k-1}$, for all $1 \leq k \leq d-1$.

Note that for r sufficiently large Theorem 18.1.3 specializes to Theorem 18.1.1.

MULTICOMPLEXES AND MACAULAY'S THEOREM

A **multicomplex** \mathcal{M} is a nonempty collection of monomials in finitely many variables such that if m is in \mathcal{M} then so is every divisor of m . Let $f_i(\mathcal{M})$ be the number of degree i monomials in \mathcal{M} ; $f(\mathcal{M}) = (f_0, f_1, \dots)$ is called the **f -vector** of \mathcal{M} .

THEOREM 18.1.4 Macaulay's Theorem

For $f \in \mathbb{N}^\infty$ the following are equivalent:

- (i) f is the f -vector of a multicomplex;
- (ii) f is an M -sequence;
- (iii) $f_i = \dim_k R_i$, $i \geq 0$, for some finitely generated commutative graded k -algebra $R = \bigoplus_{i \geq 0} R_i$ such that $R_0 \cong k$ (a field) and R_1 generates R .

A simplicial complex can be viewed as a multicomplex of squarefree monomials. Hence, a K -sequence is (except for a shift in the indexing) an M -sequence: If (f_0, \dots, f_{d-1}) is a K -sequence then $(1, f_0, \dots, f_{d-1})$ is an M -sequence. For this reason (and others, see, e.g., Theorem 18.2.2), properties of M -sequences are of interest also if one cares mainly about the special case of simplicial complexes.

A multicomplex is *pure* if all its maximal (under divisibility) monomials have the same degree.

THEOREM 18.1.5

Let (f_0, \dots, f_r) be the f -vector of a pure multicomplex, $f_r \neq 0$. Then $f_i \leq f_j$ for all $i < j \leq r - i$.

COMMENTS

Simplicial complexes (abstract and geometric) are treated in most books on algebraic topology; see, e.g., [Mun84, Spa66]. The Kruskal-Katona theorem (independently discovered by M.-P. Schützenberger, J.B. Kruskal, G.O.H. Katona, L.H. Harper, and B. Lindström during the years 1959–1966) is discussed in many places and several proofs have appeared; see, e.g., [And87, Zie95].

A Kruskal-Katona type theorem for simplicial complexes with vertex-transitive symmetry group appears in [FK96].

Theorems 18.1.2 and 18.1.3 are from [Bjö96] and [FFK88] respectively. (Remark: The definition of the $\partial_k^{(r)}(\cdot)$ operator is incorrectly stated in [FFK88], in particular the uniqueness claim in [FFK88, Lemma 1.1] is incorrect. The version stated here was suggested to us by J. Eckhoff.)

For Macaulay's theorem we refer to [And87, Sta96]. There is a common generalization of Macaulay's theorem and the Kruskal-Katona theorem due to Clements and Lindström; see [And87]. Theorem 18.1.5 is from [Hib89].

18.2 BETTI NUMBER CONSTRAINTS

GLOSSARY

The **Euler characteristic** $\chi(\Delta)$ of a simplicial complex Δ with f -vector (f_0, \dots, f_{d-1}) is $\chi(\Delta) = \sum_{i=0}^{d-1} (-1)^i f_i$.

The **h -vector** (h_0, \dots, h_d) of a $(d-1)$ -dimensional simplicial complex is defined by

$$\sum_{i=0}^d h_i x^{d-i} = \sum_{i=0}^d f_{i-1} (x-1)^{d-i}.$$

The corresponding **g -vector** $(g_0, \dots, g_{\lfloor d/2 \rfloor})$ is defined by $g_0 = 1$ and $g_i = h_i - h_{i-1}$, for $i \geq 1$.

The **Betti number** $\beta_i(\Delta)$ is the dimension (as a \mathbb{Q} -vector space) of the i th reduced simplicial homology group $\tilde{H}_i(\Delta, \mathbb{Q})$; see any textbook on algebraic topology (e.g., [Mun84]) for the definition. We call $(\beta_0, \dots, \beta_{\dim \Delta})$ the **Betti sequence** of Δ .

The **link** $\ell k_\Delta(F)$ of a face F is the subcomplex of Δ defined by $\ell k_\Delta(F) = \{A \in \Delta \mid A \cap F = \emptyset, A \cup F \in \Delta\}$. Note that $\ell k_\Delta(\emptyset) = \Delta$.

A simplicial complex Δ is **acyclic** if $\beta_i(\Delta) = 0$ for all i .

A simplicial complex Δ is **Cohen-Macaulay** if $\beta_i(\ell k_\Delta(F)) = 0$ for all $F \in \Delta$ and all $i < \dim \ell k_\Delta(F)$.

A simplicial complex Δ is **m -Leray** if $\beta_i(\ell k_\Delta(F)) = 0$ for all $F \in \Delta$ and all $i \geq m$.

FIXED BETTI NUMBERS

The most basic relationship between f -vectors and Betti numbers is the **Euler-Poincaré formula**:

$$\chi(\Delta) = f_0 - f_1 + f_2 - \dots = 1 + \beta_0 - \beta_1 + \beta_2 - \dots$$

This is in fact the only linear one in the following complete set of relations.

THEOREM 18.2.1

For $f = (f_0, f_1, \dots) \in \mathbb{N}^{(\infty)}$ and $\beta = (\beta_0, \beta_1, \dots) \in \mathbb{N}^{(\infty)}$ the following are equivalent:

- (i) f is the f -vector of some simplicial complex with Betti sequence β ;
- (ii) if $\chi_{k-1} = \sum_{j \geq k} (-1)^{j-k} (f_j - \beta_j)$, $k \geq 0$, then $\chi_{-1} = 1$ and $\partial_{k+1}(\chi_k + \beta_k) \leq \chi_{k-1}$ for all $k \geq 1$.

By putting $\beta_i = 0$ for all i one gets as a special case a characterization of the f -vectors of acyclic simplicial complexes, viz., $\sum_{i \geq 0} f_{i-1} x^i = (1+x) \sum_{i \geq 0} f'_{i-1} x^i$, where (f'_0, f'_1, \dots) is a K -sequence.

COHEN-MACAULAY COMPLEXES

Examples of Cohen-Macaulay complexes are triangulations of manifolds whose Betti numbers vanish below the top dimension, in particular triangulations of spheres and balls. Other examples are matroid complexes (the independent sets of a matroid), Tits buildings, and the order complexes (simplicial complex of totally ordered subsets) of several classes of posets, e.g., semimodular lattices (including distributive and geometric lattices). Shellable complexes (see Chapters 17 and 20) are Cohen-Macaulay. Cohen-Macaulay complexes are always pure.

The definition of h -vector given in the glossary shows that the h -vector and the f -vector of a complex mutually determine each other via the formulas:

$$h_i = \sum_{j=0}^i (-1)^{i-j} \binom{d-j}{i-j} f_{j-1}, \quad f_{i-1} = \sum_{j=0}^i \binom{d-j}{i-j} h_j,$$

for $0 \leq i \leq d$. Hence, we may state f -vector results in terms of h -vectors whenever convenient.

THEOREM 18.2.2

For $h = (h_0, \dots, h_d) \in \mathbb{Z}^{d+1}$ the following are equivalent:

- (i) h is the h -vector of a $(d-1)$ -dimensional Cohen-Macaulay complex;
- (ii) h is the h -vector of a $(d-1)$ -dimensional shellable complex;
- (iii) h is an M -sequence.

Since there are a total of $\binom{n+k-1}{k}$ monomials of degree k in n variables, and by Theorems 18.1.4 and 18.2.2 the h -vector of a $(d-1)$ -dimensional Cohen-Macaulay complex counts certain monomials in $h_1 = f_0 - d$ variables, we derive the inequalities

$$0 \leq h_i \leq \binom{f_0 - d + i - 1}{i}$$

for the h -vectors of Cohen-Macaulay complexes. The lower bound can be improved for complexes with fixed-point-free involutive symmetry.

THEOREM 18.2.3

Let $h = (h_0, \dots, h_d)$ be the h -vector of a Cohen-Macaulay complex admitting an automorphism α of order 2, such that $\alpha(F) \neq F$ for all $F \in \Delta \setminus \{\emptyset\}$. Then

$$h_i \geq \binom{d}{i} \quad \text{for } 0 \leq i \leq d.$$

Consequently, $f_{d-1} = h_0 + \dots + h_d \geq 2^d$.

Another condition on a Cohen-Macaulay complex that forces stricter conditions on its h -vector is being r -colorable.

THEOREM 18.2.4

For $h = (h_0, \dots, h_d) \in \mathbb{Z}^{d+1}$ the following are equivalent:

- (i) h is the h -vector of a $(d-1)$ -dimensional and d -colorable Cohen-Macaulay complex;
- (ii) (h_1, \dots, h_d) is the f -vector of a d -colorable simplicial complex.

Hence in this case the h -vector is not only an M -sequence, but the special kind of K -sequence characterized in Theorem 18.1.3.

LERAY COMPLEXES

Examples of Leray complexes arise as follows. Let $\mathcal{K} = \{K_1, \dots, K_t\}$ be a family of convex sets in \mathbb{R}^m , and let $\Delta(\mathcal{K}) = \{A \subseteq \{1, \dots, t\} \mid \bigcap_{i \in A} K_i \neq \emptyset\}$. Then the simplicial complex $\Delta(\mathcal{K})$ is m -Leray.

Fix $m \geq 0$, and let $f = (f_0, \dots, f_{d-1})$ be the f -vector of a simplicial complex Δ . Define

$$h_k^* = \begin{cases} f_k & \text{for } 0 \leq k \leq m-1 \\ \sum_{j \geq 0} (-1)^j \binom{k+j-m}{j} f_{k+j} & \text{for } k \geq m. \end{cases}$$

The sequence $h^* = (h_0^*, \dots, h_{d-1}^*)$ is the h^* -vector of Δ . The two vectors f and h^* mutually determine each other.

THEOREM 18.2.5

For $h^* = (h_0^*, h_1^*, \dots) \in \mathbb{Z}^{(\infty)}$ the following are equivalent:

- (i) h^* is the h^* -vector of an m -Leray complex;
- (ii) h^* is the h^* -vector of $\Delta(\mathcal{K})$ for some family \mathcal{K} of convex sets in \mathbb{R}^m ;
- (iii)

$$\begin{cases} h_k^* \geq 0 & \text{for } k \geq 0 \\ \partial_{k+1}(h_k^*) \leq h_{k-1}^* & \text{for } 1 \leq k \leq m-1 \\ \partial_m(h_k^*) \leq h_{k-1}^* - h_k^* & \text{for } k \geq m. \end{cases}$$

COMMENTS

The Euler-Poincaré formula (due to Poincaré 1899) is proved in most books on algebraic topology. Theorem 18.2.1 is from [BK88]. A good general source on Cohen-Macaulay complexes is [Sta96]; Theorems 18.2.2, 18.2.3, and 18.2.4, as well as references to the original sources, can be found there. A generalization of Theorem 18.2.2 to complexes whose k -skeleton is Cohen-Macaulay appears in [Bjö96]. There are several additional results about h -vectors of Cohen-Macaulay complexes. For instance, for complexes with nontrivial automorphism groups, see [Sta96, Section III.8]; for matroid complexes, see [Sta96, Section III.3]; and for Cohen-Macaulay complexes that are r -colorable for $r < d$, see the references mentioned in [Sta96, Section III.4].

Cohen-Macaulay complexes are closely related to certain commutative rings [Sta96], and via this connection such complexes have also been of use in the theory of splines; see [Sta96, Section III.5] and also [Chapter 53](#) of this Handbook.

Theorem 18.2.5 was conjectured by Eckhoff and proved by Kalai [Kal84, Kal86].

18.3 SIMPLICIAL POLYTOPES, SPHERES, AND MANIFOLDS

GLOSSARY

A **triangulated d -ball** is a simplicial complex Δ whose realization $\|\Delta\|$ is homeomorphic to the ball $\{x \in \mathbb{R}^d \mid x_1^2 + \cdots + x_d^2 \leq 1\}$. A **triangulated $(d-1)$ -sphere** is a simplicial complex whose realization is homeomorphic to the sphere $\{x \in \mathbb{R}^d \mid x_1^2 + \cdots + x_d^2 = 1\}$. Equivalently, it is the boundary of a triangulated d -ball. Examples of triangulated $(d-1)$ -spheres are given by the boundary complexes of simplicial d -polytopes.

A **pseudomanifold** is a pure simplicial complex Δ such that

- (i) each face of codimension 1 is contained in precisely two maximal faces; and
- (ii) the dual graph (whose vertices are the maximal faces of Δ and whose edges are the faces of codimension 1) is connected.

An **Eulerian pseudomanifold** is a pseudomanifold Δ such that Δ and the link of each face have the Euler characteristic of a sphere of the corresponding dimension.

A pure $(d-1)$ -dimensional simplicial complex Δ is a **homology manifold** if it is connected and the link of each nonempty face has the Betti numbers of a sphere of the same dimension. It is a **homology sphere** if, in addition, Δ itself has the Betti numbers of a $(d-1)$ -sphere. Examples of homology manifolds are given by triangulations of compact connected topological manifolds, i.e., spaces that are locally Euclidean.

The **cyclic d -polytope with n vertices** $C_d(n)$ is the convex hull of any n points on the moment curve in \mathbb{R}^d . (See [Section 16.1.4](#).)

The following implications hold among these various classes, all of them strict:

$$\text{polytope boundary} \Rightarrow \text{sphere} \Rightarrow \text{homology sphere} \Rightarrow \\ \text{Eulerian pseudomanifold} \Rightarrow \text{pseudomanifold}$$

$$\text{homology sphere} \Rightarrow \text{homology manifold} \Rightarrow \text{pseudomanifold}$$

$$\text{homology sphere} \Rightarrow \text{Cohen-Macaulay complex}$$

PSEUDOMANIFOLDS

The following results give the basic lower and upper bounds on f -vectors of pseudomanifolds.

THEOREM 18.3.1 Lower Bound Theorem

For a $(d-1)$ -dimensional pseudomanifold Δ with n vertices,

$$f_k(\Delta) \geq \begin{cases} \binom{d}{k}n - \binom{d+1}{k+1}k & \text{for } 1 \leq k \leq d-2 \\ (d-1)n - (d-2)(d+1) & \text{for } k = d-1. \end{cases}$$

THEOREM 18.3.2 Upper Bound Theorem

Let Δ be a $(d-1)$ -dimensional homology manifold with n vertices, such that either

- (i) d is even, or
- (ii) $d = 2k + 1$ is odd, and either $\chi(\Delta) = 2$ or $\beta_k \leq 2\beta_{k-1} + 2 \sum_{i=0}^{k-3} \beta_i$.

Then $f_k(\Delta) \leq f_k(C_d(n))$ for $1 \leq k \leq d-1$.

This upper bound theorem applies when the homology manifold is Eulerian (irrespective of dimension); in particular, it applies to all simplicial polytopes and spheres. By the geometric operation of “pulling vertices” (Section 17.2), one can extend this to all convex polytopes.

THEOREM 18.3.3

If P is any convex d -polytope with n vertices, then $f(P) \leq f(C_d(n))$.

The given lower and upper bounds are best possible within the class of simplicial polytope boundaries. The lower bound is attained by the class of stacked polytopes

(Sections 17.4.2 and 20.2). To make the upper bound numerically explicit, we give the formula for the f -vector of a cyclic polytope.

THEOREM 18.3.4

For $d \geq 2$ and $0 \leq k \leq d - 1$, the number of k -faces of the cyclic polytope $C_d(n)$ with n vertices is

$$f_k(C_d(n)) = \frac{n - \delta(n - k - 2)}{n - k - 1} \sum_{j=0}^{\lfloor d/2 \rfloor} \binom{n-1-j}{k+1-j} \binom{n-k-1}{2j-k-1+\delta},$$

where $\delta = d - 2\lfloor d/2 \rfloor$.

In particular,

$$f_{d-1}(C_d(n)) = \binom{n - \lceil \frac{d}{2} \rceil}{\lfloor \frac{d}{2} \rfloor} + \binom{n - \lceil \frac{d+1}{2} \rceil}{\lfloor \frac{d-1}{2} \rfloor},$$

which shows that for fixed d the number of facets is $O(n^{\lfloor d/2 \rfloor})$.

POLYTOPES AND SPHERES

For boundaries of simplicial d -polytopes and, more generally, for Eulerian pseudo-manifolds, we have the following basic relations.

THEOREM 18.3.5 Dehn-Sommerville Equations

For d -dimensional Eulerian pseudomanifolds,

$$h_i = h_{d-i} \quad \text{for all } 0 \leq i \leq d.$$

These equations give a complete description of the linear span of all f -vectors of d -polytopes (equivalently, $(d-1)$ -spheres). (The affine span is defined by including the relation $h_0 = 1$.)

One consequence of the Dehn-Sommerville equations is the following relation between the h -vector of a triangulated ball K and the g -vector of its boundary ∂K .

THEOREM 18.3.6

For a triangulated d -ball K and its boundary $(d-1)$ -sphere ∂K ,

$$g_i(\partial K) = h_i(K) - h_{d+1-i}(K) \quad \text{for } i \geq 1.$$

A complete characterization of the f -vectors of simplicial (and, by duality, simple) convex polytopes is given in terms of the h -vector and g -vector.

THEOREM 18.3.7 g -Theorem

A nonnegative integer vector $h = (h_0, \dots, h_d)$ is the h -vector of a simplicial convex d -polytope if and only if

- (i) $h_i = h_{d-i}$, and
- (ii) $(g_0, \dots, g_{\lfloor d/2 \rfloor})$ is an M -sequence.

One consequence of (ii) is that $g_i \geq 0$. For centrally symmetric polytopes, we get a better lower bound.

THEOREM 18.3.8

For centrally symmetric simplicial d -polytopes,

$$g_i = h_i - h_{i-1} \geq \binom{d}{i} - \binom{d}{i-1} \quad \text{for } i \leq \lfloor d/2 \rfloor.$$

The following arithmetic property of the numbers of k -faces of all simplicial d -polytopes is a consequence of the g -theorem.

THEOREM 18.3.9

Given $0 \leq k < d$ there exist positive integers $G(k, d)$ and $N(k, d)$ such that

- (i) $G(k, d)$ divides $f_k(P)$ for every simplicial d -polytope P , and
- (ii) if $G(k, d)$ divides n and $n > N(k, d)$, then $n = f_k(P)$ for some simplicial d -polytope P .

COMMENTS

The Lower Bound Theorem 18.3.1 is due to Kalai and Gromov in the generality given here; see [Kal87] including the note added in proof. The $k = d - 1$ case had earlier been done by Klee and the case of polytope boundaries by Barnette. See [Kal87] for a discussion of the history of this result.

The Upper Bound Theorem 18.3.2 is due to Novik [Nov98]. The case of polytopes (Theorem 18.3.3) was first proved by McMullen (see [MS71]), and extended to spheres by Stanley (see [Sta96]). The computation of the f -vector of the cyclic polytope can be found in [Grü67, Sections 4.7.3 and 9.6.1] or [MS71].

The Dehn-Sommerville equations for polytopes are classical; proofs can be found in [Grü67, Sta86, Zie95]. The extension to Eulerian pseudomanifolds is due to Klee [Kle64]; an equivariant version appears in [Bar92]. The D-S equations imply an upper bound on the average number of j -faces contained in a k -face of a simple polytope (roughly, the number of j -faces of a k -dimensional cube) due to Nikulin. This has been useful in the theory of hyperbolic reflection groups. See [Nik87, Theorem C] for references and ramifications; see also Theorem 18.5.16, which is a similar result for arrangements and zonotopes.

The g -theorem was conjectured by McMullen and proved by Billera, Lee, and Stanley [BL81, Sta80]. More recently, another proof of the necessity of these conditions was given by McMullen [McM93]. It is not known whether the second condition of Theorem 18.3.7 holds for general triangulated spheres. The g -theorem has a convenient reformulation as a one-to-one correspondence (via matrix multiplication) between f -vectors of simplicial polytopes and M -sequences, see [Bjö87, Zie95]. Theorem 18.3.8 was proved by Stanley [Sta87a], for another proof see [Nov99]. Theorem 18.3.9 is from Björner and Linusson [BL99], where also an explicit expression for the modulus $G(k, d)$ is given.

The question of characterizing f -vectors for compact manifolds more general than spheres is at the present far beyond our reach. However, much interesting

work has been done on the more restrictive question of minimizing the number of vertices of triangulations for given manifolds, see e.g. [Küh90, Küh95, BL00, Lu02]. This is of interest for efficient presentations of manifolds to computers.

The study of *f*-vectors of unbounded polyhedra can be approached by studying the *f*-vectors of ***polytope pairs*** (P, F) , where P is a polytope and F is a maximal face of P . See [BL93] for a summary of such results.

18.4 CELL COMPLEXES

GLOSSARY

Convex polytopes and *faces* of such are defined in [Chapter 16](#).

A ***polyhedral complex*** Γ is a finite collection of convex polytopes in \mathbb{R}^n such that (i) if $\pi \in \Gamma$ and σ is a face of π , then $\sigma \in \Gamma$; and (ii) if $\pi, \sigma \in \Gamma$ and $\pi \cap \sigma \neq \emptyset$, then $\pi \cap \sigma$ is a face of both. The ***space*** of Γ is $\|\Gamma\| = \bigcup_{\pi \in \Gamma} \pi$, a subspace of \mathbb{R}^n . Examples of polyhedral complexes are given by ***boundary complexes*** ∂P of convex polytopes P (i.e., the collection of all proper faces). A geometric simplicial complex (defined in Section 18.1) is a polyhedral complex all of whose cells are simplices. A ***cubical complex*** is a polyhedral complex all of whose cells are (combinatorially isomorphic to) cubes.

A ***regular cell complex*** Γ is a family of closed balls (homeomorphs of $\{x \in \mathbb{R}^j \mid |x| \leq 1\}$) in a Hausdorff space $\|\Gamma\|$ such that (i) the interiors of the balls partition $\|\Gamma\|$ and (ii) the boundary of each ball in Γ is a union of other balls in Γ . The members of Γ are called (closed) ***cells*** or ***faces***. The ***dimension*** of a cell is its topological dimension and $\dim \Gamma = \max_{\sigma \in \Gamma} \dim \sigma$.

A regular cell complex has the ***intersection property*** if, whenever the intersection of two cells is nonempty, then this intersection is also a cell in the complex. Polyhedral complexes are examples of regular cell complexes with the intersection property. Regular cell complexes with the intersection property can be reconstructed up to homeomorphism from the corresponding “abstract” complex consisting of the family of vertex sets of its cells.

For a regular cell complex Γ , let f_i be the number of i -dimensional cells, and let $\beta_i = \dim_{\mathbb{Q}} \tilde{H}_i(\|\Gamma\|, \mathbb{Q})$. The latter denotes i -dimensional reduced singular homology with rational coefficients of the space $\|\Gamma\|$; see [Mun84, Spa66] for explanations of this concept. Then we have the ***f-vector*** $f = (f_0, f_1, \dots)$ and the ***Betti sequence*** $\beta = (\beta_0, \beta_1, \dots)$ of Γ . These definitions generalize those previously given in the simplicial case.

BASIC *f*-VECTOR RELATIONS

Among the classes of complexes

- simplicial complexes
- polyhedral complexes

- regular cell complexes with the intersection property
- regular cell complexes

each is a proper subclass of its successor. Thus one may wonder how many of the relations for f -vectors of simplicial complexes given in Sections 18.1–18.3 can be extended to these broader classes of complexes. Also, what new phenomena (not visible in the simplicial case) arise? Some answers are given in this section and the following one, but current knowledge is quite fragmentary. We begin here with the most general relations.

THEOREM 18.4.1

(f_0, \dots, f_d) is the f -vector of a d -dimensional regular cell complex if and only if $f_d \geq 1$ and $f_i \geq 2$ for all $0 \leq i < d$.

THEOREM 18.4.2

f is the f -vector of a regular cell complex with the intersection property if and only if f is a K -sequence.

Let $\beta = (\beta_0, \beta_1, \dots) \in \mathbb{N}^{(\infty)}$ be fixed, and for every sequence $f = (f_0, f_1, \dots)$ let

$$\chi_{k-1} = \sum_{j \geq k} (-1)^{j-k} (f_j - \beta_j) \quad \text{for } k \geq 0.$$

THEOREM 18.4.3

(f_0, \dots, f_d) is the f -vector of a d -dimensional regular cell complex with Betti sequence β if and only if $\chi_{-1} = 1$ and $\chi_k \geq 1$ for $0 \leq k < d$.

THEOREM 18.4.4

For $f \in \mathbb{N}^{(\infty)}$ the following are equivalent:

- f is the f -vector of a regular cell complex with the intersection property and with Betti sequence β ;
- $\chi_{-1} = 1$ and $\partial_{k+1}(\chi_k + \beta_k) \leq \chi_{k-1}$ for all $k \geq 1$.

These results show that the f -vectors of regular cell complexes (with or without Betti number constraints) are considerably more general than the f -vectors of simplicial complexes, but that the two classes of f -vectors agree in the presence of the intersection property.

COMMENTS

Regular cell complexes are known as **regular CW complexes** in the topological literature [LW69]. The nonregular CW complexes offer an even more general class of cell complexes [LW69, Mun84, Spa66], but there is very little one can say about f -vectors in that generality. See [BLS⁺93, Section 4.7] for a detailed discussion of regular cell complexes from a combinatorial point of view.

For the results of this section see [BK88, BK91, BK89]. A characterization of f -vectors of (cubical) subcomplexes of a cube can be found in [Lin71], and of regular cell decompositions of spheres in [Bay88].

18.5 GENERAL POLYTOPES AND SPHERES

GLOSSARY

A **flag** of faces in a (polyhedral) $(d-1)$ -complex Δ is a chain $F_1 \subsetneq F_2 \subsetneq \cdots \subsetneq F_k$ of faces F_i in Δ . It is an **S -flag** if

$$S = \{\dim F_1, \dots, \dim F_k\} \subseteq \{0, 1, \dots, d-1\}.$$

Let $f_S = f_S(\Delta)$ denote the number of S -flags in Δ . The function $S \mapsto f_S$, $S \subseteq \{0, 1, \dots, d-1\}$, is called the **flag f -vector** of Δ . If

$$h_S = \sum_{T \subseteq S} (-1)^{|S|-|T|} f_T,$$

then the function $S \mapsto h_S$, $S \subseteq \{0, 1, \dots, d-1\}$, is called the **flag h -vector**.

For $S \subseteq \{0, \dots, d-1\}$ and noncommuting symbols \mathbf{a} and \mathbf{b} , let $u_S = u_0 u_1 \cdots u_{d-1}$ be the \mathbf{ab} -word defined by $u_i = \mathbf{a}$ if $i \notin S$ and $u_i = \mathbf{b}$ otherwise. When Δ is spherical (or, more generally, Eulerian), then the \mathbf{ab} -polynomial $\sum h_S u_S$ is also a polynomial in $\mathbf{c} = \mathbf{a} + \mathbf{b}$ and $\mathbf{d} = \mathbf{ab} + \mathbf{ba}$. (Note that the degree of \mathbf{c} is 1 and the degree of \mathbf{d} is 2.) The resulting \mathbf{cd} -polynomial

$$\sum h_S u_S = \sum \phi_w w,$$

where the right-hand sum is over all \mathbf{cd} -words w of degree d , is called the **cd -index** $\Phi(\Delta)$ of Δ . For 2-, 3-, and 4-polytopes, the \mathbf{cd} -index is $\mathbf{c}^2 + (f_0 - 2)\mathbf{d}$, $\mathbf{c}^3 + (f_0 - 2)\mathbf{dc} + (f_2 - 2)\mathbf{cd}$, and $\mathbf{c}^4 + (f_0 - 2)\mathbf{dc}^2 + (f_1 - f_0)\mathbf{cdc} + (f_3 - 2)\mathbf{c}^2\mathbf{d} + (f_{02} - 2f_2 - 2f_0 + 4)\mathbf{d}^2$, respectively.

For any convex d -polytope P , we define the **toric h -vector** and **toric g -vector** recursively by $h(P, x) = \sum_{i=0}^d h_i x^{d-i}$ and $g(P, x) = \sum_{i=0}^{\lfloor d/2 \rfloor} g_i x^i$, where $g_i = h_i - h_{i-1}$ and the following relations hold:

- (i) $g(\emptyset, x) = h(\emptyset, x) = 1$; and
- (ii) $h(P, x) = \sum_{G \text{ face of } P, G \neq P} g(G, x)(x-1)^{d-1-\dim G}$.

(Compare to Section 17.4.1, where this toric h -vector is defined for any polyhedral complex. In the notation given there, we have defined h and g for the complex ∂P .) When P is simplicial, this definition coincides with that of the usual h -vector, as defined in Section 18.2. For 2-, 3-, and 4-polytopes, the g -polynomial is $1 + (f_0 - 3)x$, $1 + (f_0 - 4)x$, and $1 + (f_0 - 5)x + (10 - 3f_0 - 3f_3 + f_{03})x^2$, respectively.

A **rational polytope** is one whose vertices all have rational coordinates. Equivalently, all maximal faces are determined by linear forms with rational coefficients.

A **cubical polytope** is one that has a cubical boundary complex. For any cubical $(d-1)$ -complex with f -vector (f_0, \dots, f_{d-1}) , define the **cubical h -vector** $h^c = (h_0^c, \dots, h_d^c)$ by

$$h_i^c = (-1)^i 2^{d-1} + \sum_{j=1}^i (-1)^{i-j} 2^{j-1} f_{j-1} \sum_{k=0}^{i-j} \binom{d-j}{k} \quad \text{for } i = 0, \dots, d.$$

The **cubical g -vector** $g^c = (g_0^c, \dots, g_{\lfloor d/2 \rfloor}^c)$ is defined by $g_0^c = h_0^c = 2^{d-1}$ and $g_i^c = h_i^c - h_{i-1}^c$ for $i \geq 1$.

An **Eulerian polyhedral complex** is one whose first barycentric subdivision is an Eulerian pseudomanifold. Examples are boundary complexes of polytopes and **spherical** polyhedral complexes, i.e., those whose underlying space is homeomorphic to a sphere.

A (central) **hyperplane arrangement** is a collection \mathcal{H} of n linear hyperplanes in \mathbb{R}^d , given by normal vectors x_1, \dots, x_n (see [Section 6.1.3](#)). The arrangement is **essential** if the normals x_i span \mathbb{R}^d . The associated **zonotope** is the Minkowski sum of the n line segments $[-x_i, x_i]$, i.e., $Z = \{\sum \lambda_i x_i \mid -1 \leq \lambda_i \leq 1\}$ (see [Section 16.1.4](#)).

LINEAR RELATIONS

We give the linear equalities on the invariants defined above that are known to hold for all boundary complexes of polytopes and, more generally, for all Eulerian polyhedral complexes.

THEOREM 18.5.1

For $(d-1)$ -dimensional Eulerian polyhedral complexes, the following relations always hold for the flag h , the toric h , and the flag f :

- (i) $h_S = h_{\{0, \dots, d-1\} \setminus S}$ for all $S \subseteq \{0, \dots, d-1\}$;
- (ii) $h_i = h_{d-i}$ for $0 \leq i \leq d$; and
- (iii) $\sum_{j=i+1}^{k-1} (-1)^{j-i-1} f_{S \cup \{j\}} = (1 - (-1)^{k-i-1}) f_S$ whenever $i, k \in S \cup \{-1, d\}$ with $i \leq k-2$ and $S \cap \{i+1, \dots, k-1\} = \emptyset$.

It is known that the relations in Theorem 18.5.1(iii), the **generalized Dehn-Sommerville equations**, completely describe the linear span of all flag f -vectors of Eulerian complexes, and so they imply those in (i). Since the toric h is known to be a linear function of the flag f , they imply those in (ii) as well. The linear span of flag f -vectors has dimension e_d , where e_d is the d th Fibonacci number (defined by the recurrence $e_d = e_{d-1} + e_{d-2}$, $e_0 = e_1 = 1$). There are e_d **cd**-words of degree d . Furthermore, the coefficients ϕ_w of the **cd**-index, considered as linear expressions in the f_S , form a linear basis for the span of flag f -vectors of d -polytopes. The affine span of all flag f -vectors is defined by including the relation $f_\emptyset = 1$.

For cubical polytopes and spheres, the cubical h -vector satisfies the analogue of the Dehn-Sommerville equations.

THEOREM 18.5.2

For cubical d -polytopes and cubical $(d-1)$ -spheres,

$$h_i^c = h_{d-i}^c \quad \text{for all } 0 \leq i \leq d.$$

These give all linear relations satisfied by f -vectors of cubical polytopes and spheres. The cubical h -vector satisfies, as well, the equations of Theorem 18.3.6, linking the h of a cubical ball to the g of its boundary sphere.

LINEAR INEQUALITIES

Some linear inequalities that hold for flag f -vectors of all polytope boundaries are given in this section. The list is not thought to be complete, although there are no conjectures for what the complete set might be.

For a Cohen-Macaulay polyhedral complex, i.e., one whose first barycentric subdivision is a Cohen-Macaulay simplicial complex, the flag h is always nonnegative.

THEOREM 18.5.3

For a Cohen-Macaulay polyhedral $(d-1)$ -complex Γ , we have $h_S(\Gamma) \geq 0$ for all $S \subseteq \{0, \dots, d-1\}$.

For general convex polytopes, we also have nonnegativity of the \mathbf{cd} -index. In fact, the \mathbf{cd} -index of any d -polytope is minimized termwise by the \mathbf{cd} -index of the d -simplex $\Delta^{(d)}$.

THEOREM 18.5.4

For a convex d -polytope P ,

$$\phi_w(P) \geq \phi_w(\Delta^{(d)}) \geq 0$$

for all \mathbf{cd} -words w of degree d .

There are also relations between the \mathbf{cd} -coefficients ϕ_w for any polytope.

THEOREM 18.5.5

For any d -polytope P

$$\phi_{udv}(P) \geq \phi_{uc^2v}(P),$$

for any \mathbf{cd} -words u and v with $\deg u + \deg v = d-2$.

For rational convex polytopes, it is known, further, that the toric h is unimodal.

THEOREM 18.5.6

For a rational¹ convex d -polytope, $g_i \geq 0$ for $i \leq \lfloor d/2 \rfloor$.

Related to this is the following *nonlinear* inequality holding between the g -vectors of a polytope P and any of its faces F . We denote by P/F the *link* of F in P , i.e., the polytope whose lattice of faces is (isomorphic to) the interval $[F, P]$ in the face lattice of P .

THEOREM 18.5.7

For a rational¹ polytope P and any face F , we have the polynomial inequality

$$g(P, t) - g(F, t)g(P/F, t) \geq 0,$$

i.e., all coefficients of this polynomial are nonnegative.

We have a similar relation between the \mathbf{cd} -index of a polytope and that of any face.

THEOREM 18.5.8

For any polytope P and any face F , we have the polynomial inequalities

$$\Phi(P) \geq \begin{cases} \mathbf{c} \cdot \Phi(F) \cdot \Phi(P/F) \\ \Phi(F) \cdot \mathbf{c} \cdot \Phi(P/F) \\ \Phi(F) \cdot \Phi(P/F) \cdot \mathbf{c} \end{cases}$$

¹Note (added in January 2003): As a consequence of recent work by K. Karu (Hard Lefschetz theorem for nonrational polytopes, preprint, December 2002, arXiv:math AG/0112087), it appears that the word “rational” can be removed in Theorems 18.5.6 and 18.5.7.

where $\Phi(P)$, $\Phi(F)$, and $\Phi(P/F)$ are the **cd**-indices of P , F , and P/F , respectively.

As with f -vectors of polytopes, their flag f -vectors, flag h -vectors and **cd**-indices satisfy the upper bound theorem.

THEOREM 18.5.9

If P is a d -dimensional polytope with n vertices, then for any S ,

$$\begin{aligned} f_S(P) &\leq f_S(C_d(n)) \\ h_S(P) &\leq h_S(C_d(n)) \end{aligned}$$

and termwise as polynomials

$$\Phi(P) \leq \Phi(C_d(n))$$

where $C_d(n)$ is the cyclic d -polytope with n vertices.

Finally, we have the following lower bounds for the number of vertices of polytopes with no triangular faces (this includes the class of cubical polytopes), and for the combined numbers of vertices and facets of centrally symmetric polytopes.

THEOREM 18.5.10

A d -polytope with no triangular 2-face has at least 2^d vertices.

THEOREM 18.5.11

There exists a constant $c > 0$ such that

$$\log f_0 \cdot \log f_{d-1} > cd,$$

for any centrally symmetric d -polytope.

HYPERPLANE ARRANGEMENTS AND ZONOTOPES

An essential hyperplane arrangement \mathcal{H} defines a decomposition of \mathbb{R}^d into polyhedral cones (as in Section 6.1.3). This decomposition $\Gamma_{\mathcal{H}}$, a regular cell complex if intersected with the unit sphere, has a flag f -vector dual to that of its associated zonotope Z , in the sense that $f_S(\Gamma_{\mathcal{H}}) = f_{d-S}(Z)$, where $S = \{i_1, \dots, i_k\} \subseteq \{1, \dots, d\}$ and $d - S = \{d - i_k, \dots, d - i_1\}$.

THEOREM 18.5.12

The flag f -vector of an arrangement (or zonotope) depends only on the matroid (linear dependency structure) of the underlying point configuration $\{x_1, \dots, x_n\}$.

Although a fairly special subclass of polytopes, the zonotopes nonetheless are varied enough to carry all the linear information carried by flag numbers of general polytopes.

THEOREM 18.5.13

The flag f -vectors of zonotopes (and thus of hyperplane arrangements) satisfy the generalized Dehn-Sommerville equations, and there are no other linear relations not implied by these.

When it comes to linear *inequalities*, however, a difference between zonotopes and general polytopes emerges. As with general convex polytopes, we have non-negativity of the \mathbf{cd} -index for zonotopes. However, the \mathbf{cd} -index of any d -zonotope is minimized termwise by the \mathbf{cd} -index of the d -cube $C^{(d)}$.

THEOREM 18.5.14

For a convex d -zonotope Z , $\phi_w(Z) \geq \phi_w(C^{(d)}) \geq 0$ for all \mathbf{cd} -words w of degree d . Further, if the word w has k \mathbf{d} 's, then 2^k divides $\phi_w(Z)$.

There is also a strengthening of Theorem 18.5.5 for zonotopes.

THEOREM 18.5.15

For any d -zonotope Z

$$\phi_{udv}(Z) - \phi_{uc^2v}(Z) \geq \phi_{udv}(C^{(d)}) - \phi_{uc^2v}(C^{(d)})$$

for any \mathbf{cd} -words u and v with $\deg u + \deg v = d - 2$.

The following result has the most direct interpretation when it is stated for arrangements, where it bounds the average number of $\{i_1, \dots, i_k\}$ -flags in an i_k -face by the number of $\{i_1-1, \dots, i_k-1\}$ -flags in an (i_k-1) -cube.

THEOREM 18.5.16

For a hyperplane arrangement \mathcal{H} in \mathbb{R}^d and $S = \{i_1, \dots, i_k\} \subseteq \{1, \dots, d\}$ with $k \geq 2$,

$$\frac{f_S(\Gamma_{\mathcal{H}})}{f_{i_k}(\Gamma_{\mathcal{H}})} < \binom{i_k - 1}{i_1 - 1, i_2 - i_1, \dots, i_k - i_{k-1}} 2^{i_k - i_1}.$$

There is a straightforward reformulation of Theorem 18.5.16 for zonotopes that is easily seen not to be valid for all polytopes.

GENERAL 3- AND 4-POLYTOPES

We describe here the situation for flag f -vectors of 3- and 4-polytopes. The equations in Theorem 18.5.1(iii) reduce consideration to (f_0, f_2) when $d = 3$ and to (f_0, f_1, f_2, f_{02}) when $d = 4$.

THEOREM 18.5.17

For 3-polytopes, the following is known about the vector (f_0, f_2) .

- (i) An integer vector (f_0, f_2) is the f -vector of a 3-polytope if and only if $f_0 \leq 2f_2 - 4$ and $f_2 \leq 2f_0 - 4$.
- (ii) An integer vector (f_0, f_2) is the f -vector of a cubical 3-polytope if and only if $f_2 = f_0 - 2$, $f_0 \geq 8$, and $f_0 \neq 9$.
- (iii) If $(f_0, f_2) = (f_0(Z), f_2(Z))$ for a 3-zonotope Z , then f_0 and f_1 are both even integers, $f_0 \leq 2f_2 - 4$, and $f_2 \leq f_0 - 2$.

For 4-polytopes, much less is known.

THEOREM 18.5.18

Flag f -vectors (f_0, f_1, f_2, f_{02}) of 4-polytopes satisfy the following inequalities.

- (i) $f_{02} \geq 3f_2$
- (ii) $f_{02} \geq 3f_1$
- (iii) $f_{02} + f_1 + 10 \geq 3f_2 + 4f_0$
- (iv) $6f_1 \geq 6f_0 + f_{02}$
- (v) $f_0 \geq 5$
- (vi) $f_0 + f_2 \geq f_1 + 5$
- (vii) $2(f_{02} - 3f_2) \leq \binom{f_0}{2}$
- (viii) $2(f_{02} - 3f_1) \leq \binom{f_2 - f_1 + f_0}{2}$
- (ix) $f_{02} - 4f_2 + 3f_1 - 2f_0 \leq \binom{f_0}{2}$
- (x) $f_{02} + f_2 - 2f_1 - 2f_0 \leq \binom{f_2 - f_1 + f_0}{2}.$

It is not known, for example, whether (i)–(vi) give all linear inequalities holding for flag f -vectors of 4-polytopes.

COMMENTS

It is thought that the best route to an eventual characterization of f -vectors of general polytopes lies in an understanding of their flag f -vectors. The latter inherit many of the algebraic properties of f -vectors of simplicial polytopes that led to their characterization, while having a rich theory of their own.

The relations in Theorem 18.5.1 hold more generally for the case of enumeration of chains in Eulerian posets; see the article by Stanley in [BMSW94]. The relations in Theorem 18.5.1(iii) are proved in [BB85]. An expression for the toric h in terms of the flag f can be found in the article by Bayer in [BMSW94]. The article by Kalai in the same volume contains an extensive discussion of g -vectors for both simplicial and general polytopes. Expressions for the (toric) g and h -vectors in terms of the flag h -vector or the **cd**-index can be found in [BE00]. The form of the cubical Dehn-Sommerville equations given in Theorem 18.5.2 appeared in [Adi96].

Theorem 18.5.3 can be found in [Sta96, Theorem III.4.4] (where h_S is denoted $\beta(S)$). The nonnegativity of the **cd**-index for polytopes in Theorem 18.5.4 holds as well for certain shellable spheres and is due to Stanley (see [Sta96, Section III.4]). That the **cd**-index is minimized over polytopes by simplices is shown in [BE00], while Theorem 18.5.5 is proved in [Ehr01]. Theorem 18.5.6 appears in [Sta87b]. Relationships between these classes of inequalities and those that can be derived from them are discussed in [Ste01b]. Nonnegativity of *certain* **cd**-coefficients for all spheres is shown in [Bay01] and [Rea02], and for odd-dimensional simplicial manifolds in [Nov00].

The problem of determining all linear inequalities for flag f -vectors has been considered for classes of partially ordered sets more general than the face posets of polytopes and spheres. In [BH00a], the (Catalan many) extreme rays are determined for the closed convex cone determined by flag f -vectors of all graded posets (posets with a rank function and having minimum and maximum elements). A nice description of the finite minimum set of inequalities is lacking, however. In [BH01], a partial family of extreme rays is determined for the subcone determined by all Eulerian posets. See [BH00b] for more such results.

There is a notion of convolution product of flag f numbers, originally due to Kalai [Kal88], that can be used to produce new linear inequalities from given ones;

see, for example, [BL93, Section 3.10]. The algebraic properties of this product have been developed in [BL00]; this has led to a deeper understanding of the combinatorial and algebraic properties of the \mathbf{cd} -index via duality of Hopf algebras (see [BHvW03]).

Theorem 18.5.7, due to Braden and MacPherson [BM99], gives a connection between the g -vector of a polytope P and that of one of its faces. The analogous Theorem 18.5.8 for \mathbf{cd} -indices can be found in [BE00] (as can the upper bound theorem, 18.5.9). These are examples of “monotonicity theorems” related to face numbers. For similar theorems relating h -vectors of subcomplexes and subdivisions of a simplicial complex Δ , see Sections III.9–10 of [Sta96] and the references given there.

Theorems 18.5.10 and 18.5.11 are due to Blind and Blind [BB90] and Figiel, Lindenstrauss, and Milman [FLM77], respectively.

For the fact that the flag f -vector of a zonotope or arrangement (or, more generally, of an oriented matroid) depends only on the underlying matroid, see [BLS⁺93, Cor. 4.6.3]. For expressions giving the \mathbf{cd} -index of a zonotope in terms of the flag h -vector of its underlying geometric lattice, see [BER97, Corollary 3.2] and [BHvW03, Proposition 3.5]. That the only linear relations satisfied by zonotopes are the generalized Dehn-Sommerville equations of Theorem 18.5.1(iii), as well as the divisibility property in Theorem 18.5.14, is proved in [BER98]. The bounds on the \mathbf{cd} -indices of zonotopes in Theorem 18.5.14 are proved in [BER97]; the bounds in Theorem 18.5.15 can be found in [Ehr01]. Theorem 18.5.16 is due to Varchenko for the case $k = 2$ (see [BLS⁺93, Proposition 4.6.9]) and to Liu. The stronger version given here is due to Stenson; in fact, [Ste02, Theorem 9] gives a stronger inequality (see also [Ste01a]).

Theorem 18.5.17(i) can be found in [Grü67, Section 10.3]; 18.5.17(ii) appears in dual form (for 4-valent 3-polytopes) in [Bar83]; 18.5.17(iii) can be derived using the methods of [Grü67, Section 18.2] (see also [BER98]). Theorem 18.5.18 can be found in [Bay87]; see also [HZ00]. An interesting general discussion of f -vectors of 4-polytopes (ordinary and flag) and an up-to-date survey of this topic is given by Ziegler [Zie02]. In particular, a good case is made there that the situation for f -vectors of 4-polytopes is much more complicated than that for polytopes in dimension 3. One reason for this is that *neighborly cubical* d -polytopes begin to exist for $d = 4$: for any $n \geq d \geq 2r + 2$, there is a cubical convex d -polytope whose r -skeleton is combinatorially equivalent to that of the n -dimensional cube [JZ00] (see also [BBC97], where spheres having this property are constructed). In particular, for any $n \geq 4$, there is a cubical 4-polytope with the graph of the n -cube. These polytopes show that the ratio f_3/f_0 is not bounded over cubical 4-polytopes.

18.6 OPEN PROBLEMS

PROBLEM 18.6.1

Characterize the f -vectors of triangulations of the $(d-1)$ -sphere.

[It has been conjectured that the conditions of the g -theorem provide the answer.]

PROBLEM 18.6.2

Characterize the f -vectors of triangulations of the d -ball.

PROBLEM 18.6.3

Characterize the f -vectors of triangulations of the d -torus.

[It is known that $f(2\text{-torus}) = \{(n, 3n, 2n) \mid n \geq 7\}$, but the question is open for $d \geq 3$.]

PROBLEM 18.6.4

Characterize the f -vectors of d -polytopes.

[The answer is known for $d \leq 3$ (Theorem 18.5.17(i)), but for $d \geq 4$ there is not even a conjectured answer.]

PROBLEM 18.6.5 I. Bárány

Does there exist a constant $c_d > 0$ such that $f_i \geq c_d \cdot \min\{f_0, f_{d-1}\}$ for all d -polytopes and all i ? Will $c_d = 1$ do?

PROBLEM 18.6.6

Characterize the f -vectors of centrally symmetric d -polytopes.

[The question is open in the simplicial as well as in the general case. Even an upper bound conjecture in the simplicial and centrally symmetric case is missing.]

PROBLEM 18.6.7 Conjecture of G. Kalai

The total number of faces (counting P but not \emptyset) of a centrally symmetric convex d -polytope P is $\geq 3^d$.

[Verified in the simplicial case as a consequence of Theorem 18.3.8.]

PROBLEM 18.6.8

The clique complex of a graph is the collection of vertex sets of all its cliques (complete induced subgraphs). Characterize the f -vectors of clique complexes.

PROBLEM 18.6.9 J. Eckhoff and G. Kalai

Is the f -vector of any $(r-1)$ -dimensional clique complex the f -vector of some r -colorable complex?

PROBLEM 18.6.10 Conjecture of Charney and Davis [Sta96, p. 100]

Let (g_0, \dots, g_k) be the g -vector of a clique complex homeomorphic to the sphere S^{2k-1} . Then $g_k - g_{k-1} + \dots + (-1)^k g_0 \geq 0$.

PROBLEM 18.6.11 Conjecture of Stanley [Sta96, p. 102]

*Every coefficient ϕ_w of the **cd**-index of a spherical regular cell complex is nonnegative.*

[This conjecture, if true, gives the most general possible linear inequalities for flag f -vectors of spherical regular cell complexes (i.e., regular cell complexes homeomorphic to the sphere).]

[For simplicial spheres, the **cd**-coefficients satisfy the conclusion of Theorem 18.5.4.]

PROBLEM 18.6.12 Conjecture of Ehrenborg [Ehr01, Conj. 5.1]

*For d -polytopes P (and more generally for simplicial $(d-1)$ -spheres) the **cd**-index*

satisfies

$$\phi_{udv}(P) - \phi_{uc^2v}(P) \geq \phi_{udv}(\Delta^{(d)}) - \phi_{uc^2v}(\Delta^{(d)}),$$

where $\deg u + \deg v = d - 2$, and $\Delta^{(d)}$ is the d -simplex.

PROBLEM 18.6.13 *Adin* [Adi96]

The “generalized lower bound conjecture” for cubical d -polytopes and $(d-1)$ -spheres: $g_i^c \geq 0$ for $i \leq \lfloor d/2 \rfloor$.

[This has been shown to be the best possible set of linear inequalities for cubical $(d-1)$ -spheres [BBC97]. The case $i = 1$ is implied by Theorem 18.5.10.]

More generally, characterize the f -vectors of cubical polytopes.

PROBLEM 18.6.14

Characterize the flag f -vectors of polytopes and of zonotopes. In particular, determine a complete set of linear inequalities holding for flag f -vectors of polytopes and of zonotopes.

PROBLEM 18.6.15

Characterize (toric) h -vectors of general polytopes.

PROBLEM 18.6.16

Characterize flag f -vectors of colored complexes (here f_S is the number of simplices with color set S); of pure colored complexes; of graded posets [all linear inequalities are known here [BH00a]]; of Eulerian posets [see [BH01]]; of Eulerian lattices.

18.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Surveys of f -vector theory are given in [BL93, Bjö87, BK89, KK95, Sta85]. Books treating f -vectors (among other things) include [And87, BMSW94, Grü67, MS71, Sta96, Zie95].

RELATED CHAPTERS

[Chapter 6: Oriented matroids](#)

[Chapter 16: Basic properties of convex polytopes](#)

[Chapter 17: Subdivisions and triangulations of polytopes](#)

[Chapter 53: Splines and geometric modeling](#)

REFERENCES

[Adi96] R.M. Adin. A new cubical h -vector. *Discrete Math.*, 157:3–14, 1996.

[And87] I. Anderson. *Combinatorics of Finite Sets*. Clarendon Press, Oxford, 1987.

- [BBC97] E.K. Babson, L.J. Billera, and C. Chan. Neighborly cubical spheres and a cubical lower bound conjecture. *Israel J. Math.*, 102:297–315, 1997.
- [Bar83] D.W. Barnette. *Map Coloring, Polyhedra, and the Four Color Theorem*. Number 8 of *Dolciani Math. Exp.*, Math. Assoc. America, Washington, 1983.
- [Bar92] A.I. Barvinok. On equivariant generalization of Dehn-Sommerville equations. *European J. Combin.*, 13:419–428, 1992.
- [Bay87] M.M. Bayer. The extended f-vectors of 4-polytopes. *J. Combin. Theory. Ser. A*, 44:141–151, 1987.
- [Bay88] M.M. Bayer. Barycentric subdivisions. *Pacific J. Math.*, 135:1–16, 1988.
- [Bay01] M.M. Bayer. Signs in the \mathbf{cd} -index of Eulerian partially ordered sets. *Proc. Amer. Math. Soc.*, 129:2219–2226, 2001.
- [BB85] M.M. Bayer and L.J. Billera. Generalized Dehn-Sommerville relations for polytopes, spheres and Eulerian partially ordered sets. *Invent. Math.*, 79:143–157, 1985.
- [BE00] M.M. Bayer and R. Ehrenborg. The toric h -vector of partially ordered sets. *Trans. Amer. Math. Soc.*, 352:4515–4531, 2000.
- [BH01] M.M. Bayer and G. Hetyei. Flag vectors of Eulerian partially ordered sets. *European J. Combin.*, 22:5–26, 2001.
- [BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. North-Holland, Amsterdam, 1993.
- [BE00] L.J. Billera and R. Ehrenborg. Monotonicity of the \mathbf{cd} -index for polytopes. *Math. Z.*, 233:421–441, 2000.
- [BER97] L.J. Billera, R. Ehrenborg, and M. Readdy. The $\mathbf{c}\text{-}2\mathbf{d}$ -index of oriented matroids. *J. Combin. Theory Ser. A*, 80:79–105, 1997.
- [BER98] L.J. Billera, R. Ehrenborg, and M. Readdy. The \mathbf{cd} -index of zonotopes and arrangements. In B. Sagan and R. Stanley, editors, *Mathematical Essays in Honor of Gian-Carlo Rota*, Birkhäuser, Boston, 1998.
- [BH00a] L.J. Billera and G. Hetyei. Linear inequalities for flags in graded posets. *J. Combin. Theory Ser. A*, 89:77–104, 2000.
- [BH00b] L.J. Billera and G. Hetyei. Decompositions of partially ordered sets. *Order*, 17:141–166, 2000.
- [BHvW03] L.J. Billera, S.K. Hsiao, and S. van Willigenburg. Peak quasisymmetric functions and Eulerian enumeration. *Adv. Math.*, 176:248–276, 2003.
- [BL81] L.J. Billera and C.W. Lee. A proof of the sufficiency of McMullen’s conditions for f -vectors of simplicial polytopes. *J. Combin. Theory Ser. A*, 31:237–255, 1981.
- [BL00] L.J. Billera and N. Liu. Noncommutative enumeration in graded posets. *J. Algebraic Combin.*, 12:7–24, 2000.
- [BMSW94] T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors. *Polytopes: Abstract, Convex, and Computational*. Volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.* Kluwer, Dordrecht, 1994.
- [Bjö87] A. Björner. Face numbers of complexes and polytopes. In *Proc. Internat. Cong. Math., Berkeley, 1986*, pages 1408–1418. Amer. Math. Soc., Providence, 1987.
- [Bjö96] A. Björner. Nonpure shellability, f -vectors, subspace arrangements, and complexity. In L.J. Billera, C. Greene, R. Simion, and R. Stanley, editors, *Formal Power Series and Algebraic Combinatorics, DIMACS Ser. in Discrete Math. and Theor. Comput. Sci.*, pages 25–53. Amer. Math. Soc., Providence, 1996.
- [BK88] A. Björner and G. Kalai. An extended Euler-Poincaré theorem. *Acta Math.*, 161:279–303, 1988.

- [BK89] A. Björner and G. Kalai. On f -vectors and homology. In G. Bloom, R.L. Graham, and J. Malkevitch, editors, *Combinatorial Mathematics: Proc. 3rd Internat. Conf., New York, 1985*, volume 555 of *Ann. New York Acad. Sci.*, pages 63–80. New York Acad. Sci., 1989.
- [BK91] A. Björner and G. Kalai. Extended Euler-Poincaré relations for cell complexes. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, pages 81–89, volume 4 of *DIMACS Series in Discrete Math. and Theor. Comput. Sci.*, Amer. Math. Soc., Providence, 1991.
- [BLS⁺93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*. Volume 46 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993. Second edition, 1999.
- [BL99] A. Björner and S. Linusson. The number of k -faces of a simple d -polytope. *Discrete Comput. Geom.*, 21:1–16, 1999.
- [BL00] A. Björner and F.H. Lutz. Simplicial manifolds, bistellar flips and a 16-vertex triangulation of the Poincaré homology 3-sphere. *Experiment. Math.*, 9:275–289, 2000.
- [BB90] G. Blind and R. Blind. Convex polytopes without triangular faces. *Israel J. Math.*, 71:129–134, 1990.
- [BM99] T.C. Braden and R. MacPherson. Intersection homology of toric varieties and a conjecture of Kalai. *Comment. Math. Helv.*, 74:442–455, 1999.
- [Ehr01] R. Ehrenborg. Inequalities for polytopes and zonotopes. Preprint, 2001.
- [FLM77] T. Figiel, J. Lindenstrauss, and V.D. Milman. The dimension of almost spherical sections of convex bodies. *Acta Math.*, 139:53–94, 1977.
- [FFK88] P. Frankl, Z. Füredi, and G. Kalai. Shadows of colored complexes. *Math. Scand.*, 63:169–178, 1988.
- [FK96] E. Friedgut and G. Kalai. Every monotone graph property has a sharp threshold. *Proc. Amer. Math. Soc.*, 124:2993–3002, 1996.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Interscience, London, 1967. Revised edition (V. Kaibel, V. Klee, and G.M. Ziegler, editors), Volume 221 of *Grad. Texts in Math.*, Springer-Verlag, New York, 2003.
- [Hib89] T. Hibi. What can be said about pure O-sequences? *J. Combin. Theory Ser. A*, 50:319–322, 1989.
- [HZ00] A. Höppner and G.M. Ziegler. A census of flag-vectors of 4-polytopes. In G. Kalai and G.M. Ziegler, editors, *Polytopes—Combinatorics and Computation*, volume 29 of DMV Sem., pages 105–110, Birkhäuser-Verlag, Basel, 2000.
- [JZ00] M. Joswig and G.M. Ziegler. Neighborly Cubical Polytopes. *Discrete Comput. Geom.*, 24:325–344, 2000.
- [Kal84] G. Kalai. A characterization of f -vectors of families of convex sets in \mathbb{R}^d . Part I: Necessity of Eckhoff's conditions. *Israel J. Math.*, 48:175–195, 1984.
- [Kal86] G. Kalai. A characterization of f -vectors of families of convex sets in \mathbb{R}^d . Part II: Sufficiency of Eckhoff's conditions. *J. Combin. Theory Ser. A*, 41:167–188, 1986.
- [Kal87] G. Kalai. Rigidity and the lower bound theorem I. *Invent. Math.*, 88:125–151, 1987.
- [Kal88] G. Kalai. A new basis of polytopes. *J. Comb. Theory Ser. A*, 49:191–208, 1988.
- [Kle64] V. Klee. A combinatorial analogue of Poincaré's duality theorem. *Canad. J. Math.*, 16:517–531, 1964.
- [KK95] V. Klee and P. Kleinschmidt. Convex polytopes and related complexes. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 875–917. North-Holland, Amsterdam, 1995.
- [Küh90] W. Kühnel. Triangulations of manifolds with few vertices. In F. Tricerri, editor, *Advances in Differential Geometry and Topology*, pages 59–114. World Scientific, Singapore, 1990.

- [Küh95] W. Kühnel. *Tight Polyhedral Submanifolds and Tight Triangulations*. Volume 1612 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1995.
- [Lin71] B. Lindström. The optimal number of faces in cubical complexes. *Ark. Mat.*, 8:245–257, 1971.
- [LW69] A.T. Lundell and S. Weingram. *The Topology of CW Complexes*. Van Nostrand, New York, 1969.
- [Lu02] F. Lutz. Triangulated manifolds with few vertices. Springer-Verlag, Berlin, in preparation.
- [McM93] P. McMullen. On simple polytopes. *Invent. Math.*, 113:419–444, 1993.
- [MS71] P. McMullen and G.C. Shephard. *Convex Polytopes and the Upper Bound Conjecture*. Volume 3 of *London Math. Soc. Lecture Note Ser.*, Cambridge University Press, 1971.
- [Mun84] J.R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Reading, 1984.
- [Nik87] V.V. Nikulin. Discrete reflection groups in Lobachevsky spaces and algebraic surfaces. In *Proc. Internat. Cong. Math., Berkeley, 1986*, pages 654–671. Amer. Math. Soc., Providence, 1987.
- [Nov98] I. Novik. Upper bound theorems for homology manifolds. *Israel J. Math.*, 108:45–82, 1998.
- [Nov99] I. Novik. The lower bound theorem for centrally symmetric simple polytopes. *Mathematika*, 46:231–240, 1999.
- [Nov00] I. Novik. Lower bounds for the cd -index of odd-dimensional simplicial manifolds. *European J. Combin.*, 21:533–541, 2000.
- [Rea02] N. Reading. *On the Structure of Bruhat Order*. Ph.D. Thesis, University of Minnesota, Minneapolis, 2002.
- [Spa66] E.H. Spanier. *Algebraic Topology*. McGraw-Hill, New York, 1966.
- [Sta80] R.P. Stanley. The number of faces of simplicial convex polytopes. *Adv. Math.*, 35:236–238, 1980.
- [Sta85] R.P. Stanley. The number of faces of simplicial polytopes and spheres. In J.E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, editors, *Discrete Geometry and Convexity*, volume 440 of *Ann. New York Acad. Sci.*, pages 212–223. New York Acad. Sci., 1985.
- [Sta86] R.P. Stanley. *Enumerative Combinatorics*, Volume I. Wadsworth, Monterey, 1986. Second printing by Cambridge Univ. Press, 1997.
- [Sta87a] R.P. Stanley. On the number of faces of centrally-symmetric simplicial polytopes. *Graphs Combin.*, 3:55–66, 1987.
- [Sta87b] R.P. Stanley. Generalized h -vectors, intersection cohomology of toric varieties, and related results. In M. Nagata and H. Matsumura, editors, *Commutative Algebra and Combinatorics*, volume 11 of *Adv. Stud. Pure Math.*, pages 187–213. Kinokuniya, Tokyo and North-Holland, Amsterdam, 1987.
- [Sta96] R.P. Stanley. *Combinatorics and Commutative Algebra*, 2nd Ed. Volume 41 of *Progr. Math.*, Birkhäuser, Boston, 1996.
- [Ste01a] C. Stenson. *Linear Inequalities for Flag f -vectors of Polytopes*. Ph.D. Thesis, Cornell Univ., Ithaca, 2001.
- [Ste01b] C. Stenson. Relationships among flag f -vector inequalities for polytopes. *Discrete Comput. Geom.*, to appear.
- [Ste02] C. Stenson. Tight inequalities for polytopes. Preprint, 2002.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995. Revised edition, 1998.
- [Zie02] G.M. Ziegler. Face numbers of 4-polytopes and 3-spheres. In *Proc. Internat. Cong. Math., Beijing, 2002*, pages 625–634. Higher Ed. Press, Beijing, 2002.

19 SYMMETRY OF POLYTOPES AND POLYHEDRA

Egon Schulte

INTRODUCTION

Symmetry of geometric figures is among the most frequently recurring themes in science. The present chapter discusses symmetry of discrete geometric structures, namely of polytopes, polyhedra, and related polytope-like figures. These structures have an outstanding history of study unmatched by almost any other geometric object. The most prominent symmetric figures, the regular solids, occur from very early times and are attributed to Plato (427-347 B.C.E.). Since then, many changes in point of view have occurred about these figures and their symmetry. With the arrival of group theory in the 19th century, many of the early approaches were consolidated and the foundations were laid for a more rigorous development of the theory. In this vein, Schläfli (1814-1895) extended the concept of regular polytopes and tessellations to higher dimensional spaces and explored their symmetry groups as reflection groups.

Today we owe much of our present understanding of symmetry in geometric figures (in a broad sense) to the influential work of Coxeter, which provided a unified approach to regularity of figures based on a powerful interplay of geometry and algebra [Cox73]. Coxeter's work also greatly influenced modern developments in this area, which received a further impetus from work by Grünbaum and Danzer [Grü77a, DS82]. In the past 25 years, the study of regular figures has been extended in several directions that are all centered around an abstract combinatorial polytope theory and a combinatorial notion of regularity [MS02].

History teaches us that the subject has shown an enormous potential for revival. One explanation for this is the appearance of polyhedral structures in many contexts that have little apparent relation to regularity, such as the occurrence of many of them in nature as crystals [Fej64, Sen95, Wel77].

19.1 REGULAR CONVEX POLYTOPES AND REGULAR TESSELLATIONS IN \mathbb{E}^d

Perhaps the most important (but certainly the most investigated) symmetric polytopes are the regular convex polytopes in Euclidean spaces. See [Grü67] and [Zie95] for general properties of convex polytopes, or [Chapter 16](#) in this Handbook. The most comprehensive text on regular convex polytopes and regular tessellations is [Cox73]; many combinatorial aspects are also discussed in [MS02].

GLOSSARY

Convex d-polytope: The intersection P of finitely many closed halfspaces in a

Euclidean space, which is bounded and d -dimensional.

Face: The empty set and P itself are *improper faces* of dimension -1 and d , respectively. A *proper face* F of P is the (nonempty) intersection of P with a supporting hyperplane of P . (Recall that a hyperplane H *supports* P at F if $P \cap H = F$ and P lies in one of the closed halfspaces bounded by H .)

Vertex, edge, i -face, facet: Face of P of dimension $0, 1, i$, or $d-1$, respectively.

Vertex figure: A vertex figure of P at a vertex x is the intersection of P with a hyperplane H that strictly separates x from the other vertices of P . (If P is regular, one can take H to be the hyperplane passing through the midpoints of the edges that contain x .)

Face lattice of a polytope: The set $\mathcal{F}(P)$ of all faces of P , ordered by inclusion. As a partially ordered set, this is a ranked lattice. Also, $\mathcal{F}(P) \setminus \{P\}$ is called the *boundary complex* of P .

Flag: A maximal totally ordered subset of $\mathcal{F}(P)$.

Isomorphism of polytopes: A bijection $\varphi : \mathcal{F}(P) \rightarrow \mathcal{F}(Q)$ between the face lattices of two polytopes P and Q such that φ preserves incidence in both directions; that is, $F \subseteq G$ in $\mathcal{F}(P)$ if and only if $F\varphi \subseteq G\varphi$ in $\mathcal{F}(Q)$. If such an isomorphism exists, P and Q are *isomorphic*.

Dual of a polytope: A convex d -polytope Q is the dual of P if there is a *duality* $\varphi : \mathcal{F}(P) \rightarrow \mathcal{F}(Q)$; that is, a bijection reversing incidences in both directions, meaning that $F \subseteq G$ in $\mathcal{F}(P)$ if and only if $F\varphi \supseteq G\varphi$ in $\mathcal{F}(Q)$. A polytope has many duals but any two are isomorphic, justifying speaking of “the dual.” (If P is regular, one can take Q to be the convex hull of the facet centers of P , or a rescaled copy of this.)

Self-dual polytope: A polytope that is isomorphic to its dual.

Symmetry: A Euclidean isometry of the ambient space (affine hull of P) that maps P to itself.

Symmetry group of a polytope: The group $G(P)$ of all symmetries of P .

Regular polytope: A polytope whose symmetry group $G(P)$ is transitive on the flags.

Schl  fli symbol: A symbol $\{p_1, \dots, p_{d-1}\}$ that encodes the local structure of a regular polytope. For each $i = 1, \dots, d-1$, if F is any $(i+1)$ -face of P , then p_i is the number of i -faces of F that contain a given $(i-2)$ -face of F .

Tessellation: A family T of convex d -polytopes in Euclidean d -space \mathbb{E}^d , called the *tiles* of T , such that the union of all tiles of T is \mathbb{E}^d , and any two distinct tiles do not have interior points in common. All tessellations are assumed to be *locally finite*, meaning that each point of \mathbb{E}^d has a neighborhood meeting only finitely many tiles, and *face-to-face*, meaning that the intersection of any two tiles is a face of each (possibly the empty face); see [Chapter 3](#). The concept of a tessellation extends to other spaces including spherical space (Euclidean unit sphere) and hyperbolic space.

Face lattice of a tessellation: A *proper face* of T is a nonempty face of a tile of T . *Improper faces* of T are the empty set and the whole space \mathbb{E}^d . The set $\mathcal{F}(T)$ of all (proper and improper) faces is a ranked lattice called the face lattice of T . Concepts such as isomorphism and duality carry over from polytopes.

Symmetry group of a tessellation: The group $G(T)$ of all symmetries of T ; that is, of all isometries of the ambient (spherical, Euclidean, or hyperbolic) space that preserve T . Concepts such as regularity and Schläfli symbol carry over from polytopes.

Apeirogon: A tessellation of the real line with closed intervals of the same length. This can also be regarded as an infinite polygon whose edges are given by the intervals.

ENUMERATION AND CONSTRUCTION

The convex regular polytopes P in \mathbb{E}^d are known for each d . If $d = 1$, P is a line segment and $|G(P)| = 2$. In all other cases, up to similarity, P can be uniquely described by its Schläfli symbol $\{p_1, \dots, p_{d-1}\}$. For convenience one writes $P = \{p_1, \dots, p_{d-1}\}$. If $d = 2$, P is a convex regular p -gon for some $p \geq 3$, and $P = \{p\}$; also, $G(P) = D_p$, the dihedral group of order $2p$.

The regular polytopes P with $d \geq 3$ are summarized in Table 19.1.1, which also includes the numbers f_0 and f_{d-1} of vertices and facets, the order of $G(P)$, and the diagram notation (Section 19.6) for the group (following [Hum90]). Here and below, p^n will be used to denote a string of n consecutive p 's. For $d = 3$ the list consists of the five Platonic solids (Figure 19.1.1). The regular d -simplex, d -cube, and d -cross-polytope occur in each dimension d . (These are line segments if $d = 1$, and triangles or squares if $d = 2$.) The dimensions 3 and 4 are exceptional in that there are 2 (respectively 3) more regular polytopes. If $d \geq 3$, the facets and vertex figures of $\{p_1, \dots, p_{d-1}\}$ are the regular $(d-1)$ -polytopes $\{p_1, \dots, p_{d-2}\}$ and $\{p_2, \dots, p_{d-1}\}$, respectively, whose Schläfli symbols, when superposed, give the original. The dual of $\{p_1, \dots, p_{d-1}\}$ is $\{p_{d-1}, \dots, p_1\}$. Self-duality occurs only for $\{3^{d-1}\}$, $\{p\}$, and $\{3, 4, 3\}$. Except for $\{3^{d-1}\}$ and $\{p\}$ with p odd, all regular polytopes are centrally symmetric.

TABLE 19.1.1 The convex regular polytopes in \mathbb{E}^d ($d \geq 3$).

DIMENSION	NAME	SCHLÄFLI SYMBOL	f_0	f_{d-1}	$ G(P) $	DIAGRAM
$d \geq 3$	d -simplex	$\{3^{d-1}\}$	$d+1$	$d+1$	$(d+1)!$	A_d
	d -cross-polytope	$\{3^{d-2}, 4\}$	$2d$	2^d	$2^d d!$	B_d (or C_d)
	d -cube	$\{4, 3^{d-2}\}$	2^d	$2d$	$2^d d!$	B_d (or C_d)
$d = 3$	icosahedron	$\{3, 5\}$	12	20	120	H_3
	dodecahedron	$\{5, 3\}$	20	12	120	H_3
$d = 4$	24-cell	$\{3, 4, 3\}$	24	24	1152	F_4
	600-cell	$\{3, 3, 5\}$	120	600	14400	H_4
	120-cell	$\{5, 3, 3\}$	600	120	14400	H_4

The regular tessellations T in \mathbb{E}^d are also known. If $d = 1$, T is an apeirogon and $G(T)$ is the infinite dihedral group. For $d \geq 2$ see the list in Table 19.1.2. The first $d - 1$ entries in $\{p_1, \dots, p_d\}$ give the Schläfli symbol for the (regular) tiles of T , the last $d - 1$ that for the (regular) vertex figures. (A vertex figure at a vertex x is the convex hull of the midpoints of the edges emanating from x .) The cubical

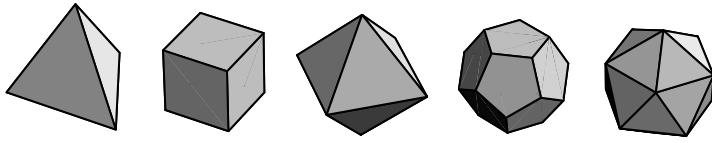


FIGURE 19.1.1

The five Platonic solids.

Tetrahedron

Cube

Octahedron

Dodecahedron

Icosahedron

tessellation occurs for each d , while for $d = 2$ and $d = 4$ there is a dual pair of exceptional tessellations.

TABLE 19.1.2 The regular tessellations in \mathbb{E}^d ($d \geq 2$).

DIMENSION	SCHLÄFLI SYMBOL	TILES	VERTEX-FIGURES
$d \geq 2$	$\{4, 3^{d-2}, 4\}$	d -cubes	d -cross-polytopes
$d = 2$	$\{3, 6\}$ $\{6, 3\}$	triangles hexagons	hexagons triangles
$d = 4$	$\{3, 3, 4, 3\}$ $\{3, 4, 3, 3\}$	4-cross-polytopes 24-cells	24-cells 4-cross-polytopes

As vertices of the plane polygon $\{p\}$ we can take the points corresponding to the p th roots of unity. The d -simplex can be defined as the convex hull of the $d+1$ points in \mathbb{E}^{d+1} corresponding to the permutations of $(1, 0, \dots, 0)$. As vertices of the d -cross-polytope in \mathbb{E}^d choose the $2d$ permutations of $(\pm 1, 0, \dots, 0)$, and for the d -cube take the 2^d points $(\pm 1, \dots, \pm 1)$. The midpoints of the edges of a 4-cross-polytope are the 24 vertices of a regular 24-cell. The coordinates for the remaining regular polytopes are more complicated [Cox73, pp. 52,157].

For the cubical tessellation $\{4, 3^{d-2}, 4\}$ take the vertex set to be \mathbb{Z}^d (giving the square tessellation if $d = 2$). For the triangle tessellation $\{3, 6\}$ choose as vertices the integral linear combinations of two unit vectors inclined at $\pi/3$. Locating the face centers gives the vertices of the hexagonal tessellation $\{6, 3\}$. For $\{3, 3, 4, 3\}$ in \mathbb{E}^4 take the alternating vertices of the cubical tessellation; that is, the integral points with an even coordinate sum. Its dual $\{3, 4, 3, 3\}$ (with 24-cells as tiles) has the vertices at the centers of the tiles of $\{3, 3, 4, 3\}$.

The regular polytopes and tessellations have been with us since before recorded history, and a strong strain of mathematics since classical times has centered on them. The classical theory intersects with diverse mathematical areas such as Lie algebras and Lie groups, Tits buildings [Tit74], finite and combinatorial group theory [Bue95, Mag74], geometric and algebraic combinatorics, graphs and combinatorial designs [BCN89], singularity theory, and Riemann surfaces.

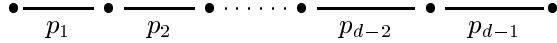
SYMMETRY GROUPS

For a convex regular d -polytope P in \mathbb{E}^d , pick a fixed (**base**) flag Φ , and consider the maximal simplex C (**chamber**) in the barycentric subdivision (**chamber complex**) of P whose vertices are the centers of the nonempty faces in Φ . Then

C is a fundamental region for $G(P)$ in P and $G(P)$ is generated by the reflections R_0, \dots, R_{d-1} in the walls of C that contain the center of P , where R_i is the reflection in the wall opposite to the vertex of C corresponding to the i -face in Φ . If $P = \{p_1, \dots, p_{d-1}\}$, then

$$\begin{cases} R_i^2 = (R_j R_k)^2 = 1 & (0 \leq i, j, k \leq d-1, |j-k| \geq 2) \\ (R_{i-1} R_i)^{p_i} = 1 & (1 \leq i \leq d-1) \end{cases}$$

is a presentation for $G(P)$ in terms of these generators. In particular, $G(P)$ is a finite (spherical) Coxeter group with string diagram



(see [Section 19.6](#)).

If T is a regular tessellation of \mathbb{E}^d , pick Φ and C as before. Now $G(T)$ is generated by the $d+1$ reflections in all walls of C giving R_0, \dots, R_d (as above). The presentation for $G(T)$ carries over, but now $G(T)$ is an infinite (Euclidean) Coxeter group.

19.2 REGULAR STAR-POLYTOPES

The regular star-polyhedra and star-polytopes are obtained by allowing the faces or vertex figures to be *starry* (star-like). This leads to very beautiful figures that are closely related to the regular convex polytopes. See Coxeter [Cox73] for a comprehensive account; see also McMullen and Schulte [MS02]. In defining star-polytopes, we shall combine the approach of [Cox73] and McMullen [McM68] and introduce them via the associated starry polytope-configuration.

GLOSSARY

***d*-polytope-configuration:** A finite family Π of affine subspaces, called *elements*, of Euclidean d -space \mathbb{E}^d , ordered by inclusion, such that the following conditions are satisfied. Π contains the empty set \emptyset and \mathbb{E}^d as (*improper*) elements. The dimensions of the other (*proper*) elements can take the values $0, 1, \dots, d-1$, and the affine hull of their union is \mathbb{E}^d . As a partially ordered set, Π is a ranked lattice. For $F, G \in \Pi$ with $F \subseteq G$ call $G/F := \{H \in \Pi | F \subseteq H \subseteq G\}$ the *subconfiguration* of Π defined by F and G ; this has itself the structure of a $(\dim(G) - \dim(F) - 1)$ -polytope-configuration. As further conditions, each G/F contains at least 2 proper elements if $\dim(G) - \dim(F) = 2$, and as a partially ordered set, each G/F (including Π itself) is connected if $\dim(G) - \dim(F) \geq 3$. (See the definition of an abstract polytope in [Section 19.8](#).) It can be proved that in \mathbb{E}^d every Π satisfies the stronger condition that each G/F contains exactly 2 proper elements if $\dim(G) - \dim(F) = 2$.

Regular polytope-configuration: A polytope-configuration Π whose symmetry group $G(\Pi)$ is flag-transitive. (A flag is a maximal totally ordered subset of Π .)

Regular star-polygon: For positive integers n and k with $(n, k) = 1$ and $1 < k < \frac{n}{2}$, up to similarity the regular star-polygon $\{\frac{n}{k}\}$ is the connected plane

polygon whose consecutive vertices are $(\cos(\frac{2\pi k j}{n}), \sin(\frac{2\pi k j}{n}))$ for $j = 0, 1, \dots, n-1$. If $k = 1$, the same plane polygon bounds a (nonstarry) convex n -gon with Schläfli symbol $\{n\}$ ($= \{\frac{n}{1}\}$). With each regular (convex or star-) polygon $\{\frac{n}{k}\}$ is associated a regular 2-polytope-configuration obtained by replacing each edge by its affine hull.

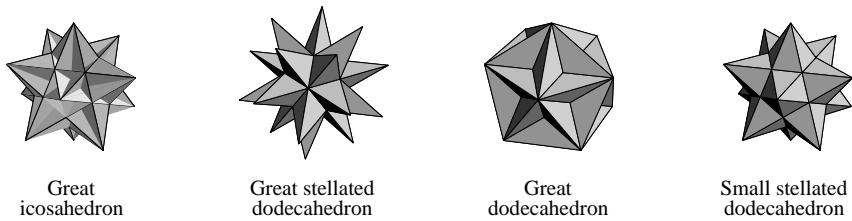
Star-polytope-configuration: A d -polytope-configuration Π is **nonstarry** if it is the family of affine hulls of the faces of a convex d -polytope. It is **starry**, or a **star-polytope-configuration**, if it is not nonstarry. For instance, among the 2-polytope-configurations that are associated with a regular (convex or star-) polygon $\{\frac{n}{k}\}$ for a given n , the one with $k = 1$ is nonstarry and those for $k > 1$ are starry. In the first case the corresponding n -gon is convex, and in the second case it is genuinely star-like. In general, the starry polytope configurations are those that belong to genuinely star-like polytopes (that is, star-polytopes).

Regular star-polytope: If $d = 2$, a regular star-polytope is a regular star-polygon. Defined inductively, if $d \geq 3$, a regular d -star-polytope P is a finite family of regular convex $(d-1)$ -polytopes or regular $(d-1)$ -star-polytopes such that the family consisting of their affine hulls as well as the affine hulls of their “faces” is a regular d -star-polytope-configuration $\Pi = \Pi(P)$. Here, the faces of the polytopes can be defined in such a way that they correspond to the elements in the associated polytope-configuration. The symmetry groups of P and Π are the same.

ENUMERATION AND CONSTRUCTION

Regular star-polytopes P can only exist for $d = 2, 3$, or 4 . As regular convex polytopes, they are also uniquely determined by the Schläfli symbol $\{p_1, \dots, p_{d-1}\}$, but now at least one entry is not integral. Again the symbols for the facets and vertex figures, when superposed, give the original. If $d = 2$, $P = \{\frac{n}{k}\}$ for some k with $(n, k) = 1$ and $1 < k < \frac{n}{2}$, and $G(P) = D_n$. For $d = 3$ and 4 the star-polytopes are listed in [Table 19.2.1](#) together with the numbers f_0 and f_{d-1} of vertices and facets, respectively.

FIGURE 19.2.1
The four
Kepler-Poinsot
polyhedra.



Every regular d -star-polytope has the same vertices and symmetry group as a regular convex d -polytope. The four regular star-polyhedra (3-star-polytopes) are also known as the **Kepler-Poinsot polyhedra** (Figure 19.2.1). They can be constructed from the icosahedron $\{3, 5\}$ or dodecahedron $\{5, 3\}$ by two kinds of operations, **stellation** or **faceting** [Cox73]. Loosely speaking, in the former operation one extends the faces of a polyhedron symmetrically until they again form a polyhedron, while in the latter operation the vertices of a polyhedron are redis-

 TABLE 19.2.1 The regular star-polytopes in \mathbb{E}^d ($d \geq 3$).

DIMENSION	SCHLÄFLI SYMBOL	f_0	f_{d-1}
$d = 3$	$\{3, \frac{5}{2}\}$	12	20
	$\{\frac{5}{2}, 3\}$	20	12
	$\{5, \frac{5}{2}\}$	12	12
	$\{\frac{5}{2}, 5\}$	12	12
$d = 4$	$\{3, 3, \frac{5}{2}\}$	120	600
	$\{\frac{5}{2}, 3, 3\}$	600	120
	$\{3, 5, \frac{5}{2}\}$	120	120
	$\{\frac{5}{2}, 5, 3\}$	120	120
	$\{3, \frac{5}{2}, 5\}$	120	120
	$\{\frac{5}{2}, \frac{5}{2}, 3\}$	120	120
	$\{5, 3, \frac{5}{2}\}$	120	120
	$\{\frac{5}{2}, 3, 5\}$	120	120
	$\{5, \frac{5}{2}, 5\}$	120	120
	$\{\frac{5}{2}, \frac{5}{2}, \frac{5}{2}\}$	120	120

tributed in classes that are then the vertex sets for the faces of a new polyhedron. Regarded as regular maps on surfaces (Section 19.3), the polyhedra $\{3, \frac{5}{2}\}$ (*great icosahedron*) and $\{\frac{5}{2}, 3\}$ (*great stellated dodecahedron*) are of genus 0, while $\{5, \frac{5}{2}\}$ (*great dodecahedron*) and $\{\frac{5}{2}, 5\}$ (*small stellated dodecahedron*) are of genus 4.

The ten regular star-polytopes in \mathbb{E}^4 all have the same vertices and symmetry groups as the 600-cell $\{3, 3, 5\}$ or 120-cell $\{5, 3, 3\}$ and can be derived from these by 4-dimensional stellation or faceting operations [Cox73, McM68]. See also [Cox93] for their names, which describe the various relationships among the polytopes. For presentations of their symmetry groups that reflect the finer combinatorial structure of the star-polytopes, see also [MS02].

The dual of $\{p_1, \dots, p_{d-1}\}$ (which is obtained by dualizing the associated star-polytope-configuration using reciprocation with respect to a sphere) is $\{p_{d-1}, \dots, p_1\}$. Regarded as abstract polytopes (Section 19.8), the star-polytopes $\{p_1, \dots, p_{d-1}\}$ and $\{q_1, \dots, q_{d-1}\}$ are isomorphic if and only if the symbol $\{q_1, \dots, q_{d-1}\}$ is obtained from $\{p_1, \dots, p_{d-1}\}$ by replacing each entry 5 by $\frac{5}{2}$ and each $\frac{5}{2}$ by 5.

19.3 REGULAR SKEW POLYHEDRA

Regular skew polyhedra are finite or infinite polyhedra whose vertex figures are skew (antiprismatic) polygons. The standard reference is Coxeter [Cox68]. Topologically, these polyhedra are regular maps on surfaces. For general properties of regular maps see Coxeter and Moser [CM80], McMullen and Schulte [MS02], or [Chapter 21](#) of this Handbook.

GLOSSARY

(Right) prism, antiprism (with regular bases): A convex 3-polytope whose

vertices are contained in two parallel planes and whose set of 2-faces consists of the two **bases** (contained in the parallel planes) and the 2-faces in the **mantle** that connects the bases. The bases are congruent regular polygons. For a (right) prism, each base is a translate of the other by a vector perpendicular to its affine hull, and the mantle 2-faces are rectangles. For a (right) antiprism, each base is a translate of a reciprocal (dual) of the other by a vector perpendicular to its affine hull, and the mantle 2-faces are isosceles triangles. (The prism or antiprism is **semiregular** if its mantle 2-faces are squares or equilateral triangles, respectively; see [Section 19.5](#).)

Map on a surface: A decomposition (tessellation) P of a closed surface S into nonoverlapping simply connected regions, the **2-faces** of P , by arcs, the **edges** of P , joining pairs of points, the **vertices** of P , such that two conditions are satisfied. First, each edge belongs to exactly two 2-faces. Second, if two distinct edges intersect, they meet in one vertex or in two vertices.

Regular map: A map P on S whose combinatorial automorphism group $\Gamma(P)$ is transitive on the flags (incident triples consisting of a vertex, an edge, and a 2-face).

Polyhedron: A map P on a closed surface S embedded (without self-intersections) into a Euclidean space, such that two conditions are satisfied. Each 2-face of P is a convex plane polygon, and any two adjacent 2-faces do not lie in the same plane. See also the more general definition in [Section 19.4](#) below.

Skew polyhedron: A polyhedron P such that for at least one vertex x , the vertex figure of P at x is not a plane polygon; the **vertex figure** at x is the polygon whose vertices are the vertices of P adjacent to x and whose edges join consecutive vertices as one goes around x .

Regular polyhedron: A polyhedron P whose symmetry group $G(P)$ is flag-transitive. (For a regular skew polyhedron P in \mathbb{E}^3 or \mathbb{E}^4 , each vertex figure must be a 3-dimensional antiprismatic polygon, meaning that it contains all edges of an antiprism that are not edges of a base. See also [Section 19.4](#).)

ENUMERATION

In \mathbb{E}^3 all, and in \mathbb{E}^4 all finite, regular skew polyhedra are known [Cox68]. In these cases the (orientable) polyhedron P is completely determined by the extended Schläfli symbol $\{p, q|r\}$, where the 2-faces of P are convex p -gons such that q meet at each vertex, and r is the number of edges in each edge path of P that leaves, at each vertex, exactly two 2-faces of P on the right. The group $G(P)$ is isomorphic to $\Gamma(P)$ and has the presentation

$$\rho_0^2 = \rho_1^2 = \rho_2^2 = (\rho_0\rho_1)^p = (\rho_1\rho_2)^q = (\rho_0\rho_2)^2 = (\rho_0\rho_1\rho_2\rho_1)^r = 1$$

(but the generators ρ_i are not all hyperplane reflections). The polyhedra $\{p, q|r\}$ and $\{q, p|r\}$ are duals, and the vertices of one can be obtained as the centers of the 2-faces of the other.

In \mathbb{E}^3 there are just three regular skew polyhedra: $\{4, 6|4\}$, $\{6, 4|4\}$, and $\{6, 6|3\}$. These are the (infinite) **Petrie-Coxeter polyhedra**. For example, $\{4, 6|4\}$ consists of half the square faces of the cubical tessellation $\{4, 3, 4\}$ in \mathbb{E}^3 .

TABLE 19.3.1 The finite regular skew polyhedra in \mathbb{E}^4 .

SCHLÄFLI SYMBOL	f_0	f_2	GROUP ORDER	GENUS
$\{4, 4 r\}$	r^2	r^2	$8r^2$	1
$\{4, 6 3\}$	20	30	240	6
$\{6, 4 3\}$	30	20	240	6
$\{4, 8 3\}$	144	288	2304	73
$\{8, 4 3\}$	288	144	2304	73

The finite regular skew polyhedra in \mathbb{E}^4 (or equivalently, in spherical 3-space) are listed in Table 19.3.1. There is an infinite sequence of toroidal polyhedra as well as two pairs of duals related to the (self-dual) 4-simplex $\{3, 3, 3\}$ and 24-cell $\{3, 4, 3\}$. For drawings of projections of these polyhedra into 3-space see [BW88, SW91]; Figure 19.3.1 represents $\{4, 8|3\}$.

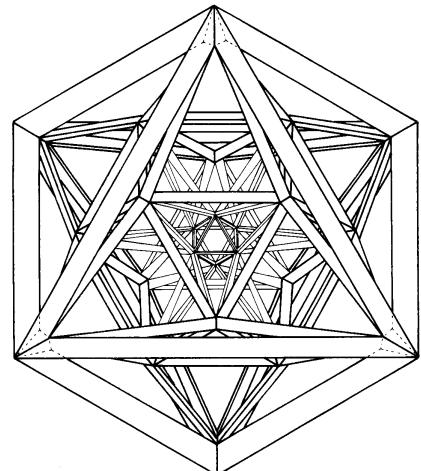


FIGURE 19.3.1
A projection of $\{4, 8|3\}$ into \mathbb{R}^3 .

These projections are examples of **combinatorially regular polyhedra** in ordinary 3-space; see [BW93] and Chapter 21 in this Handbook. For regular polyhedra in \mathbb{E}^4 with planar, but not necessarily convex, 2-faces, see also [ABM00, Bra00]. For regular skew polyhedra in hyperbolic 3-space, see [Gar67].

19.4 THE GRÜNBAUM-DRESS POLYHEDRA

A new impetus to the study of regular figures came from Grünbaum [Grü77b], who generalized the regular skew polyhedra by allowing skew polygons as faces as well as vertex figures. This restored the symmetry in the definition of polyhedra. For the classification of these “new” regular polyhedra in \mathbb{E}^3 , see [Grü77b], [Dre85], and

[MS02]. The proper setting for this subject is, strictly speaking, in the context of realizations of abstract regular polytopes (see [Section 19.8](#)).

GLOSSARY

Polygon: A figure P in Euclidean space \mathbb{E}^d consisting of a (finite or infinite) sequence of distinct points, called the *vertices* of P , joined in successive pairs, and closed cyclicly if finite, by line segments, called the *edges* of P , such that each compact set in \mathbb{E}^d meets only finitely many edges.

Zigzag polygon: A (zigzag-shaped) infinite plane polygon P whose vertices alternately lie on two parallel lines and whose edges are all of the same length.

Antiprismatic polygon: A closed polygon P in 3-space whose vertices are alternately vertices of each of the two (regular convex) bases of a (right) antiprism Q (Section 19.3), such that the orthogonal projection of P onto the plane of a base gives a regular star-polygon (Section 19.2). This star-polygon (and thus P) has twice as many vertices as each base, and is a convex polygon if and only if the edges of P are just those edges of Q that are not edges of a base.

Prismatic polygon: A closed polygon P in 3-space whose vertices are alternately vertices of each of the two (regular convex) bases of a (right) prism Q (Section 19.3), such that the orthogonal projection of P onto the plane of a base traverses twice a regular star-polygon in that plane (Section 19.2). Each base of Q (and thus the star-polygon) is assumed to have an odd number of vertices. The star-polygon is a convex polygon if and only if each edge of P is a diagonal in a rectangular 2-face in the mantle of Q .

Helical polygon: An infinite polygon in 3-space whose vertices lie on a helix given parametrically by $(a \cos \beta t, a \sin \beta t, bt)$, where $a, b \neq 0$ and $0 < \beta < \pi$, and are obtained as t ranges over the integers. Successive integers correspond to successive vertices.

Polyhedron: A (finite or infinite) family P of polygons in \mathbb{E}^d , called the *2-faces* of P , such that three conditions are satisfied. First, each edge of one of the 2-faces is an edge of exactly one other 2-face. Second, for any two edges F and F' of (2-faces of) P there exist chains $F = G_0, G_1, \dots, G_n = F'$ of edges and H_1, \dots, H_n of 2-faces such that each H_i is incident with G_{i-1} and G_i . Third, each compact set in \mathbb{E}^d meets only finitely many 2-faces.

Regular: A polygon or polyhedron P is regular if its symmetry group $G(P)$ is transitive on the flags.

Petrie polygon of a polyhedron: A polygonal path along the edges of a regular polyhedron P such that any two successive edges, but no three, are edges of a 2-face of P .

Petrie dual: The family of all Petrie polygons of a regular polyhedron P . This is itself a regular polyhedron, and its Petrie dual is P itself.

ENUMERATION

For a systematic discussion of regular polygons in arbitrary Euclidean spaces see [Cox93]. In light of the geometric classification scheme for the new regular polyhedra in \mathbb{E}^3 proposed in [Grü77b], it is useful to classify the regular polygons in \mathbb{E}^3

into seven groups: convex polygons, plane star-polygons (Section 19.2), apeirogons (Section 19.1), zigzag polygons, antiprismatic polygons, prismatic polygons, and helical polygons. These correspond to the four kinds of isometries in \mathbb{E}^3 : rotation, rotatory reflection (a reflection followed by a rotation in the reflection plane), glide reflection, and twist.

The 2-faces and vertex figures of a regular polyhedron P in \mathbb{E}^3 are regular polygons of the above kind. (The vertex figure at a vertex x is the polygon whose vertices are the vertices of P adjacent to x and whose edges join two such vertices y and z if and only if $\{y, x\}$ and $\{x, z\}$ are edges of a 2-face in P . For a regular P , this is a single polygon.) It is convenient to group the regular polyhedra in \mathbb{E}^3 into 8 classes. The first four are the traditional regular polyhedra: the five Platonic solids; the three planar tessellations; the four regular star-polyhedra (Kepler-Poinsot polyhedra); and the three infinite regular skew polyhedra (Petrie-Coxeter polyhedra). The four other classes and their polyhedra can be described as follows: the class of nine finite polyhedra with finite skew (antiprismatic) polygons as faces; the class of infinite polyhedra with finite skew (prismatic or antiprismatic) polygons as faces, which includes three infinite families as well as three individual polyhedra; the class of polyhedra with zigzag polygons as faces, which contains six infinite families; and the class of polyhedra with helical polygons as faces, which has three infinite families and six individual polyhedra.

Alternatively, these forty-eight polyhedra can be described as follows [MS02]. There are eighteen finite regular polyhedra, namely the nine classical finite regular polyhedra (Platonic solids and Kepler-Poinsot polyhedra), and their Petrie duals. The regular tessellations of the plane, and their Petrie duals (with zigzag 2-faces), are the six planar polyhedra in the list. From those, twelve further polyhedra are obtained as blends (in the sense of Section 19.8) with a line segment or an apeirogon (Section 19.1). The six blends with a line segment have finite skew, or (infinite planar) zigzag, 2-faces with alternate vertices on a pair of parallel planes; the six blends with an apeirogon have helical polygons or zigzag polygons as 2-faces. Finally, there are twelve further polyhedra that are not blends; they fall into a single family and are related to the cubical tessellation of \mathbb{E}^3 . Each polyhedron can be described by a generalized Schläfli symbol, which encodes the geometric structure of the polygonal faces and vertex figures, tells whether or not the polyhedron is a blend, and indicates a presentation of the symmetry group. For more details see [MS02] (or [Grü77b, Dre85, Joh]).

19.5 SEMIREGULAR AND UNIFORM CONVEX POLYTOPES

The very stringent requirements in the definition of regularity of polytopes can be relaxed in many different ways, yielding a great variety of weaker regularity notions. We shall only consider polytopes and polyhedra that are convex. See Johnson [Joh] for a detailed discussion, or Martini [Mar94] for a survey.

GLOSSARY

Semiregular: A convex d -polytope P is semiregular if its facets are regular and its symmetry group $G(P)$ is transitive on the vertices of P .

Uniform: A convex polygon is uniform if it is regular. Recursively, if $d \geq 3$, a convex d -polytope P is uniform if its facets are uniform and its symmetry group $G(P)$ is transitive on the vertices of P .

Regular-faced: P is regular-faced if all its facets (and lower-dimensional faces) are regular.

ENUMERATION

Each regular polytope is semiregular, and each semiregular polytope is uniform. Also, by definition each uniform 3-polytope is semiregular. For $d = 3$ the family of semiregular (uniform) convex polyhedra consists of the Platonic solids, two infinite classes of prisms and antiprisms, as well as the thirteen polyhedra known as Archimedean solids [Fej64]. The seven semiregular polyhedra whose symmetry group is edge-transitive are also called the *quasiregular* polyhedra.

Besides the regular polytopes, there are only seven semiregular polytopes in higher dimensions: three for $d = 4$, and one for each of $d = 5, 6, 7, 8$ (for a short proof, see [BB91]). However, there are many more uniform polytopes but a complete list is known only for $d = 4$ [Joh]. Except for the regular 4-polytopes and the prisms over uniform 3-polytopes, there are exactly 40 uniform 4-polytopes.

For $d = 3$ all, for $d = 4$ all save one, and for $d \geq 5$ many, uniform polytopes can be obtained by a method called *Wythoff's construction*. This method proceeds from a finite Euclidean reflection group W in \mathbb{E}^d , or the even (rotation) subgroup W^+ of W , and constructs the polytopes as the convex hull of the orbit under W or W^+ of a point, the initial vertex, in the fundamental region of the group, which is a d -simplex (chamber) or the union of two adjacent d -simplices in the corresponding chamber complex of W , respectively; see Sections 19.1 and 19.6.

The regular-faced polytopes have also been described for each dimension. In general, such a polytope can have different kinds of facets (and vertex figures). For $d = 3$ the complete list contains exactly 92 regular-faced convex polyhedra and includes all semiregular polyhedra. For each $d \geq 5$, there are only two regular-faced d -polytopes that are not semiregular. Except for $d = 4$, each regular-faced d -polytope has a nontrivial symmetry group.

There are many further generalizations of the notion of regularity [Mar94]. However, in most cases complete lists of the corresponding polytopes are either not known or available only for $d = 3$. The variants that have been considered include: *isogonal* polytopes (requiring vertex-transitivity of $G(P)$), or *isohedral* polytopes, the reciprocals of the isogonal polytopes, with a facet-transitive group $G(P)$; more generally, *k -face-transitive* polytopes (requiring transitivity of $G(P)$ on the k -faces), for a single value or several values of k ; *congruent-faceted*, or *monohedral*, polytopes (requiring congruence of the facets); and *equifaceted* polytopes (requiring combinatorial isomorphism of the facets). Similar problems have also been considered for nonconvex polytopes or polyhedra, and for tilings [GS87].

19.6 REFLECTION GROUPS

Symmetry properties of geometric figures are closely tied to the algebraic structure

of their symmetry groups, which are often subgroups of finite or infinite reflection groups. A classical reference for reflection groups is Coxeter [Cox73]. A more recent text is Humphreys [Hum90].

GLOSSARY

Reflection group: A group generated by (hyperplane) reflections in a finite-dimensional space V . The space can be a real or complex vector space (or affine space). A **reflection** is a linear (or affine) transformation whose eigenvalues, save one, are all equal to 1, while the remaining eigenvalue is a primitive k th root of unity for some $k \geq 2$; in the real case, it is -1 . If the space is equipped with further structure, the reflections are assumed to preserve it. For example, if V is real Euclidean, the reflections are Euclidean reflections.

Coxeter group: A group W , finite or infinite, that is generated by finitely many generators $\sigma_1, \dots, \sigma_n$ and has a presentation of the form $(\sigma_i \sigma_j)^{m_{ij}} = 1$ ($i, j = 1, \dots, n$), where the m_{ij} are positive integers or ∞ such that $m_{ii} = 1$ and $m_{ij} = m_{ji} \geq 2$ ($i \neq j$). The matrix $(m_{ij})_{ij}$ is the **Coxeter matrix** of W .

Coxeter diagram: A labeled graph \mathcal{D} that represents a Coxeter group W as follows. The nodes of \mathcal{D} represent the generators σ_i of W . The i th and j th node are joined by a (single) branch if and only if $m_{ij} > 2$. In this case, the branch is labeled m_{ij} if $m_{ij} \neq 3$ (and remains unlabeled if $m_{ij} = 3$).

Irreducible Coxeter group: A Coxeter group W whose Coxeter diagram is connected. (Each Coxeter group W is the direct product of irreducible Coxeter groups, with each factor corresponding to a connected component of the diagram of W .)

Root system: A finite set \mathcal{R} of nonzero vectors, the **roots**, in \mathbb{E}^d satisfying the following conditions. \mathcal{R} spans \mathbb{E}^d , and $\mathcal{R} \cap \mathbb{R}e = \{\pm e\}$ for each $e \in \mathcal{R}$. For each $e \in \mathcal{R}$, the Euclidean reflection S_e in the linear hyperplane orthogonal to e maps \mathcal{R} onto itself. Moreover, the numbers $2(e, e')/(e', e')$, with $e, e' \in \mathcal{R}$, are integers (**Cartan integers**); here \langle , \rangle denotes the standard inner product on \mathbb{E}^d . (These conditions define **crystallographic** root systems. Sometimes the integrality condition is omitted to give a more general notion of root system.) The group W generated by the reflections S_e ($e \in \mathcal{R}$) is a finite Coxeter group, called the **Weyl group** of \mathcal{R} .

GENERAL PROPERTIES

Every Coxeter group $W = \langle \sigma_1, \dots, \sigma_n \rangle$ admits a faithful representation as a reflection group in the real vector space \mathbb{R}^n . This is obtained as follows. If W has Coxeter matrix $M = (m_{ij})_{ij}$ and e_1, \dots, e_n is the standard basis of \mathbb{R}^n , define the symmetric bilinear form \langle , \rangle_M by

$$\langle e_i, e_j \rangle_M := -\cos(\pi/m_{ij}) \quad (i, j = 1, \dots, n),$$

with appropriate interpretation if $m_{ij} = \infty$. For $i = 1, \dots, n$ the linear transfor-

tion $S_i : \mathbb{R}^n \mapsto \mathbb{R}^n$ given by

$$xS_i := x - 2\langle e_i, x \rangle_M e_i \quad (x \in \mathbb{R}^n)$$

is the orthogonal reflection in the hyperplane orthogonal to e_i . Let $O(M)$ denote the orthogonal group corresponding to $\langle \cdot, \cdot \rangle_M$. Then $\sigma_i \mapsto S_i$ ($i = 1, \dots, n$) defines a faithful representation $\rho : W \mapsto GL(\mathbb{R}^n)$, called the **canonical representation**, such that $W\rho \subseteq O(M)$.

The group W is finite if and only if the associated form $\langle \cdot, \cdot \rangle_M$ is positive definite; in this case, $\langle \cdot, \cdot \rangle_M$ determines a Euclidean geometry on \mathbb{R}^n . In other words, each finite Coxeter group is a finite Euclidean reflection group. Conversely, every finite Euclidean reflection group is a Coxeter group. The finite Coxeter groups have been completely classified by Coxeter and are usually listed in terms of their Coxeter diagrams.

The finite irreducible Coxeter groups with string diagrams are precisely the symmetry groups of the convex regular polytopes, with a pair of dual polytopes corresponding to a pair of groups that are related by reversing the order of the generators. See [Section 19.1](#) for an explanation about how the generators act on the polytopes. [Table 19.1.1](#) also lists the names for the corresponding Coxeter diagrams.

For $p_1, \dots, p_{n-1} \geq 2$ write $[p_1, \dots, p_{n-1}]$ for the Coxeter group with string diagram $\bullet -_{p_1} \bullet -_{p_2} \bullet - \cdots -_{p_{n-2}} \bullet -_{p_{n-1}} \bullet$. Then $[p_1, \dots, p_{n-1}]$ is the automorphism group of the universal abstract regular n -polytope $\{p_1, \dots, p_{n-1}\}$; see [Section 19.8](#). The regular honeycombs $\{p_1, \dots, p_{n-1}\}$ on the sphere (convex regular polytopes) or in Euclidean or hyperbolic space are examples of such universal polytopes. The spherical honeycombs are exactly the finite universal regular polytopes (with $p_i > 2$ for all i). The Euclidean honeycombs arise exactly when $p_i > 2$ for all i and the bilinear form $\langle \cdot, \cdot \rangle_M$ for $[p_1, \dots, p_{n-1}]$ is positive semidefinite (but not positive definite). Similarly, the hyperbolic honeycombs correspond exactly to the groups $[p_1, \dots, p_{n-1}]$ that are Coxeter groups of “hyperbolic type” [MS02].

There are exactly two sources of finite Coxeter groups, to some extent overlapping: the symmetry groups of convex regular polytopes, and the Weyl groups of (crystallographic) root systems, which are important in Lie Theory. Every root system \mathcal{R} has a set of **simple roots**; this is a subset \mathcal{S} of \mathcal{R} , which is a basis of \mathbb{E}^d such that every $e \in \mathcal{R}$ is a linear combination of vectors in \mathcal{S} with integer coefficients that are all nonnegative or all nonpositive. The distinguished generators of the Weyl group W are given by the reflections S_e in the linear hyperplane orthogonal to e ($e \in \mathcal{S}$), for some set \mathcal{S} of simple roots of \mathcal{R} . The irreducible Weyl groups in \mathbb{E}^2 are the symmetry groups of the triangle, square, or hexagon. The diagrams A_d , B_d , C_d , and F_4 of [Table 19.1.1](#) all correspond to irreducible Weyl groups and root systems (with B_d and C_d corresponding to a pair of dual root systems), but H_3 and H_4 do not (they correspond to a noncrystallographic root system [CMP98]). There is one additional series of irreducible Weyl groups in \mathbb{E}^d with $d \geq 4$ (a certain subgroup of index 2 in B_d), whose diagram is denoted by D_d . The remaining irreducible Weyl groups occur in dimensions 6, 7, and 8, with diagrams E_6 , E_7 , and E_8 , respectively.

Each Weyl group W stabilizes the lattice spanned by a set \mathcal{S} of simple roots, the **root lattice** of \mathcal{R} . These lattices have many interesting geometric properties and occur also in the context of sphere packings (see Conway and Sloane [CS99] and [Chapter 61](#) of this Handbook). The irreducible Coxeter groups W of Euclidean

type, or, equivalently, the infinite discrete irreducible Euclidean reflection groups, are intimately related to Weyl groups; they are also called *affine Weyl groups*.

The complexifications of the reflection hyperplanes for a finite Coxeter group give an example of a complex *hyperplane arrangement* (see [BLS⁺93], [OT92], and Chapter 6). The topology of the set-theoretic complement of these *Coxeter arrangements* in complex space has been extensively studied.

For hyperbolic reflection groups, see Vinberg [Vin85]. In hyperbolic space, a discrete irreducible reflection group need not have a fundamental region that is a simplex.

19.7 COMPLEX REGULAR POLYTOPES

Complex regular polytopes are subspace configurations in unitary complex space that share many properties with regular polytopes in real spaces. For a detailed account see Coxeter [Cox93]. The subject originated with Shephard [She52].

GLOSSARY

Complex d -polytope: A d -polytope-configuration as defined in Section 19.2, but now the elements, or *faces*, are subspaces in unitary complex d -space \mathbb{C}^d . However, unlike in real space, the subconfigurations G/F with $\dim(G) - \dim(F) = 2$ can contain more than 2 proper elements. A *complex polygon* is a complex 2-polytope.

Regular complex polytope: A complex polytope P whose (unitary) symmetry group $G(P)$ is transitive on the flags (the maximal sets of mutually incident faces).

ENUMERATION AND GROUPS

The regular complex d -polytopes P are completely known for each d . Every d -polytope can be uniquely described by a *generalized Schläfli symbol*

$$p_0\{q_1\}p_1\{q_2\}p_2 \cdots p_{d-2}\{q_{d-1}\}p_{d-1},$$

which we explain below. For $d = 1$, the regular polytopes are precisely the point sets on the complex line, which in corresponding real 2-space are the vertex sets of regular convex polygons; the Schläfli symbol is simply p if the real polygon is a p -gon. In general, the entry p_i is the Schläfli symbol for the complex 1-polytope that occurs as the 1-dimensional subconfiguration G/F of P , where F is an $(i-1)$ -face and G an $(i+1)$ -face of P such that $F \subseteq G$. As is further explained below, the p_i i -faces in this subconfiguration are cyclicly permuted by a hyperplane reflection that leaves the whole polytope invariant. Note that, unlike in real Euclidean space, a hyperplane reflection in unitary complex space need not have period 2 but can have any finite period greater than 1. The meaning of the entries q_i is also given below.

The regular complex polytopes P with $d \geq 2$ are summarized in Table 19.7.1, which includes the numbers f_0 and f_{d-1} of vertices and facets (($d-1$)-faces) and the group order. Listed are only the nonreal polytopes as well as only one polytope from each pair of duals. A complex polytope is *real* if, up to an affine transformation of \mathbb{C}^d , all its faces are subspaces that can be described by linear equations over the reals. In particular, $p_0\{q_1\}p_1 \dots p_{d-2}\{q_{d-1}\}p_{d-1}$ is real if and only if $p_i = 2$ for each i ; in this case, $\{q_1, \dots, q_{d-1}\}$ is the Schläfli symbol for the related regular polytope in real space. As in real space, each polytope $p_0\{q_1\}p_1 \dots p_{d-2}\{q_{d-1}\}p_{d-1}$ has a dual (reciprocal) and its Schläfli symbol is $p_{d-1}\{q_{d-1}\}p_{d-2} \dots p_1\{q_1\}p_0$; the symmetry groups are the same and the numbers of vertices and facets are interchanged. The polytope $p\{4\}2\{3\}2 \dots 2\{3\}2$ is the *generalized complex d-cube*, and its dual $2\{3\}2 \dots 2\{3\}2\{4\}p$ the *generalized complex d-cross-polytope*; if $p = 2$, these are the real d -cubes and d -cross-polytopes, respectively.

TABLE 19.7.1 The nonreal complex regular polytopes (up to duality).

DIMENSION	POLYTOPE	f_0	f_{d-1}	$ G(P) $
$d \geq 1$	$p\{4\}2\{3\}2 \dots 2\{3\}2$	p^d	pd	$p^d d!$
$d = 2$	3{3}3 3{6}2 3{4}3 4{3}4 3{8}2 4{6}2 4{4}3 3{5}3 5{3}5 3{10}2 5{6}2 5{4}3	8 24 24 24 72 96 96 120 120 360 600 600	8 16 24 24 48 48 72 120 120 240 240 360	24 48 72 96 144 192 288 360 600 720 1200 1800
$d = 3$	3{3}3{3}3 3{3}3{4}2	27 72	27 54	648 1296
$d = 4$	3{3}3{3}3{3}3	240	240	155 520

The symmetry group $G(P)$ of a complex regular d -polytope P is a finite unitary reflection group in \mathbb{C}^d ; if $P = p_0\{q_1\}p_1 \dots p_{d-2}\{q_{d-1}\}p_{d-1}$, then the notation for the group $G(P)$ is $p_0[q_1]p_1 \dots p_{d-2}[q_{d-1}]p_{d-1}$. If $\Phi = \{\emptyset = F_{-1}, F_0, \dots, F_{d-1}, F_d = \mathbb{C}^d\}$ is a flag of P , then for each $i = 0, 1, \dots, d-1$ there is a unitary reflection R_i that fixes F_j for $j \neq i$ and cyclicly permutes the p_i i -faces in the subconfiguration F_{i+1}/F_{i-1} of P . These generators R_i can be chosen in such a way that in terms of R_0, \dots, R_{d-1} , the group $G(P)$ has a presentation of the form

$$\left\{ \begin{array}{ll} R_i^{p_i} = 1 & (0 \leq i \leq d-1), \\ R_i R_j = R_j R_i & (0 \leq i < j-1 \leq d-2), \\ R_i R_{i+1} R_i R_{i+1} R_i \dots = R_{i+1} R_i R_{i+1} R_i R_{i+1} \dots & \text{with } q_{i+1} \text{ generators on each side } (0 \leq i \leq d-2). \end{array} \right.$$

This explains the entries q_i in the Schläfli symbol. Conversely, any d unitary reflections that satisfy the first two sets of relations, and generate a finite group, can be used to determine a regular complex polytope by a complex analogue of Wythoff's construction (see [Section 19.5](#)). If P is real, then $G(P)$ is conjugate, in the general linear group of \mathbb{C}^d , to a finite (real) Coxeter group (see [Section 19.6](#)). Complex regular polytopes are only one source for finite unitary reflection groups; there are also others [Cox93, ST54].

See Cuypers [Cuy95] for the classification of quaternionic regular polytopes (polytope-configurations in quaternionic space).

19.8 ABSTRACT REGULAR POLYTOPES

Abstract regular polytopes are combinatorial structures that generalize the familiar regular polytopes. The terminology adopted is patterned after the classical theory. Many symmetric figures discussed in earlier sections could be treated (and their structure clarified) in this more general framework. Much of the research in this area is quite recent. For a comprehensive account see McMullen and Schulte [MS02].

GLOSSARY

Abstract d -polytope: A partially ordered set P , with elements called *faces*, that satisfies the following conditions. P is equipped with a *rank function* with range $\{-1, 0, \dots, d\}$, which associates with a face F its *rank* $\text{rank } F$; if $\text{rank } F = j$, F is a *j-face*, or a *vertex*, an *edge*, or a *facet* if $j = 0, 1$, or $d - 1$, respectively. P has a unique minimal element F_{-1} of rank -1 and a unique maximal element F_d of rank d . These two elements are the *improper* faces; the others are *proper*. The *flags* (maximal totally ordered subsets) of P all contain exactly $d + 2$ faces (including F_{-1} and F_d). If $F < G$ in P , then $G/F := \{H \in P | F \leq H \leq G\}$ is said to be a *section* of P . All sections of P (including P itself) are *connected*, meaning that, given two proper faces H, H' of a section G/F , there is a sequence $H = H_0, H_1, \dots, H_k = H'$ of proper faces of G/F (for some k) such that H_{i-1} and H_i are incident for each $i = 1, \dots, k$. (That is, P is *strongly connected*.) Finally, if $F < G$ with $0 \leq \text{rank } F + 1 = j = \text{rank } G - 1 \leq d - 1$, there are exactly two j -faces H such that $F < H < G$. (Note that this last condition basically says that P is topologically real. The condition is violated for nonreal complex polytopes.)

Faces and co-faces: We can safely identify a face F of P with the section $F/F_{-1} = \{H \in P | H \leq F\}$. The section $F_d/F = \{H \in P | F \leq H\}$ is the *co-face* of P , or the *vertex figure* if F is a vertex.

Regular polytope: An abstract polytope P whose *automorphism group* $\Gamma(P)$ (the group of order-preserving permutations of P) is transitive on the flags. (Then $\Gamma(P)$ must be simply flag-transitive.)

C-group: A group Γ generated by involutions $\sigma_1, \dots, \sigma_m$ (that is, a quotient of a Coxeter group) such that the *intersection property* holds:

$$\langle \sigma_i | i \in I \rangle \cap \langle \sigma_i | i \in J \rangle = \langle \sigma_i | i \in I \cap J \rangle \quad \text{for all } I, J \subset \{1, \dots, m\}.$$

The letter “C” stands for “Coxeter.” (Coxeter groups are C-groups, but not vice versa.)

String C-group: A C-group $\Gamma = \langle \sigma_1, \dots, \sigma_m \rangle$ such that $(\sigma_i \sigma_j)^2 = 1$ if $1 \leq i < j - 1 \leq m - 1$. (Then Γ is a quotient of a Coxeter group with a string Coxeter diagram.)

Realization: For a regular (abstract) d -polytope P with vertex-set \mathcal{F}_0 , a surjection $\beta : \mathcal{F}_0 \rightarrow V$ onto a set V of points in a Euclidean space, such that each automorphism of P induces an isometric permutation of V . Then V is the *vertex set* of the realization β .

Chiral polytope: An abstract polytope P whose automorphism group $\Gamma(P)$ has exactly two orbits on the flags, with adjacent flags in different orbits. (Two flags are adjacent if they differ in exactly one face.) Chiral polytopes are an important class of nearly regular polytopes.

GENERAL PROPERTIES

Abstract 2-polytopes are isomorphic to ordinary n -gons or apeirogons (Section 19.2). Except for some degenerate cases, the abstract 3-polytopes with finite faces and vertex figures are in one-to-one correspondence with the maps on surfaces (Section 19.3). Accordingly, a finite (abstract) 4-polytope P has facets and vertex figures that are isomorphic to maps on surfaces.

The group $\Gamma(P)$ of every regular d -polytope P is a string C-group. Fix a flag $\Phi := \{F_{-1}, F_0, \dots, F_d\}$, the *base flag* of P . Then $\Gamma(P)$ is generated by *distinguished generators* $\rho_0, \dots, \rho_{d-1}$ (with respect to Φ), where ρ_i is the unique automorphism that keeps all but the i -face of Φ fixed. These generators satisfy relations

$$(\rho_i \rho_j)^{p_{ij}} = 1 \quad (i, j = 0, \dots, d-1),$$

with $p_{ii} = 1$, $p_{ij} = p_{ji} \geq 2$ ($i \neq j$), and $p_{ij} = 2$ if $|i - j| \geq 2$; in particular, $\Gamma(P)$ is a string C-group with generators $\rho_0, \dots, \rho_{d-1}$. The numbers $p_i := p_{i-1,i}$ determine the (*Schläfli*) *type* $\{p_1, \dots, p_{d-1}\}$ of P . The group $\Gamma(P)$ is a quotient of the Coxeter group $[p_1, \dots, p_{d-1}]$ (Section 19.6), but in general the quotient is proper.

Conversely, if Γ is a string C-group with generators $\rho_0, \dots, \rho_{d-1}$, then it is the group of a regular d -polytope P , and $\rho_0, \dots, \rho_{d-1}$ are the distinguished generators with respect to some base flag of P . The i -faces of P are the right cosets of the subgroup $\Gamma_i := \langle \rho_k | k \neq i \rangle$ of Γ , and in P , $\Gamma_i \varphi \leq \Gamma_j \psi$ if and only if $i \leq j$ and $\Gamma_i \varphi \cap \Gamma_j \psi \neq \emptyset$. For any $p_1, \dots, p_{d-1} \geq 2$, $[p_1, \dots, p_{d-1}]$ is a string C-group and the corresponding d -polytope is the *universal* regular d -polytope $\{p_1, \dots, p_{d-1}\}$; every other regular d -polytope of the same type $\{p_1, \dots, p_{d-1}\}$ is derived from it by making identifications. Examples are the regular spherical, Euclidean, and hyperbolic honeycombs. The one-to-one correspondence between string C-groups and the groups of regular polytopes sets up a powerful dialogue between groups on one hand and polytopes on the other.

There is also a similar such dialogue for chiral polytopes (see Schulte and Weiss [SW94]). If P is chiral and $\Phi := \{F_{-1}, F_0, \dots, F_d\}$ is its base flag, then $\Gamma(P)$ is generated by automorphisms $\sigma_1, \dots, \sigma_{d-1}$, where σ_i fixes all the faces in $\Phi \setminus \{F_{i-1}, F_i\}$ and cyclically permutes consecutive i -faces of P in the (polygonal) section F_{i+1}/F_{i-2} of rank 2. The orientation of each σ_i can be chosen in such

a way that the resulting *distinguished generators* $\sigma_1, \dots, \sigma_{d-1}$ of $\Gamma(P)$ satisfy relations

$$\sigma_i^{p_i} = (\sigma_j \sigma_{j+1} \dots \sigma_k)^2 = 1 \quad (i, j, k = 1, \dots, d-1 \text{ and } j < k),$$

with p_i determined by the type $\{p_1, \dots, p_{d-1}\}$ of P . Moreover, a certain intersection property (resembling that for C-groups) holds for $\Gamma(P)$. Conversely, if Γ is a group generated by $\sigma_1, \dots, \sigma_{d-1}$, and if these generators satisfy the above relations and the intersection property, then Γ is the group of a chiral polytope, or the rotation subgroup of index 2 in the group of a regular polytope. Each isomorphism type of chiral polytope occurs combinatorially in two *enantiomorphic* (mirror image) forms; these correspond to two sets of generators σ_i of the group determined by a pair of adjacent base flags.

Abstract polytopes are closely related to buildings and diagram geometries [Bue95, Tit74]. They are essentially the “thin diagram geometries with a string diagram.” The universal regular polytopes $\{p_1, \dots, p_{d-1}\}$ correspond to “thin buildings.”

CLASSIFICATION BY TOPOLOGICAL TYPE

Abstract polytopes are not a priori embedded into an ambient space. Therefore for abstract polytopes, the traditional enumeration of regular polytopes is replaced by the classification by global or local topological type. On the group level, this translates into the enumeration of finite string C-groups with certain kinds of presentations.

Every *locally spherical* abstract regular polytope P of rank $d+1$ is a quotient of a regular tessellation $\{p_1, \dots, p_d\}$ in spherical, Euclidean, or hyperbolic d -space; in other words, P is a regular tessellation on the corresponding spherical, Euclidean, or hyperbolic space form. In this context, the classical regular convex polytopes are precisely the abstract regular polytopes that are locally spherical and globally spherical. The *projective regular polytopes* are the regular tessellations in real projective d -space, and are obtained as quotients of the centrally symmetric regular convex polytopes under the central inversion.

Much work has also been done in the toroidal and locally toroidal case [MS02]. A *regular toroid* of rank $d+1$ is the quotient of a regular tessellation $\{p_1, \dots, p_d\}$ in Euclidean d -space by a lattice that is invariant under all symmetries of the vertex figure of $\{p_1, \dots, p_d\}$; in other words, a regular toroid is a regular tessellation on the d -torus. If $d = 2$, these are the reflexible regular torus maps of [CM80]. For $d \geq 3$ there are three infinite sequences of *cubical toroids* of type $\{4, 3^{d-2}, 4\}$, and for $d = 4$ there are two infinite sequences of *exceptional toroids* for each of the types $\{3, 3, 4, 3\}$ and $\{3, 4, 3, 3\}$. Their groups are known in terms of generators and relations.

For $d \geq 2$, the d -torus is the only d -dimensional compact Euclidean space form that can admit a regular or chiral tessellation. Further, chirality can only occur if $d = 2$ (yielding the irreflexible torus maps of [CM80]). Little is known about regular tessellations on hyperbolic space forms (again, see [CM80] and [MS02]).

For regular d -polytopes P_1 and P_2 , let $\langle P_1, P_2 \rangle$ denote the class of all regular $(d+1)$ -polytopes with facets isomorphic to P_1 and vertex figures isomorphic to P_2 . Each nonempty class $\langle P_1, P_2 \rangle$ contains a *universal polytope* denoted by $\{P_1, P_2\}$, which “covers” all other polytopes in its class. Classification by local topological

type means enumeration of all finite universal polytopes $\{P_1, P_2\}$ where P_1 and P_2 are of the prescribed (global) topological type. There are variants of this definition. A polytope Q in $\langle P_1, P_2 \rangle$ is **locally toroidal** if P_1 and P_2 are regular convex polytopes (spheres) or regular toroids, with at least one of the latter kind.

Locally toroidal regular polytopes can only exist in ranks 4, 5, and 6 [MS02]. The enumeration is complete for rank 5, and nearly complete for rank 4. In rank 6, a list of finite polytopes is known that is conjectured to be complete. The enumeration in rank 4 involves analysis of the Schläfli types $\{4, 4, r\}$ with $r = 3, 4, \{6, 3, r\}$ with $r = 3, 4, 5, 6$, and $\{3, 6, 3\}$, and their duals. Here, complete lists of finite universal regular polytopes are known for each type except $\{4, 4, 4\}$ and $\{3, 6, 3\}$; the type $\{4, 4, 4\}$ is almost settled, and for $\{3, 6, 3\}$ partial results are known. In rank 5, only the types $\{3, 4, 3, 4\}$ and its dual occur. Finally, in rank 6, there are $\{3, 3, 3, 4, 3\}$, $\{3, 3, 4, 3, 3\}$, and $\{3, 4, 3, 3, 4\}$, and their duals. On the group level, the classification of toroidal and locally toroidal polytopes amounts to the classification of certain C-groups that are defined in terms of generators and relations. These groups are quotients of Euclidean or hyperbolic Coxeter groups and are obtained from those by either one or two extra defining relations. Very little is known about the corresponding classification for chiral polytopes.

REALIZATIONS

A good number of the geometric figures discussed in the earlier sections could be described in the general context of realizations of abstract regular polytopes. For an account of realizations see [MS02] or McMullen [McM94].

Let $\beta : \mathcal{F}_0 \mapsto V$ be a realization of a regular d -polytope P , and let \mathcal{F}_j denote the set of j -faces of P ($j = -1, 0, \dots, d$). With $\beta_0 := \beta$, $V_0 := V$, then for $j = 1, \dots, d$, β recursively induces a surjection $\beta_j : \mathcal{F}_j \mapsto V_j$, with $V_j \subset 2^{V_{j-1}}$, given by

$$F\beta_j := \{G\beta_{j-1} \mid G \in \mathcal{F}_{j-1}, G \leq F\}$$

for each $F \in \mathcal{F}_j$. It is convenient to identify β and $\{\beta_j\}_{j=0}^d$ and also call the latter a realization of P . The realization is **faithful** if each β_j is a bijection; otherwise, it is **degenerate**. Its **dimension** is the dimension of the affine hull of V . Each realization corresponds to a (not necessarily faithful) representation of the automorphism group $\Gamma(P)$ as a group of Euclidean isometries.

The traditional approach in the study of regular figures starts from a Euclidean (or other) space and describes all figures of a specified kind that are regular according to some geometric definition of regularity. For example, the Grünbaum-Dress polyhedra of Section 19.4 are the realizations in \mathbb{E}^3 of abstract regular 3-polytopes P that are both discrete and faithful; their symmetry group is flag-transitive and is isomorphic to the automorphism group $\Gamma(P)$.

A rather new approach proceeds from a given abstract regular polytope P and describes all the realizations of P . For a finite P , each realization β is uniquely determined by its **diagonal vector** Δ , whose components are the squared lengths of the diagonals (pairs of vertices) in the diagonal classes of P modulo $\Gamma(P)$. Each orthogonal representation of $\Gamma(P)$ yields one or more (possibly degenerate) realizations of P . Then taking the sum of two representations of $\Gamma(P)$ is equivalent to an operation for the related realizations called a **blend**, which in turn amounts to adding the corresponding diagonal vectors. If we identify the realizations with their diagonal vectors, then the space of all realizations of P becomes a closed con-

vex cone $C(P)$, the ***realization cone of*** P , whose finer structure is given by the irreducible representations of $\Gamma(P)$. The extreme rays of $C(P)$ correspond to the ***pure*** (unblended) realizations, which are given by the irreducible representations of $\Gamma(P)$. Each realization of P is a blend of pure realizations.

For instance, a regular n -gon P has $\lfloor \frac{1}{2}n \rfloor$ diagonal classes, and for each $k = 1, \dots, \lfloor \frac{1}{2}n \rfloor$, there is a planar regular star-polygon $\{\frac{n}{k}\}$ if $(n, k) = 1$ (Section 19.2), or a “degenerate star-polygon $\{\frac{n}{k}\}$ ” if $(n, k) > 1$; the latter is a degenerate realization of P , which reduces to a line segment if $n = 2k$. For the regular icosahedron P there are 3 pure realizations. Apart from the usual icosahedron $\{3, 5\}$ itself, there is another 3-dimensional pure realization, namely the great icosahedron $\{3, \frac{5}{2}\}$ (Section 19.2). The final pure realization is induced by its covering of $\{3, 5\}/2$, the ***hemi-icosahedron*** (obtained from P by identifying antipodal vertices), all of whose diagonals are edges; thus its vertices must be those of a 5-simplex. The regular d -simplex has (up to similarity) a unique realization. The regular d -cross-polytope and d -cube have 2 and d pure realizations, respectively. For other polytopes see [BS00, MS02, MW99, MW00].

19.9 SOURCES AND RELATED MATERIAL

SURVEYS

- [Ban96]: A popular book on the geometry and visualization of polyhedral and nonpolyhedral figures with symmetries in higher dimensions.
- [BLS⁺93]: A monograph on oriented matroids and their applications.
- [BW93]: A survey on polyhedral manifolds and their embeddings in real space.
- [BCN89]: A monograph on distance-regular graphs and their symmetry properties.
- [Bue95]: A handbook of incidence geometry, with articles on buildings and diagram geometries.
- [CS99]: A monograph on sphere packings and related topics.
- [Cox70]: A short text on certain chiral tessellations of 3-dimensional manifolds.
- [Cox73]: A monograph on the traditional regular polytopes, regular tessellations, and reflection groups.
- [Cox93]: A monograph on complex regular polytopes and complex reflection groups.
- [CM80]: A monograph on discrete groups and their presentations.
- [DGS81]: A collection of papers on various aspects of symmetry, contributed in honor of H.S.M. Coxeter’s 70th birthday.
- [DuV64]: A monograph on geometric aspects of the quaternions with applications to symmetry.
- [Fej64]: A monograph on regular figures, mainly in 3 dimensions.
- [Grü67]: A monograph on convex polytopes. The second edition is a reprint of the original one, updated with extensive notes about recent developments.
- [GS87]: A monograph on plane tilings and patterns.
- [Hum90]: A monograph on Coxeter groups and reflection groups.

- [Joh]: A monograph on uniform polytopes and semiregular figures.
- [Mag74]: A book on discrete groups of Möbius transformations and non-Euclidean tessellations.
- [Mar94]: A survey on symmetric convex polytopes and a hierarchical classification by symmetry.
- [Mon87]: A book on the topology of the three-manifolds of classical plane tessellations.
- [McM94]: A survey on abstract regular polytopes with emphasis on geometric realizations.
- [MS02]: A monograph on abstract regular polytopes and their groups.
- [OT92]: A monograph on hyperplane arrangements.
- [Rob84]: A text about symmetry classes of convex polytopes.
- [Sen95]: An introduction to the geometry of mathematical quasicrystals and related tilings.
- [SF88]: A text on interdisciplinary aspects of polyhedra and their symmetries.
- [SMT⁺95]: A collection of twenty-six papers by H.S.M. Coxeter.
- [Tit74]: A text on buildings and their classification.
- [Wel77]: A monograph on three-dimensional polyhedral geometry and its applications in crystallography.
- [Zie95]: A graduate textbook on convex polytopes.

RELATED CHAPTERS

- [Chapter 3: Tilings](#)
- [Chapter 6: Oriented matroids](#)
- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 21: Polyhedral maps](#)
- [Chapter 61: Sphere packing and coding theory](#)
- [Chapter 62: Crystals and quasicrystals](#)

REFERENCES

- [ABM00] J.L. Arocha, J. Bracho, and L. Montejano. Regular projective polyhedra with planar faces, Part I. *Aequationes Math.*, 59:55–73, 2000.
- [Ban96] T.F. Banchoff. *Beyond the Third Dimension*. Freeman, New York, 1996.
- [BLS⁺93] A. Björner, M. Las Vergnas, B. Sturmfels, N. White and G.M. Ziegler. *Oriented Matroids*. Cambridge Univ. Press, 1993; second ed. 1999.
- [BB91] G. Blind and R. Blind. The semiregular polytopes. *Comment. Math. Helv.*, 66:150–154, 1991.
- [BW88] J. Bokowski and J.M. Wills. Regular polyhedra with hidden symmetries. *Math. Intelligencer*, 10:27–32, 1988.

- [Bra00] J. Bracho. Regular projective polyhedra with planar faces, Part II. *Aequationes Math.*, 59:160–176, 2000.
- [BW93] U. Brehm and J.M. Wills. Polyhedral manifolds. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 535–554. Elsevier, Amsterdam, 1993.
- [BCN89] A.E. Brouwer, A.M. Cohen, and A. Neumaier. *Distance-Regular Graphs*. Springer-Verlag, Berlin, 1989.
- [Bue95] F. Buekenhout, editor. *Handbook of Incidence Geometry*. Elsevier, Amsterdam, 1995.
- [BS00] H. Burgiel and D. Stanton. Realizations of regular abstract polyhedra of types {3, 6} and {6, 3}. *Discrete Comput. Geom.*, 24:241–255, 2000.
- [CMP98] L. Chen, R.V. Moody, and J. Patera. Non-crystallographic root systems, pages 135–178. In J. Patera, editor, *Quasicrystals and Discrete Geometry*, Amer. Math. Soc., Providence, 1998.
- [CS99] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*, third edition. Springer-Verlag, New York, 1999.
- [Cox68] H.S.M. Coxeter. Regular skew polyhedra in 3 and 4 dimensions and their topological analogues. In *Twelve Geometric Essays*, pages 75–105. Southern Illinois Univ. Press, Carbondale, 1968.
- [Cox70] H.S.M. Coxeter. *Twisted Honeycombs*. Regional Conference Series in Mathematics, volume 4. Amer. Math. Soc., Providence, 1970.
- [Cox73] H.S.M. Coxeter. *Regular Polytopes* (3rd edition). Dover, New York, 1973.
- [Cox93] H.S.M. Coxeter. *Regular Complex Polytopes* (2nd edition). Cambridge Univ. Press, 1993.
- [CM80] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups* (4th edition). Springer-Verlag, Berlin, 1980.
- [Cuy95] H. Cuypers. Regular quaternionic polytopes. *Linear Algebra Appl.*, 226/228:311–329, 1995.
- [DS82] L. Danzer and E. Schulte. Reguläre Inzidenzkomplexe, I. *Geom. Dedicata*, 13:295–308, 1982.
- [DGS81] C. Davis, B. Grünbaum, and F.A. Sherk. *The Geometric Vein (The Coxeter Festschrift)*. Springer-Verlag, New York, 1981.
- [Dre85] A.W.M. Dress. A combinatorial theory of Grünbaum's new regular polyhedra. Part II: Complete enumeration. *Aequationes Math.*, 29:222–243, 1985.
- [DuV64] P. Du Val. *Homographies, Quaternions and Rotations*. Oxford Univ. Press, 1964.
- [Fej64] L. Fejes Tóth. *Regular Figures*. Macmillan, New York, 1964.
- [Gar67] C.W.L. Garner. Regular skew polyhedra in hyperbolic three-space. *J. Canad. Math. Soc.*, 19:1179–1186, 1967.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Interscience, London, 1967; second edition edited by V. Kaibel, V. Klee, and G.M. Ziegler, volume 221 of *Graduate Texts in Math.*, Springer-Verlag, New York, 2003.
- [Grü77a] B. Grünbaum. Regularity of graphs, complexes and designs. In *Problèmes combinatoires et théorie des graphes*, pages 191–197. Number 260 of *Colloq. Int. CNRS*, Orsay, 1977.
- [Grü77b] B. Grünbaum. Regular polyhedra – old and new. *Aequationes Math.*, 16:1–20, 1977.
- [GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns*. Freeman, New York, 1987.
- [Hum90] J.E. Humphreys. *Reflection Groups and Coxeter Groups*. Cambridge Univ. Press, 1990.

- [Joh] N.W. Johnson. *Uniform Polytopes*. To appear.
- [Mag74] W. Magnus. *Noneuclidean Tessellations and Their Groups*. Academic Press, New York, 1974.
- [Mar94] H. Martini. A hierarchical classification of Euclidean polytopes with regularity properties. In T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 71–96. Kluwer, Dordrecht, 1994.
- [McM68] P. McMullen. Regular star-polytopes, and a theorem of Hess. *Proc. London Math. Soc.* (3), 18:577–596, 1968.
- [McM94] P. McMullen. Modern developments in regular polytopes. In T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 97–124. Kluwer, Dordrecht, 1994.
- [MS02] P. McMullen and E. Schulte. *Abstract Regular Polytopes*. Volume 92 of *Encyclopedia Math. Appl.* Cambridge Univ. Press, 2002.
- [MW99] B.R. Monson and A.I. Weiss. Realizations of regular toroidal maps. *Canad. J. Math.*, 51:1240–1257, 1999.
- [MW00] B.R. Monson and A.I. Weiss. Realizations of regular toroidal maps of type {4, 4}. *Discrete Comput. Geom.*, 24:453–465, 2000.
- [Mon87] J.M. Montesinos. *Classical Tessellations and Three-Manifolds*. Springer-Verlag, New York, 1987.
- [OT92] P. Orlik and H. Terao. *Arrangements of Hyperplanes*. Springer-Verlag, New York, 1992.
- [Rob84] S.A. Robertson. *Polytopes and Symmetry*. *London Math. Soc. Lecture Notes Ser.*, volume 90. Cambridge Univ. Press, 1984.
- [SW94] E. Schulte and A.I. Weiss. Chirality and projective linear groups. *Discrete Math.*, 131:221–261, 1994.
- [SW91] E. Schulte and J.M. Wills. Combinatorially regular polyhedra in three-space. In K.H. Hofmann and R. Wille, editors, *Symmetry of Discrete Mathematical Structures and Their Symmetry Groups*, pages 49–88. Heldermann Verlag, Berlin, 1991.
- [Sen95] M. Senechal. *Quasicrystals and Geometry*. Cambridge Univ. Press, 1995.
- [SF88] M. Senechal and G. Fleck. *Shaping Space*. Birkhäuser, Boston, 1988.
- [She52] G.C. Shephard. Regular complex polytopes. *Proc. London Math. Soc.* (3), 2:82–97, 1952.
- [ST54] G.C. Shephard and J.A. Todd. Finite unitary reflection groups. *Canad. J. Math.*, 6:274–304, 1954.
- [SMT⁺95] F.A. Sherk, P. McMullen, A.C. Thompson, and A.I. Weiss, editors. *Kaleidoscopes: Selected Writings of H.S.M. Coxeter*. Wiley-Interscience, New York, 1995.
- [Sti01] J. Stillwell. The story of the regular 120-cell. *Notices Amer. Math. Soc.*, 48:17–24, 2001.
- [Tit74] J. Tits. *Buildings of Spherical Type and Finite BN-Pairs*. Springer-Verlag, New York, 1974.
- [Vin85] E.B. Vinberg. Hyperbolic reflection groups. *Uspekhi Mat. Nauk*, 40:29–66, 1985. (= *Russian Math. Surveys*, 40:31–75, 1985).
- [Wel77] A.F. Wells. *Three-dimensional Nets and Polyhedra*. Wiley-Interscience, New York, 1977.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, New York, 1995.

20 POLYTOPE SKELETONS AND PATHS

Gil Kalai

INTRODUCTION

The k -dimensional **skeleton** of a d -polytope P is the set of all faces of the polytope of dimension at most k . The 1-skeleton of P is called the **graph** of P and denoted by $G(P)$. $G(P)$ can be regarded as an abstract graph whose vertices are the vertices of P , with two vertices adjacent if they form the endpoints of an edge of P .

In this chapter, we will describe results and problems concerning graphs and skeletons of polytopes. In Section 20.1 we briefly describe the situation for 3-polytopes. In Section 20.2 we consider general properties of polytopal graphs—subgraphs and induced subgraphs, connectivity and separation, expansion, and other properties. In Section 20.3 we discuss problems related to diameters of polytopal graphs in connection with the simplex algorithm and the Hirsch conjecture. The short Section 20.4 is devoted to polytopal digraphs. Section 20.5 is devoted to skeletons of polytopes, connectivity, collapsibility and shellability, empty faces and polytopes with “few vertices,” and the reconstruction of polytopes from their low-dimensional skeletons; finally we consider what can be said about the collections of all k -faces of a d -polytope, first for $k = d - 1$ and then when k is fixed and d is large compared to k .

20.1 THREE-DIMENSIONAL POLYTOPES

GLOSSARY

Convex polytopes and their *faces* (and, in particular their *vertices*, *edges*, and *facets*) are defined in [Chapter 16](#) of this Handbook.

A graph is **d -polytopal** if it is the graph of some d -polytope.

The following standard graph-theoretic concepts are used: *subgraphs*, *induced subgraphs*, the *complete graph* K_n on n vertices, *Cycles*, *trees*, a *spanning tree* of a graph, *valence* (or *degree*) of a vertex in a graph, *planar* graphs, *d -connected* graphs, *coloring* of a graph, *subdivision* of a graph, and *Hamiltonian* graphs.

We briefly discuss results on 3-polytopes. Some of the following theorems are the starting points of much research, sometimes of an entire theory. Only in a few cases are there high-dimensional analogues, and this remains an interesting goal for further research.

THEOREM 20.1.1 *Whitney* [Whi32]

Let G be the graph of a 3-polytope P . Then the graphs of faces of P are precisely the induced cycles in G that do not separate G .

THEOREM 20.1.2 *Steinitz* [Ste22]

A graph G is a graph of a 3-polytope if and only if G is planar and 3-connected.

Steinitz's theorem is the first of several theorems that describe the tame behavior of 3-polytopes. These theorems fail already in dimension four; see [Chapter 16](#).

The theory of planar graphs is a wide and rich theory. Let us quote here the fundamental theorem of Kuratowski:

THEOREM 20.1.3 *Kuratowski* [Kur22, Tho81]

A graph G is planar if and only if G does not contain a subdivision of K_5 or $K_{3,3}$.

THEOREM 20.1.4 *Lipton and Tarjan* [LT79], strengthened by *Miller* [Mil86]

The graph of every 3-polytope with n vertices can be separated, by $2\sqrt{2n}$ vertices forming a circuit in the graph, into connected components of size at most $2n/3$.

It is worth mentioning that the Koebe circle packing theorem gives a new approach to both the Steinitz and Lipton-Tarjan theorems. (See [Zie95, PA95]).

Euler's formula $V - E + F = 2$ has many applications concerning graphs of 3-polytopes; in higher dimensions, our knowledge of face numbers of polytopes (see [Chapter 18](#)) applies to the study of their graphs and skeletons. Simple applications of Euler's theorem are:

THEOREM 20.1.5

Every 3-polytopal graph has a vertex of valence at most 5. (Equivalently, every 3-polytope has a face with at most five sides.)

THEOREM 20.1.6

Every 3-polytope has either a trivalent vertex or a triangular face.

A deeper application of Euler's theorem is:

THEOREM 20.1.7 *Kotzig* [Kot55]

Every 3-polytope has two adjacent vertices the sum of whose valences is at most 13.

For a simple 3-polytope P , let $p_k = p_k(P)$ be the number of k -sized faces of P .

THEOREM 20.1.8 *Eberhard* [Ebe91]

For every finite sequence (p_k) of nonnegative integers with $\sum_{k \geq 3} (6 - k)p_k = 12$, there exists a simple 3-polytope P with $p_k(P) = p_k$ for every $k \neq 6$.

Eberhard's theorem is the starting point of a large number of results and problems, see, e.g., [Juc76, J93, GZ74]. While no high-dimensional direct analogues are known or even conjectured, the results and problems on facet-forming polytopes and nonfacets mentioned below seem related.

THEOREM 20.1.9 *Motzkin* [Mot64]

The graph of a simple 3-polytope whose facets have $0 \pmod{3}$ vertices has, all together, an even number of edges.

THEOREM 20.1.10 *Barnette* [Bar66]

Every 3-polytopal graph contains a spanning tree of maximal valence 3.

We will now describe some results and a conjecture on colorability and Hamiltonian circuits.

THEOREM 20.1.11 *Four Color Theorem: Appel-Haken [AH76, AH89, RSST97]*
The graph of every 3-polytope is 4-colorable.

THEOREM 20.1.12 *Tutte [Tut56]*

4-connected planar graphs are Hamiltonian.

Tait conjectured in 1880, and Tutte disproved in 1946, that the graph of every simple 3-polytope is Hamiltonian. This started a rich theory of trivalent planar graphs without large paths.

CONJECTURE 20.1.13 *Barnette*

Every graph of a simple 3-polytope whose facets have an even number of vertices is Hamiltonian.

Finally, there are several exact and asymptotic formulas for the numbers of distinct graphs of 3-polytopes. A remarkable enumeration theory was developed by Tutte and was further developed by several authors. We will quote one result.

THEOREM 20.1.14 *Tutte [Tut62]*

The number of rooted simplicial 3-polytopes with v vertices is

$$\frac{2(4v - 11)!}{(3v - 7)!(v - 2)!}.$$

Tutte's theory also provides efficient algorithms to generate random planar graphs of various types.

PROBLEM 20.1.15

What does a random 3-polytopal graph look like?

Motivation to study this problem (and high-dimensional extensions) comes also from physics (specifically, “quantum gravity”). See [ADJ97, Ang02, CS02]. One surprising property of random planar maps of various kinds is that the expected number of vertices of distance at most r from a given vertex behaves like r^4 (compared to r^2 for the planar grid).

20.2 GRAPHS OF d -POLYTOPES—GENERALITIES

GLOSSARY

For a graph G , TG denotes any *subdivision* of G , i.e., any graph obtained from G by replacing the edges of G by paths with disjoint interiors.

A d -polytope P is *simplicial* if all its proper faces are simplices. P is *simple* if every vertex belongs to d edges or, equivalently, if the polar of P is simplicial. P is *cubical* if all its proper faces are cubes.

A simplicial polytope P is **stacked** if it is obtained by the repeated operation of gluing a simplex along a facet.

For the definition of the *cyclic polytope* $C(d, n)$, see [Chapter 16](#).

For two graphs G and H (considered as having disjoint sets V and V' of vertices), $G + H$ denotes the graph on $V \cup V'$ that contains all edges of G and H together with all edges of the form $\{v, v'\}$ for $v \in V$ and $v' \in V'$.

A graph G is **d -connected** if G remains connected after the deletion of any set of at most $d - 1$ vertices.

An **empty simplex** of a polytope P is a set S of vertices such that S does not form a face but every proper subset of S forms a face.

A graph G whose vertices are embedded in \mathbb{R}^d is **rigid** if every small perturbation of the vertices of G that does not change the distance of adjacent vertices in G is induced by an affine rigid motion of \mathbb{R}^d . G is **generically d -rigid** if it is rigid with respect to “almost all” embeddings of its vertices into \mathbb{R}^d . (Generic rigidity is thus a graph theoretic property, but no description of it in pure combinatorial terms is known for $d > 2$; cf. [Chapter 60](#).)

A set A of vertices of a graph G is **totally separated** by a set B of vertices, if A and B are disjoint and every path between two distinct vertices in A meets B .

A graph G is an **ϵ -expander** if, for every set A of at most half the vertices of G , there are at least $\epsilon \cdot |A|$ vertices not in A that are adjacent to vertices in A .

Neighborly polytopes and $(0, 1)$ -*polytopes* are defined in Chapter 16.

The *polar dual* P^Δ of a polytope P is defined in Chapter 16.

SUBGRAPHS AND INDUCED SUBGRAPHS

THEOREM 20.2.1 *Grünbaum* [Grü65]

Every d -polytopal graph contains a TK_{d+1} .

THEOREM 20.2.2 *Kalai* [Kal87]

The graph of a simplicial d -polytope P contains a TK_{d+2} if and only if P is not stacked.

One important difference between the situation for $d = 3$ and for $d > 3$ is that K_n , for every $n > 4$, is the graph of a 4-dimensional polytope (e.g., a cyclic polytope). Simple manipulations on the cyclic 4-polytope with n vertices show:

PROPOSITION 20.2.3 *Perles* (unpublished)

- (i) *Every graph G is a spanning subgraph of the graph of a 4-polytope.*
- (ii) *For every graph G , $G + K_n$ is a d -polytopal graph for some n and some d .*

This proposition extends easily to higher-dimensional skeletons in place of graphs. It is not known what the minimal dimension is for which $G + K_n$ is d -polytopal, nor even whether $G + K_n$ (for some $n = n(G)$) can be realized in some bounded dimension uniformly for all graphs G .

CONNECTIVITY AND SEPARATION

THEOREM 20.2.4 *Balinski* [Bal61]

The graph of a d -polytope is d -connected.

A set S of d vertices that separates P must form an empty simplex; in this case, P can be obtained by gluing two polytopes along a simplex facet of each.

THEOREM 20.2.5 *Larman and Mani* [LM70]

Let G be the graph of a d -polytope. Let $e = \lfloor (d+1)/3 \rfloor$. Then for every two disjoint sequences (v_1, v_2, \dots, v_e) and (w_1, w_2, \dots, w_e) of vertices of G , there are e vertex-disjoint paths connecting v_i to w_i , $i = 1, 2, \dots, e$.

PROBLEM 20.2.6 *Larman*

Is the last theorem true for $e = \lfloor d/2 \rfloor$?

THEOREM 20.2.7 *Cauchy, Dehn, Aleksandrov, Whiteley, ...*

- (i) *Cauchy's theorem: If P is a simplicial d -polytope, $d \geq 3$, then $G(P)$ (with its embedding in \mathbb{R}^d) is rigid.*
- (ii) *Whiteley's theorem [Whi84]: For a general d -polytope P , let G' be a graph (embedded in \mathbb{R}^d) obtained from $G(P)$ by triangulating the 2-faces of P without introducing new vertices. Then G' is rigid.*

COROLLARY 20.2.8

For a simplicial d -polytope P , $G(P)$ is generically d -rigid. For a general d -polytope P and a graph G' (considered as an abstract graph) as in the previous theorem, G' is generically d -rigid.

The main combinatorial application of the above theorem is the Lower Bound Theorem (see [Chapter 18](#)) and its extension to general polytopes.

Note that Corollary 20.2.8 can be regarded also as a strong form of Balinski's theorem. It is well known and easy to prove that a generic d -rigid graph is d -connected. Therefore, for simplicial (or even 2-simplicial) polytopes, Corollary 20.2.8 implies directly that $G(P)$ is d -connected.

For general polytopes we can derive Balinski's theorem as follows. Suppose to the contrary that the graph G of a general d -polytope P is not d -connected and therefore its vertices can be separated into two parts (say, red vertices and blue vertices) by deleting a set A of $d-1$ vertices. It is easy to see that every 2-face of P can be triangulated without introducing a blue-red edge. Therefore, the resulting triangulation is not $(d-1)$ -connected and hence it is not generically d -rigid. This contradicts the assertion of Corollary 20.2.8.

Let $\mu(n, d) = f_{d-1}(C(d, n))$ be the number of facets of a cyclic d -polytope with n vertices, which, by the Upper Bound Theorem, is the maximal number of facets possible for a d -polytope with n vertices.

THEOREM 20.2.9 *Klee* [Kle64]

The number of vertices of a d -polytope that can be totally separated by n vertices is at most $\mu(n, d)$.

Klee also showed by considering cyclic polytopes with simplices stacked to each of their facets that this bound is sharp. It follows that there are graphs of simplicial d -polytopes that are not graphs of $(d-1)$ -polytopes. (After realizing that the complete graphs are 4-polytopal, one's naive thought might be that every d -polytopal graph is 4-polytopal.)

EXPANSION

Expansion properties for the graph of the d -dimensional cube are known and important in various areas of combinatorics. By direct combinatorial methods, one can obtain expansion properties of duals to cyclic polytopes. There are a few positive results and several interesting conjectures on expansion properties of graphs of large families of polytopes.

THEOREM 20.2.10 *Kalai* [Kal91]

Graphs of duals to neighborly d -polytopes with n facets are ϵ -expanders for $\epsilon = O(n^{-4})$.

This result implies that the diameter of graphs of duals to neighborly d -polytopes with n facets is $O(d \cdot n^4 \cdot \log n)$.

CONJECTURE 20.2.11 *Mihail and Vazirani* [FM92, Kai01]

Graphs of $(0,1)$ -polytopes P have the following expansion property: For every set A of at most half the vertices of P , the number of edges joining vertices in A to vertices not in A is at least $|A|$.

It is also conjectured that graphs of polytopes cannot have very good expansion properties:

CONJECTURE 20.2.12 *Polytope graphs are not very good expanders* [Kal91]

Let d be fixed. The graph of every simple d -polytope with n vertices can be separated into two parts, each having at least $n/3$ vertices, by removing $O(n^{1-1/(d-1)})$ vertices.

It is known that there are dual graphs to triangulations of S^3 that cannot be separated even by $O(n/\log n)$ vertices [MTTV97]. Dual graphs to cyclic $2k$ -polytopes with n vertices for n large look somewhat like graphs of grids in \mathbf{Z}^k and, in particular, have no separators of size $o(n^{1-1/k})$.

CONJECTURE 20.2.13 *Expansion properties of random polytopes* [Kal91]

A random simple d -polytope with n facets is an $O(1/(n-d))$ -expander.

This conjecture is vaguely stated since there are various models for random polytopes. There are models based on geometric notions of randomness. For example consider polytopes (containing the origin) that are determined by n random hyperplanes that are tangent to the unit sphere. There is much recent interest in random Gaussian perturbations of a fixed simple polytope [ST01]. We can also consider a random combinatorial type.

CONJECTURE 20.2.14 *There are only a “few” graphs of polytopes*

The number of distinct (isomorphism types) of graphs of simple d -polytopes with n vertices is at most C_d^n , where C_d is a constant depending on d .

It is even possible that the same constant applies for all dimensions and that the conjecture holds even for graphs of general polytopes. This conjecture is of interest also for dual graphs of triangulations of spheres. Conjecture 20.2.12 (and even a much weaker separation property) would imply Conjecture 20.2.14.

OTHER PROPERTIES

CONJECTURE 20.2.15 *Barnette*

Every graph of a simple d -polytope, $d \geq 4$, is Hamiltonian.

THEOREM 20.2.16

For a simple d -polytope P , $G(P)$ is 2-colorable if and only if $G(P^\Delta)$ is d -colorable.

This theorem was proved in an equivalent form for $d = 4$ by Goodman and Onishi [GO78]. (For $d = 3$ it is a classical theorem by Ore.) For the general case, see Joswig [Jos02]. This theorem is related to seeking two-dimensional analogues of Hamiltonian cycles in skeletons of polytopes and manifolds; see [Sch94].

20.3 DIAMETERS OF POLYTOPAL GRAPHS

GLOSSARY

A **d -polyhedron** is the intersection of a finite number of halfspaces in \mathbb{R}^d .

$\Delta(d, n)$ denotes the maximal diameter of the graphs of d -dimensional polyhedra P with n facets.

$\Delta_b(d, n)$ denotes the maximal diameter of the graphs of d -polytopes with n vertices.

Given a d -polyhedron P and a linear functional ϕ on \mathbb{R}^d , we denote by $G^\rightarrow(P)$ the directed graph obtained from $G(P)$ by directing an edge $\{v, u\}$ from v to u if $\phi(v) \leq \phi(u)$. $v \in P$ is a **top vertex** if ϕ attains its maximum value in P on v .

Let $H(d, n)$ be the maximum over all d -polyhedra with n facets and all linear functionals on \mathbb{R}^d of the maximum length of a minimal monotone path from any vertex to a top vertex.

Let $M(d, n)$ be the maximal number of vertices in a monotone path over all d -polyhedra with n facets and all linear functionals on \mathbb{R}^d .

For the notions of *simplicial complex*, *polyhedral complex*, *pure simplicial complex*, and the *boundary complex* of a polytope, see Chapter 18.

Given a pure $(d-1)$ -dimensional simplicial (or polyhedral) complex K , the **dual graph** $G^\Delta(K)$ of K is the graph whose vertices are the facets ($(d-1)$ -faces) of K , with two facets F, F' adjacent if $\dim(F \cap F') = d - 2$.

A pure simplicial complex K is **vertex-decomposable** if there is a vertex v of K such that $\text{lk}(v) = \{S \setminus \{v\} \mid S \in K, v \in S\}$ and $\text{ast}(v) = \{S \mid S \in K, v \notin S\}$ are both vertex-decomposable. (The complex $K = \{\emptyset\}$ consisting of the empty face alone is vertex-decomposable.)

It is a long-outstanding open problem to determine the behavior of the function $\Delta(d, n)$. In 1957, Hirsch conjectured that $\Delta(d, n) \leq n - d$. Klee and Walkup [KW67] showed that the Hirsch conjecture is false for unbounded polyhedra. The Hirsch conjecture for bounded polyhedra is still open. The special case asserting that $\Delta_b(d, 2d) = d$ is called the ***d-step conjecture***, and it was shown by Klee and Walkup to imply that $\Delta_b(d, n) \leq n - d$. Another equivalent formulation is that between any pair of vertices v and w of a polytope P there is a ***nonrevisiting path***, i.e., a path $v = v_1, v_2, \dots, v_m = w$ such that for every facet F of P , if $v_i, v_j \in F$ for $i < j$ then $v_k \notin F$ for every k with $i \leq k \leq j$.

THEOREM 20.3.1 *Klee and Walkup*

$$\Delta(d, n) \geq n - d + \min\{\lfloor d/4 \rfloor, \lfloor (n-d)/4 \rfloor\}.$$

THEOREM 20.3.2 *Holt-Klee* [HK98a, HK98b, HK98c], *Fritzsche-Holt* [FH99]

For $n > d \geq 8$

$$\Delta_b(d, n) \geq n - d.$$

THEOREM 20.3.3 *Barnette* [Bar74]

$$\Delta(d, n) \leq \frac{2}{3} \cdot (n - d + 5/2) \cdot 2^{d-3}.$$

THEOREM 20.3.4 *Kalai and Kleitman* [KK92]

$$\Delta(d, n) \leq n \cdot \binom{\log n + d}{d} \leq n^{\log d + 1}.$$

The major open problem in this area is:

PROBLEM 20.3.5

Is there a polynomial upper bound for $\Delta(d, n)$? Is there a linear upper bound for $\Delta(d, n)$?

Some special classes of polytopes are known to satisfy the Hirsch bound or to have upper bounds for their diameters that are polynomial in d and n .

THEOREM 20.3.6 *Provan and Billera* [PB80]

Let G be the dual graph that corresponds to a vertex-decomposable $(d-1)$ -dimensional simplicial complex with n vertices. Then the diameter of G is at most $n - d$.

It is known that this theorem does not imply the Hirsch conjecture (for polytopes) since there are simplicial polytopes whose boundary complexes are not vertex-decomposable. Yet such examples are not so easy to come by.

THEOREM 20.3.7 *Naddef* [Nad89]

The graph of every $(0, 1)$ d -polytope has diameter at most d .

Balinski [Bal84] proved the Hirsch bound for dual transportation polytopes, Dyer and Frieze [DF94] showed a polynomial upper bound for unimodular polyhedra, Kalai [Kal92] observed that if the ratio between the number of facets and the dimension is bounded above for the polytope and all its faces then the diameter is bounded above by a polynomial in the dimension, Kleinschmidt and Onn [KO92] proved extensions of Naddef's results to integral polytopes, and Deza and

Onn [DO95] found upper bounds for the diameter in terms of lattice points in the polytope.

The value of $\Delta(d, n)$ is a lower bound for the number of iterations needed for Dantzig's simplex algorithm for linear programming with any pivot rule. However, it is still an open problem to find pivot rules where each pivot step can be computed with a polynomial number of arithmetic operations in d and n such that the number of pivot steps needed comes close to the upper bounds for $\Delta(d, n)$ given above. See [Chapter 45](#).

The problem of routing in graphs of polytopes, i.e., finding a path between two vertices, is an interesting computational problem.

PROBLEM 20.3.8

Find an efficient routing algorithm for convex polytopes.

Using linear programming it is possible to find a path in a polytope P between two vertices that obeys the upper bounds given above such that the number of calls to the linear programming subroutine is roughly the number of edges of the path. Finding a routing algorithm for polytopes with a “small” number of arithmetic operations as a function of d and n is an interesting challenge. The subexponential simplex-type algorithms (see Chapter 45) yield subexponential routing algorithms, but improvement for routing beyond what is known for linear programming is possible.

The upper bounds for $\Delta(d, n)$ mentioned above apply even to $H(d, n)$. Klee and Minty considered a certain geometric realization of the d -cube to show that:

THEOREM 20.3.9 *Klee and Minty* [KM72]

$$M(d, 2d) \geq 2^d.$$

Recent far-reaching extensions of the Klee-Minty construction were found by Amenta and Ziegler [AZ99]. It is not known for $d > 3$ and $n \geq d + 3$ what the precise upper bound for $M(d, n)$ is and whether it coincides with the maximum number of vertices of a d -polytope with n facets given by the upper bound theorem ([Chapter 18](#)). See Pfeifle [Pfe02].

20.4 POLYTOPAL DIGRAPHS

Given a d -polytope P and a linear objective function ϕ not constant on edges, direct every edge of $G(P)$ towards the vertex with the higher value of the objective function. A directed graph obtained in this way is called a ***polytopal digraph***.

The following basic result is fundamental for the simplex algorithm and also has many applications for the combinatorial theory of polytopes.

THEOREM 20.4.1 *Folklore* (see, e.g., [Wil88])

A polytopal digraph has one sink (and one source). Moreover, every induced subgraph on the vertices of any face of the polytope has one sink (and one source).

An acyclic orientation of $G(P)$ with the property that every face has a unique sink is called an ***abstract objective function***. Joswig, Kaibel, and Körner [JKK02] showed that an acyclic orientation for which every 2-dimensional face has a unique

sink is already an abstract objective function.

The h -vector of a simplicial polytope P has a simple and important interpretation in terms of the directed graph that corresponds to the polar of P . The number $h_k(P)$ is the number of vertices v of P^Δ of outdegree k . (Recall that every vertex in a simple polytope has exactly d neighboring vertices.) Switching from ϕ to $-\phi$, one gets the Dehn-Sommerville relations $h_k = h_{d-k}$ (including the Euler relation for $k = 0$); see [Chapter 18](#).

Studying polytopal digraphs and digraphs obtained by abstract objective functions is very interesting in the three-dimensional case and in high dimensions.

THEOREM 20.4.2 *Mihalisin and Klee* [MK00]

Suppose that K is an orientation of a 3-polytopal graph G . Then the digraph K is 3-polytopal if and only if it is acyclic, has a unique source and a unique sink, and admits three independent monotone paths from the source to the sink.

Mihalisin and Klee write in their article “we hope that the present article will open the door to a broader study of polytopal digraphs.”

20.5 SKELETONS OF POLYTOPES

GLOSSARY

A pure polyhedral complex K is ***strongly connected*** if its dual graph is connected.

A ***shelling order*** of the facets of a polyhedral $(d-1)$ -dimensional sphere is an ordering of the set of facets F_1, F_2, \dots, F_n such that the simplicial complex K_i spanned by $F_1 \cup F_2 \cup \dots \cup F_i$ is a simplicial ball for every $i < n$. A polyhedral complex is ***shellable*** if there exists a shelling order of its facets.

A simplicial polytope is ***extendably shellable*** if any way to start a shelling can be continued to a shelling.

An ***elementary collapse*** on a simplicial complex is the deletion of two faces F and G so that F is maximal and G is a codimension-1 face of F that is not included in any other maximal face. A polyhedral complex is ***collapsible*** if it can be reduced to the void complex by repeated applications of elementary collapses.

A d -dimensional polytope P is ***facet-forming*** if there is a $(d+1)$ -dimensional polytope Q such that all facets of Q are combinatorially isomorphic to P . If no such Q exists, P is called a ***nonfacet***.

A ***rational polytope*** is a polytope whose vertices have rational coordinates. (Not every polytope is combinatorially isomorphic to a rational polytope; see [Chapter 16](#).)

A d -polytope P is ***k -simplicial*** if all its faces of dimension at most k are simplices. P is ***k -simple*** if its polar dual P^Δ is k -simplicial.

Zonotopes are defined in Chapters 16 and 18.

Let K be a polyhedral complex. An ***empty simplex*** S of K is a minimal nonface of K , i.e., a subset S of the vertices of K with S itself not in K , but every proper subset of S in K .

Let K be a polyhedral complex and let U be a subset of its vertices. The *induced subcomplex* of K on U , denoted by $K[U]$, is the set of all faces in K whose vertices belong to U . An *empty face* of K is an induced polyhedral subcomplex of K that is homeomorphic to a polyhedral sphere. An empty 2-dimensional face is called an *empty polygon*. An *empty pyramid* of K is an induced subcomplex of K that consists of all the proper faces of a pyramid over a face of K .

CONNECTIVITY AND SUBCOMPLEXES

THEOREM 20.5.1 *Grünbaum* [Grü65]

The i -skeleton of every d -polytope contains a subdivision of $\text{skel}_i(\Delta^d)$, the i -skeleton of a d -simplex.

THEOREM 20.5.2 *Folklore*

- (i) *For $i > 0$, $\text{skel}_i(P)$ is strongly connected.*
- (ii) *For every face F , let $U_i(F)$ be the set of all i -faces of P containing F . Then if $i > \dim F$, $U_i(F)$ is strongly connected.*

Part (ii) follows at once from the fact that the faces of P containing F correspond to faces of the quotient polytope P/F . However, properties (i) and (ii) together are surprisingly strong, and all the known upper bounds for diameters of graphs of polytopes rely only on properties (i) and (ii) for the dual polytope.

THEOREM 20.5.3 *van Kampen and Flores* [vKa32, Flo32, Wu65]

For $i \geq \lfloor d/2 \rfloor$, $\text{skel}_i(\Delta^{d+1})$ is not embeddable in S^{d-1} (and hence not in the boundary complex of any d -polytope).

(This extends the fact that K_5 is not planar.)

CONJECTURE 20.5.4 *Lockeberg*

For every partition of $d = d_1 + d_2 + \cdots + d_k$ and two vertices v and w of P , there are k disjoint paths between v and w such that the i th path is a path of d_i -faces in which any two consecutive faces have $(d_i - 1)$ -dimensional intersection.

SHELLABILITY AND COLLAPSIBILITY

THEOREM 20.5.5 *Bruggesser and Mani* [BM71]

Boundary complexes of polytopes are shellable.

The proof of Bruggesser and Mani is based on starting with a point near the center of a facet and moving from this point to infinity, and back from the other direction, keeping track of the order in which facets are seen. This proves a stronger form of shellability, in which each K_i is the complex spanned by all the facets that can be seen from a particular point in \mathbb{R}^d . It follows from shellability that:

THEOREM 20.5.6

Polytopes are collapsible.

On the other hand,

THEOREM 20.5.7 *Ziegler [Zie98]*

There are d -polytopes, $d \geq 4$, whose boundary complexes are not extendably shellable.

THEOREM 20.5.8

There are triangulations of the $(d-1)$ -sphere that are not shellable.

Lickorish [Lic91] produced explicit examples of nonshellable triangulations of S^3 . His result was that a triangulation containing a sufficiently complicated knotted triangle was not shellable. Hachimori and Ziegler [HZ00] produced simple examples and showed that a triangulation containing any knotted triangle is not “constructible,” constructibility being a strictly weaker notion than shellability. For more on shellability, see [DK78, Bjö92].

FACET-FORMING POLYTOPES AND SMALL LOW-DIMENSIONAL FACES

THEOREM 20.5.9 *Perles and Shephard [PS67]*

Let P be a d -polytope such that the maximum number of k -faces of P on any $(d-2)$ -sphere in the skeleton of P is at most $(d-1-k)/(d+1-k)f_k(P)$. Then P is a nonfacet.

An example of a nonfacet that is simple was found by Barnette [Bar69]. Some of the proofs of Perles and Shephard use metric properties of polytopes, and for a few of the results alternative proofs using shellability were found by Barnette [Bar69].

THEOREM 20.5.10 *Schulte [Sch85]*

The cuboctahedron and the icosidodecahedron are nonfacets.

PROBLEM 20.5.11

Is the icosahedron facet-forming?

For all other regular polytopes the situation is known. The simplices and cubes in any dimension and the 3-dimensional octahedron are facet-forming. All other regular polytopes with the exception of the icosahedron are known to be nonfacets.

It is very interesting to see what can be said about metric properties of facets (or of low-dimensional faces) of a convex polytope.

THEOREM 20.5.12 *Bárány (unpublished)*

There is an $\epsilon > 0$ such that every d -polytope, $d > 2$, has a facet F for which no balls B_1 of radius R and B_2 of radius $(1+\epsilon)R$ satisfy $B_1 \subset F \subset B_2$.

The stronger statement where balls are replaced by ellipses is open.

Next, we try to understand if it is possible for all the k -faces of a d -polytope to be isomorphic to a given polytope P . The following conjecture asserts that if d is large with respect to k , this can happen only if P is either a simplex or a cube.

CONJECTURE 20.5.13 *Kalai* [Kal90]

For every k there is a $d(k)$ such that every d -polytope with $d > d(k)$ has a k -face that is either a simplex or combinatorially isomorphic to a k -dimensional cube.

Recently, Julian Pfeifle showed, on the basis of the Wythoff construction (see [Chapter 19](#)), that $d(k) > (2k - 1)(k - 1)$, for $k \geq 3$.

For simple polytopes, it follows from the next theorem that if $d > ck^2$ then every d -polytope has a k -face F such that $f_r(F) \leq f_r(C_k)$. (Here, C_k denotes the k -dimensional cube.)

THEOREM 20.5.14 *Nikulin* [Nik86]

The average number of r -dimensional faces of a k -dimensional face of a simple d -dimensional polytope is at most

$$\binom{d-r}{d-k} \cdot \left(\binom{\lfloor d/2 \rfloor}{r} + \binom{\lfloor (d+1)/2 \rfloor}{r} \right) / \left(\binom{\lfloor d/2 \rfloor}{k} + \binom{\lfloor (d+1)/2 \rfloor}{k} \right).$$

Nikulin's theorem appeared in his study of reflection groups in hyperbolic spaces. The existence of reflection groups of certain types implies some combinatorial conditions on their fundamental regions (which are polytopes), and Vinberg [Vin85], Nikulin [Nik86], Khovanski [Kho86], and others showed that in high dimensions these combinatorial conditions lead to a contradiction. There are still many open problems in this direction: in particular, to narrow the gap between the dimensions above for which those reflection groups cannot exist and the dimensions for which such groups can be constructed.

THEOREM 20.5.15 *Kalai* [Kal90]

Every d -polytope for $d \geq 5$ has a 2-face with at most 4 vertices.

THEOREM 20.5.16 *Meisinger, Kleinschmidt, and Kalai* [MKK00]

Every rational d -polytope for $d \geq 9$ has a 3-face with at most 150 vertices.

The previous two theorems and the next one are proved using the linear inequalities for flag numbers that are known via intersection homology of toric varieties; see [Chapter 18](#). One can also study, in a similar fashion, quotients of polytopes.

CONJECTURE 20.5.17 *Perles*

For every k there is a $d'(k)$ such that every d -polytope with $d > d'(k)$ has a k -dimensional quotient that is a simplex.

As was mentioned in the first section, $d'(2) = 3$. The 24-cell, which is a regular 4-polytope all of whose faces are octahedra, shows that $d'(3) > 4$.

THEOREM 20.5.18 *Meisinger, Kleinschmidt, and Kalai* [MKK00]

Every d -polytope with $d \geq 9$ has a 3-dimensional quotient that is a simplex.

PROBLEM 20.5.19

For which values of k and r are there d -polytopes other than the d -simplex that are both k -simplicial and r -simple?

It is known that this can happen only when $k+r \leq d$. There are infinite families of $(d-2)$ -simplicial and 2-simple polytopes, and some examples of $(d-3)$ -simplicial and 3-simple d -polytopes.

Concerning this problem Peter McMullen recently noted that the polytopes r_{st} , discussed in Coxeter's classic book on regular polytopes [Cox63] in Sections 11.8 and 11.x, are $(r+2)$ -simplicial and $(d-r-2)$ -simple, where $d = r + s + t + 1$. These so-called **Gosset-Elite polytopes** arise by the Wythoff construction from the finite reflection groups (see Chapter 19 of this Handbook); we obtain a finite polytope whenever the reflection group generated by the Coxeter diagram with r, s, t nodes on the three arms is finite, that is, when

$$1/(r+1) + 1/(s+1) + 1/(t+1) > 1.$$

The largest exceptional example, 2_{41} , is related to the Weyl group E_8 . The Gosset-Elite polytope 2_{41} is a 4-simple 4-simplicial 8-polytope with 2160 vertices. Are there 5-simplicial 5-simple 10-polytopes?

THEOREM 20.5.20

For $d > 2$, there is no cubical d -polytope P whose dual is also cubical.

I am not aware of a reference for this result but it can easily be proved by exhibiting a covering map from the standard cubical complex realizing \mathbb{R}^{d-1} into the boundary complex of P .

We have considered the problem of finding very special polytopes as “subobjects” (faces, quotients) of arbitrary polytopes. What about realizing arbitrary polytopes as “subobjects” of very special polytopes? There is an old conjecture that every polytope can be realized as a subpolytope (namely the convex hull of a subset of the vertices) of a stacked polytope. Perles and Sturmfels asked whether every simplicial d -polytope can be realized as the quotient of some neighborly even-dimensional polytope. (Recall that a $2m$ -polytope is **neighborly** if every m vertices are the vertices of an $(m-1)$ -dimensional face.) Kortenkamp [Kor97] proved that this is the case for d -polytopes with at most $d+4$ vertices. For general polytopes, “neighborly polytopes” should be replaced here by “weakly neighborly” polytopes, introduced by Bayer [Bay93], which are defined by the property that every set of k vertices is contained in a face of dimension at most $2k-1$. The only theorem of this flavor I am aware of is by Billera and Sarangarajan [BS96], who proved that every $(0,1)$ -polytope is a face of a traveling salesman polytope.

RECONSTRUCTION

THEOREM 20.5.21 *An extension of Whitney's theorem* [Grü67]
 *d -polytopes are determined by their $(d-2)$ -skeleto*n.

THEOREM 20.5.22 *Perles* (unpublished, 1973)

*Simplicial d -polytopes are determined by their $\lfloor d/2 \rfloor$ -skeleto*n.

This follows from the following theorem (here, $\text{ast}(F, P)$ is the complex formed by the faces of P that are disjoint to all vertices in F).

THEOREM 20.5.23 *Perles* (1973)

Let P be a simplicial d -polytope.

- (i) *If F is a k -face of P , then $\text{skel}_{d-k-2}(\text{ast}(F, P))$ is contractible in $\text{skel}_{d-k-1}(\text{ast}(F, P))$.*

- (ii) If F is an empty k -simplex, then $\text{ast}(F, P)$ is homotopically equivalent to S^{d-k} ; hence, $\text{skel}_{d-k-2}(\text{ast}(F, P))$ is not contractible in $\text{skel}_{d-k-1}(\text{ast}(F, P))$.

An extension of Perles's theorem for manifolds with vanishing middle homology was proved by Dancis [Dan84].

THEOREM 20.5.24 *Blind and Mani-Levitska* [BM87]

Simple polytopes are determined by their graphs.

Blind and Mani-Levitska described their theorem in a dual form and considered $(d-1)$ -dimensional “puzzles” whose pieces are simplices and we wish to reconstruct the puzzle based on the “local” information of which two simplices share a facet. Joswig extended their result to more general puzzles where the pieces are general $(d-1)$ -dimensional polytopes, and the way in which every two pieces sharing a facet are connected is also prescribed. A simple proof is given in [Kal88]. This proof also shows that k -dimensional skeletons of simplicial polytopes are also determined by their “puzzle.” When this is combined with Perles’s theorem it follows that:

THEOREM 20.5.25 *Kalai and Perles*

Simplicial d -polytopes are determined by the incidence relations between i - and $(i+1)$ -faces for every $i > \lfloor d/2 \rfloor$.

CONJECTURE 20.5.26 *Haase and Ziegler*

Let G be the graph of a simple 4-polytope. Let H be an induced, nonseparating, 3-regular, 3-connected planar subgraph of G . Then H is the graph of a facet of P .

Haase and Ziegler [HZ02] showed that this is not the case if H is not planar. Their proof touches on the issue of embedding knots in the skeletons of 4-polytopes.

PROBLEM 20.5.27

Are simplicial spheres determined by the incidence relations between their facets and subfacets?

THEOREM 20.5.28 *Björner, Edelman, and Ziegler* [BEZ90]

Zonotopes are determined by their graphs.

THEOREM 20.5.29 *Babson, Finschi, and Fukuda* [BFF01]

Duals of cubical zonotopes are determined by their graphs.

In all instances of the above theorems except the single case of the theorem of Blind and Mani-Levitska, the proofs give reconstruction algorithms that are polynomial in the data. It is an open question if a polynomial algorithm exists to determine a simple polytope from its graph. A polynomial “certificate” for reconstruction was recently found by Joswig, Kaibel, and Körner [JKK02].

An interesting problem was whether there is an e -dimensional polytope other than the d -cube with the same graph as the d -cube.

THEOREM 20.5.30 *Joswig and Ziegler* [JZ00]

For every $d \geq e \geq 4$ there is an e -dimensional cubical polytope with 2^d vertices whose $(\lfloor e/2 \rfloor - 1)$ -skeleton is combinatorially isomorphic to the $(\lfloor e/2 \rfloor - 1)$ -skeleton of a d -dimensional cube.

Earlier, Babson, Billera, and Chan [BBC97] found such a construction for cubical spheres.

Another issue of reconstruction for polytopes that was studied extensively is the following: In which cases does the combinatorial structure of a polytope determine its geometric structure (up to projective transformations)? Such polytopes are called *projectively unique*, and the major unsolved problem is:

PROBLEM 20.5.31

Are there only finitely many projectively unique polytopes in each dimension?

McMullen [McM76] constructed projectively unique d -polytopes with $3^{d/3}$ vertices.

EMPTY FACES AND POLYTOPES WITH FEW VERTICES

THEOREM 20.5.32 *Perles* (unpublished, 1970)

Let $f(d, k, b)$ be the number of combinatorial types of k -skeletons of d -polytopes with $d + b + 1$ vertices. Then, for fixed b and k , $f(d, k, b)$ is bounded.

This follows from:

THEOREM 20.5.33 *Perles* (unpublished, 1970)

The number of empty i -pyramids for d -polytopes with $d + b$ vertices is bounded by a function of i and b .

For another proof of this theorem see [Kal94].

For a d -polytope P , let $e_i(P)$ denote the number of empty i -simplices of P .

PROBLEM 20.5.34

Characterize the sequence of numbers $(e_1(P), e_2(P), \dots, e_d(P))$ arising from simplicial d -polytopes and from general d -polytopes.

The following theorem, which was motivated by commutative-algebraic concerns, confirmed a conjecture by Kleinschmidt, Kalai, and Lee [Kal94].

THEOREM 20.5.35 *Migliore and Nagel* [MN03]

For all simplicial d -polytopes with prescribed h -vector $h = (h_0, h_1, \dots, h_d)$, the number of i -dimensional empty simplices is maximized by the Billera-Lee polytopes $P_{BL}(h)$.

$P_{BL}(h)$ is the polytope constructed by Billera and Lee [BL81] (see [Chapter 18](#)) in their proof of the sufficiency part of the g -theorem. Migliore and Nagel proved that for a prescribed f -vector, the Billera-Lee polytopes maximize even more general parameters that arise in commutative algebra: the sum of the i th Betti numbers of induced subcomplexes on j vertices for every i and j . (The case $j = i + 2$ reduces to counting missing faces.) It is quite possible that the theorem of Migliore and Nagel extends to general simplicial spheres with prescribed h -vector and to general polytopes with prescribed (toric) h -vector. (However, it is not yet known in these cases that the h -vectors are always those of Billera-Lee polytopes; see [Chapter 18](#).)

20.6 CONCLUDING REMARKS AND EXTENSIONS TO MORE GENERAL OBJECTS

The reader who compares this chapter with other chapters on convex polytopes may notice the sporadic nature of the results and problems described here. Indeed, it seems that our main limits in understanding the combinatorial structure of polytopes still lie in our ability to raise the right questions. Another feature that comes to mind (and is not unique to this area) is the lack of examples, methods of constructing them, and means of classifying them.

We have considered mainly properties of general polytopes and of simple or simplicial polytopes. There are many classes of polytopes that are either of intrinsic interest from the combinatorial theory of polytopes, or that arise in various other fields, for which the problems described in this chapter are interesting.

Most of the results of this chapter extend to much more general objects than convex polytopes. Finding combinatorial settings for which these results hold is an interesting and fruitful area. On the other hand, the results described here are not sufficient to distinguish polytopes from larger classes of polyhedral spheres, and finding delicate combinatorial properties that distinguish polytopes is an important area of research. Few of the results on skeletons of polytopes extend to skeletons of other convex bodies [LR70, LR71, GL81], and relating the combinatorial theory of polytopes with other aspects of convexity is a great challenge.

20.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Grünbaum [Grü75] is a survey on polytopal graphs and many results and further references can be found there. More material on the topic of this chapter and further relevant references can also be found in [Grü67, Zie95, BMSW94, KK95, BL93]. Martini's chapter in [BMSW94] is on the regularity properties of polytopes (a topic not covered here; cf. [Chapter 19](#)), and contains further references on facet-forming polytopes and nonfacets. The original papers on facet-forming polytopes and nonfacets contain many more results, and describe relations to questions on tiling spaces with polyhedra. Other chapters of [BMSW94] are also relevant to the topic of this chapter.

RELATED CHAPTERS

- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 18: Face numbers of polytopes and complexes](#)
- [Chapter 45: Linear programming](#)
- Chapters [7](#), [17](#), [19](#), [21](#), [46](#), and [60](#) are also related to some parts of this chapter.

REFERENCES

- [ADJ97] J. Ambjorn, B. Durhuus, and T. Jonsson. *Quantum Geometry*. Cambridge University Press, 1997.
- [AZ99] N. Amenta and G.M. Ziegler. Deformed products and maximal shadows. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemp. Math.*, pages 57–90. Amer. Math. Soc., Providence, 1999.
- [Ang02] O. Angel. Growth and percolation on the uniform infinite planar triangulation. [arXiv:math.PR/0208123](https://arxiv.org/abs/math/0208123).
- [AH76] K. Appel and W. Haken. Every planar map is four colorable. *Bull. Amer. Math. Soc.*, 82:711–712, 1976.
- [AH89] K. Appel and W. Haken. *Every Planar Map is Four Colorable*, volume 98 of *Contemporary Mathematics*. Amer. Math. Soc., Providence, 1989.
- [BBC97] E.K. Babson, L.J. Billera, and C.S. Chan. Neighborly cubical spheres and a cubical lower bound conjecture. *Israel J. Math.*, 102:297–315, 1997.
- [BFF01] E.K. Babson, L. Finschi, and K. Fukuda. Cocircuit graphs and efficient orientation reconstruction in oriented matroids. *European J. Combin.*, 22:587–600, 2001.
- [Bal61] M.L. Balinski. On the graph structure of convex polyhedra in n -space. *Pacific J. Math.*, 11:431–434, 1961.
- [Bal84] M.L. Balinski. The Hirsch conjecture for dual transportation polyhedra. *Math. Oper. Res.*, 9:629–633, 1984.
- [Bar66] D.W. Barnette. Trees in polyhedral graphs. *Canad. J. Math.*, 18:731–736, 1966.
- [Bar69] D.W. Barnette. A simple 4-dimensional nonfacet. *Israel J. Math.*, 7:16–20, 1969.
- [Bar74] D.W. Barnette. An upper bound for the diameter of a polytope. *Discrete Math.*, 10:9–13, 1974.
- [Bar80] D.W. Barnette. Nonfacets for shellable spheres. *Israel J. Math.*, 35:286–288, 1980.
- [Bay93] M.M. Bayer. Equidecomposable and weakly neighborly polytopes. *Israel J. Math.*, 81:301–320, 1993.
- [BL93] M.M. Bayer and C.W. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 485–534. North-Holland, Amsterdam, 1993.
- [BL81] L.J. Billera and C.W. Lee. A proof of the sufficiency of McMullen’s conditions for f -vectors of simplicial convex polytopes. *J. Combin. Theory Ser. A*, 31:237–255, 1981.
- [BS96] L.J. Billera and A. Sarangarajan. All 0-1 polytopes are traveling salesman polytopes. *Combinatorica*, 16:175–188, 1996.
- [BMSW94] T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors. *Polytopes: Abstract, Convex and Computational*. Volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.* Kluwer, Dordrecht, 1994.
- [Bjö92] A. Björner. Homology and shellability of matroids and geometric lattices. In N. White, editor, *Matroid Applications*, volume Vol. 40 of *Encyclopedia of Mathematics*, pages 226–283. Cambridge University Press, 1992.
- [BEZ90] A. Björner, P.H. Edelman, and G.M. Ziegler. Hyperplane arrangements with a lattice of regions. *Discrete Comput. Geom.*, 5:263–288, 1990.

- [BM87] R. Blind and P. Mani-Levitska. On puzzles and polytope isomorphisms. *Aequationes Math.*, 34:287–297, 1987.
- [BM71] H. Bruggesser and P. Mani. Shellable decompositions of cells and spheres. *Math. Scand.*, 29:197–205, 1971.
- [CS02] P. Chassaing and G. Schaeffer. Random planar lattices and integrated superBrownian excursion. [arXiv:math.CO/0205226](https://arxiv.org/abs/math/0205226).
- [Cox63] H.S.M. Coxeter. *Regular Polytopes*. Macmillan, New York, second edition, 1963. Corrected reprint, Dover, New York, 1973.
- [DK78] G. Danaraj and V. Klee. Which spheres are shellable? In B. Alspach, P. Hell, and D.J. Miller, editors, *Algorithmic Aspects of Combinatorics (Vancouver Island BC, 1976)*, volume 2 of *Ann. Discrete Math.*, pages 33–52, 1978.
- [Dan84] J. Dancis. Triangulated n -manifolds are determined by their $[n/2]+1$ -skeletons. *Topology Appl.*, 18:17–26, 1984.
- [DO95] M.M. Deza and S. Onn. Lattice-free polytopes and their diameter. *Discrete Comput. Geom.*, 13:59–75, 1995.
- [DF94] M. Dyer and A. Frieze. Random walks, totally unimodular matrices, and a randomised dual simplex algorithm. *Math. Programming* 64:1–16, 1994.
- [Ebe91] V. Eberhard. *Zur Morphologie der Polyeder*. Teubner, Leipzig, 1891.
- [FM92] T. Feder and M. Mihail. Balanced matroids. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 26–38, 1992.
- [Fis74] J.C. Fisher. An existence theorem for simple convex polyhedra. *Discrete Math.*, 7:75–97, 1974.
- [Flo32] A. Flores. Über n -dimensionale Komplexe die im R_{2n+1} absolut selbstverschlungen sind. *Ergeb. Math. Kollog.*, 6:4–7, 1932/1934.
- [FH99] K. Fritzsche and F.B. Holt. More polytopes meeting the conjectured Hirsch bound. *Discrete Math.*, 205:77–84, 1999.
- [GL81] S. Gallivan and D.G. Larman. Further results on increasing paths in the one-skeleton of a convex body. *Geom. Dedicata*, 11:19–29, 1981.
- [GO78] J.E. Goodman and H. Onishi. Even triangulations of S^3 and the coloring of graphs. *Trans. Amer. Math. Soc.*, 246:501–510, 1978.
- [Grü65] B. Grünbaum. On the facial structure of convex polytopes. *Bull. Amer. Math. Soc.*, 71:559–560, 1965.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Interscience, London, 1967. Revised edition (V. Kaibel, V. Klee, and G.M. Ziegler, editors), Springer-Verlag, New York, 2003.
- [Grü68] B. Grünbaum. Some analogues of Eberhard’s theorem on convex polytopes. *Israel J. Math.*, 6:398–411, 1968.
- [Grü69] B. Grünbaum. Planar maps with prescribed types of vertices and faces. *Mathematika*, 16:28–36, 1969.
- [Grü75] B. Grünbaum. Polytopal graphs. In D.R. Fulkerson, editor, *Studies in Graph Theory*, pages 201–224. Math. Assoc. Amer., Washington, 1975.
- [GZ74] B. Grünbaum and J. Zaks. The existence of certain planar maps. *Discrete Math.*, 10:93–115, 1974.
- [HZ02] C. Haase and G.M. Ziegler. Examples and counterexamples for the Perles conjecture. *Discrete Comput. Geom.*, 28:29–44, 2002.

- [HZ00] M. Hachimori and G.M. Ziegler. Decompositions of simplicial balls and spheres with knots consisting of few edges. *Math. Z.*, 235:159–171, 2000.
- [HK98a] F. Holt and V. Klee. Counterexamples to the strong d -step conjecture for $d \geq 5$. *Discrete Comput. Geom.*, 19:33–46, 1998.
- [HK98b] F. Holt and V. Klee. Many polytopes meeting the conjectured Hirsch bound. *Discrete Comput. Geom.*, 20:1–17, 1998.
- [HK98c] F. Holt and V. Klee. A proof of the strict monotone 4-step conjecture. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry (Mount Holyoke 1996)*, volume 223 of *Contemporary Mathematics*, Amer. Math. Soc., Providence, 1998, pages 201–216.
- [J93] S. Jendrol'. On face vectors and vertex vectors of convex polyhedra. *Discrete Math.*, 118:119–144, 1993.
- [Joc93] W. Jockusch. The lower and upper bound problems for cubical polytopes. *Discrete Comput. Geom.*, 9:159–163, 1993.
- [Jos02] M. Joswig. Projectivities in simplicial complexes and colorings of simple polytopes. *Math. Z.*, 240:243–259, 2002.
- [JKK02] M. Joswig, V. Kaibel, and F. Körner. On the k -systems of a simple polytope. *Israel J. Math.*, 129:109–118, 2002.
- [JZ00] M. Joswig and G.M. Ziegler. Neighborly cubical polytopes. *Discrete Comput. Geom.*, 24:325–344, 2000.
- [Juc76] E. Jucovič. On face-vectors and vertex-vectors of cell-decompositions of orientable 2-manifolds. *Math. Nachr.*, 73:285–295, 1976.
- [Kai01] V. Kaibel. On the expansion of graphs of 0/1-polytopes. Tech. Rep., TU Berlin, 2001; [arXiv:math.CO/0112146](https://arxiv.org/abs/math/0112146).
- [Kal87] G. Kalai. Rigidity and the lower bound theorem I. *Invent. Math.*, 88:125–151, 1987.
- [Kal88] G. Kalai. A simple way to tell a simple polytope from its graph. *J. Combin. Theory Ser. A*, 49:381–383, 1988.
- [Kal90] G. Kalai. On low-dimensional faces that high-dimensional polytopes must have. *Combinatorica*, 10:271–280, 1990.
- [Kal91] G. Kalai. The diameter of graphs of convex polytopes and f -vector theory. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—the Victor Klee Festschrift*, volume 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, Amer. Math. Soc., Providence, 1991, pages 387–411.
- [Kal92] G. Kalai. Upper bounds for the diameter and height of graphs of convex polyhedra. *Discrete Comput. Geom.*, 8:363–372, 1992.
- [Kal94] G. Kalai. Some aspects in the combinatorial theory of convex polytopes. In [BMSW94], pages 205–230.
- [KKM00] G. Kalai, P. Kleinschmidt, and G. Meisinger. Flag numbers and FLAGTOOL. In [KZ00], pages 75–103.
- [KK92] G. Kalai and D.J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bull. Amer. Math. Soc.*, 26:315–316, 1992.
- [KZ00] G. Kalai and G.M. Ziegler, editors. *Polytopes — Combinatorics and Computation*, volume 29 of *DMV Seminars*. Birkhäuser, Basel, 2000.
- [Kho86] A.G. Khovanskii. Hyperplane sections of polyhedra, toric varieties and discrete groups in Lobachevskii space. *Funktional. Anal. i Prilozhen.*, 20:50–61, 96, 1986.

- [Kle64] V. Klee. A property of d -polyhedral graphs. *J. Math. Mech.*, 13:1039–1042, 1964.
- [KK87] V. Klee and P. Kleinschmidt. The d -step conjecture and its relatives. *Math. Oper. Res.*, 12:718–755, 1987.
- [KK95] V. Klee and P. Kleinschmidt. Polyhedral complexes and their relatives. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 875–917. North-Holland, Amsterdam, 1995.
- [KM72] V. Klee and G.J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities, III*, pages 159–175. Academic Press, New York, 1972.
- [KW67] V. Klee and D.W. Walkup. The d -step conjecture for polyhedra of dimension $d < 6$. *Acta Math.*, 117:53–78, 1967.
- [KO92] P. Kleinschmidt and S. Onn. On the diameter of convex polytopes. *Discrete Math.*, 102:75–77, 1992.
- [Kor97] U.H. Kortenkamp. Every simplicial polytope with at most $d + 4$ vertices is a quotient of a neighborly polytope. *Discrete Comput. Geom.*, 18:455–462, 1997.
- [Kot55] A. Kotzig. Contribution to the theory of Eulerian polyhedra. *Mat.-Fyz. Časopis Slovensk. Akad. Vied*, 5:101–113, 1955.
- [Kur22] C. Kuratowski. Sur l'opération A de l'analysis situs. *Fund. Math.* 3:182–199, 1922.
- [Lar70] D.G. Larman. Paths on polytopes. *Proc. London Math. Soc.*, 20:161–178, 1970.
- [LM70] D.G. Larman and P. Mani. On the existence of certain configurations within graphs and the 1-skeletons of polytopes. *Proc. London Math. Soc.*, 20:144–160, 1970.
- [LR70] D.G. Larman and C.A. Rogers. Paths in the one-skeleton of a convex body. *Mathematika*, 17:293–314, 1970.
- [LR71] D.G. Larman and C.A. Rogers. Increasing paths on the one-skeleton of a convex body and the directions of line segments on the boundary of a convex body. *Proc. London Math. Soc.*, 23:683–698, 1971.
- [Lic91] W.B.R. Lickorish. Unshellable triangulations of spheres. *European J. Combin.*, 12:527–530, 1991.
- [LT79] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. *SIAM J. Applied Math.*, 36:177–189, 1979.
- [McM76] P. McMullen. Constructions for projectively unique polytopes. *Discrete Math.*, 14:347–358, 1976.
- [MKK00] G. Meisinger, P. Kleinschmidt, and G. Kalai. Three theorems, with computer-aided proofs, on three-dimensional faces and quotients of polytopes. *Discrete Comput. Geom.*, 24:413–420, 2000. The Branko Grünbaum birthday issue (G. Kalai and V. Klee, eds.).
- [MN03] J. Migliore and U. Nagel. Reduced arithmetically Gorenstein schemes and simplicial polytopes with maximal Betti numbers. *Adv. Math.*, 180:1–63, 2003.
- [MK00] J. Mihalisin and V. Klee. Convex and linear orientations of polytopal graphs. *Discrete Comput. Geom.*, 24:421–436, 2000. The Branko Grünbaum birthday issue (G. Kalai and V. Klee, eds.).
- [Mil86] G.L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *J. Comput. System Sci.*, 32:265–279, 1986.
- [MTTV97] G.L. Miller, S.-H. Teng, W. Thurston, and S.A. Vavasis. Separators for sphere-p packings and nearest neighbor graphs. *J. ACM*, 44:1–29, 1997.
- [Mot64] T.S. Motzkin. The evenness of the number of edges of a convex polyhedron. *Proc. Nat. Acad. Sci. U.S.A.*, 52:44–45, 1964.

- [Nad89] D. Naddef. The Hirsch conjecture is true for $(0,1)$ -polytopes. *Math. Programming*, 45:109–110, 1989.
- [Nik86] V.V. Nikulin. Discrete reflection groups in Lobachevsky spaces and algebraic surfaces. In volume 1 of *Proc. Internat. Congr. Math., Berkeley, 1986*, pages 654–671.
- [PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. Wiley-Interscience, New York, 1995.
- [PS67] M.A. Perles and G.C. Shephard. Facets and nonfacets of convex polytopes. *Acta Math.*, 119:113–145, 1967.
- [Pfe02] J. Pfeifle. Work in progress. TU Berlin, 2002.
- [PB80] J.S. Provan and L.J. Billera. Decompositions of simplicial complexes related to diameters of convex polyhedra. *Math. Oper. Res.*, 5:576–594, 1980.
- [RSST97] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. The four-colour theorem. *J. Combin. Theory Ser. B*, 70:2–44, 1997.
- [Sch85] E. Schulte. The existence of nontiles and nonfacets in three dimensions. *J. Combin. Theory Ser. A*, 38:75–81, 1985.
- [Sch94] C. Schulz. Polyhedral manifolds on polytopes. In M.I. Stoka, editor, *First International Conference on Stochastic Geometry, Convex Bodies and Empirical Measures (Palermo, 1993)*. *Rend. Circ. Mat. Palermo (2) Suppl.*, 35:291–298, 1994.
- [ST01] D.A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, 2001, pages 296–305.
- [Ste22] E. Steinitz. Polyeder und Raumeinteilungen. In W.F. Meyer and H. Mohrmann, editors, *Encyklopädie der mathematischen Wissenschaften, Dritter Band: Geometrie, III.1.2., Heft 9, Kapitel IIIA B 12*, pages 1–139. Teubner, Leipzig, 1922.
- [SR34] E. Steinitz and H. Rademacher. *Vorlesungen über die Theorie der Polyeder*. Springer-Verlag, Berlin, 1934; reprint, Springer-Verlag, Berlin, 1976.
- [Tho81] C. Thomassen. Kuratowski’s theorem. *J. Graph Theory*, 5:225–241, 1981.
- [Tut56] W.T. Tutte. A theorem on planar graphs. *Trans. Amer. Math. Soc.*, 82:99–116, 1956.
- [Tut62] W.T. Tutte. A census of planar triangulations. *Canad. J. Math.*, 14:21–38, 1962.
- [vKa32] E.R. van Kampen. Komplexe in euklidischen Räumen. *Abh. Math. Sem. Hamburg*, 9:72–78, 1932. Berichtigung dazu, *ibid.*, 152–153.
- [Vin85] E.B. Vinberg. Hyperbolic groups of reflections (Russian). *Uspekhi Mat. Nauk*, 40:29–66, 255, 1985.
- [Whi84] W. Whiteley. Infinitesimally rigid polyhedra. I. Statics of frameworks. *Trans. Amer. Math. Soc.*, 285:431–465, 1984.
- [Whi32] H. Whitney. Non-separable and planar graphs. *Trans. Amer. Math. Soc.*, 34:339–362, 1932.
- [Wil88] K. Williamson Hoke. Completely unimodal numberings of a simple polytope. *Discrete Appl. Math.*, 20:69–81, 1988.
- [Wu65] W.-T. Wu. *A Theory of Imbedding, Immersion, and Isotopy of Polytopes in a Euclidean space*. Science Press, Beijing, 1965.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.
- [Zie98] G.M. Ziegler. Shelling polyhedral 3-balls and 4-polytopes. *Discrete Comput. Geom.*, 19:159–174, 1998.

21 POLYHEDRAL MAPS

Ulrich Brehm and Egon Schulte

INTRODUCTION

Historically, polyhedral maps on surfaces made their first appearance as convex polyhedra. The famous Kepler-Poinsot (star) polyhedra marked the first occurrence of maps on orientable surfaces of higher genus (namely 4), and started the branch of topology dealing with regular maps. Further impetus to the subject came from the theory of automorphic functions and from the Four-Color-Problem (Coxeter and Moser [CM80], Barnette [Bar83]).

A more systematic investigation of general polyhedral maps and nonconvex polyhedra began only around 1970, and was inspired by (the original edition) of Grünbaum's book "Convex Polytopes" [Grü67]. Since then, the subject has grown into an active field of research on the interfaces of convex and discrete geometry, graph theory, and combinatorial topology. The underlying topology is mainly elementary, and many basic concepts and constructions are inspired by convex polytope theory.

21.1 POLYHEDRA

Tessellations on surfaces are natural objects of study in topology that generalize convex polyhedra and plane tessellations. For general properties of convex polyhedra, polytopes, and tessellations, see Grünbaum [Grü67], Coxeter [Cox73], Grünbaum and Shephard [GS87], and Ziegler [Zie95], or [Chapters 3, 16, 17, 18, and 19](#) of this Handbook. For a survey on polyhedral manifolds see Brehm and Wills [BW93], which also has an extensive list of references. The long list of definitions that follows places polyhedral maps in the general context of topological and geometric complexes. For an account of 2- and 3-dimensional geometric topology, see Moise [Moi77].

GLOSSARY

Polyhedral complex: A finite set Γ of convex polytopes, the *faces* of Γ , in real n -space \mathbb{R}^n , such that two conditions are satisfied. First, if $Q \in \Gamma$ and F is a face of Q , then $F \in \Gamma$. Second, if $Q_1, Q_2 \in \Gamma$, then $Q_1 \cap Q_2$ is a face of Q_1 and Q_2 (possibly the empty face \emptyset). The subset $|\Gamma| := \bigcup_{Q \in \Gamma} Q$ of \mathbb{R}^n , equipped with the induced topology, is called the *underlying space* of Γ . The *dimension* $d := \dim \Gamma$ of Γ is the maximum of the dimensions (of the affine hulls) of the elements in Γ . We also call Γ a *polyhedral d-complex*. A face of Γ of dimension 0, 1, or i is a *vertex*, an *edge*, or an *i-face* of Γ , respectively. A face that is maximal (with respect to inclusion) is called a *facet* of Γ . (In our applications, the facets are just the d -faces of Γ .)

Face poset: The set $P(\Gamma)$ of all faces of Γ , partially ordered by inclusion. As a partially ordered set, $P(\Gamma) \cup \{||\Gamma||\}$ is a ranked lattice.

(Geometric) simplicial complex: A polyhedral complex Γ all of whose nonempty faces are simplices. An *abstract simplicial complex* Δ is a family of subsets of a finite set V , the *vertex set* of Δ , such that $\{x\} \in \Delta$ for all $x \in V$, and such that $F \subseteq G \in \Delta$ implies $F \in \Delta$. Each abstract simplicial complex Δ is isomorphic (as a poset ordered by inclusion) to the face poset of a geometric simplicial complex Γ . Once such an isomorphism is fixed, we set $||\Delta|| := ||\Gamma||$, and the terminology introduced for Γ carries over to Δ . (One often omits the qualifications “geometric” or “abstract.”)

Link: The link of a vertex x in a simplicial complex Γ is the subcomplex consisting of the faces that do not contain x of all the faces of Γ containing x .

Polyhedron: A subset P of \mathbb{R}^n such that $P = ||\Gamma||$ for some polyhedral complex Γ . In general, given P , there is no canonical way to associate with it the complex Γ . However, once Γ is specified, the terminology for Γ regarding $P(\Gamma)$ is also carried over to P .

Subdivision: If Γ_1 and Γ_2 are polyhedral complexes, Γ_1 is a subdivision of Γ_2 if $||\Gamma_1|| = ||\Gamma_2||$ and each face of Γ_1 is a subset of a face of Γ_2 . If Γ_1 is a simplicial complex, this is a *simplicial subdivision*.

Combinatorial d -manifold: For $d = 1$, this is a simplicial 1-complex Δ such that $||\Delta||$ is a 1-sphere. Inductively, if $d \geq 2$, it is a simplicial d -complex Δ such that $||\Delta||$ is a topological d -manifold (without boundary) and each vertex link is a *combinatorial $(d-1)$ -sphere* (that is, a combinatorial $(d-1)$ -manifold whose underlying space is a $(d-1)$ -sphere).

Polyhedral d -manifold: A polyhedral d -complex Γ having a simplicial subdivision that is a combinatorial d -manifold. If $d = 2$, this is simply a polyhedral 2-complex Γ for which $||\Gamma||$ is a compact 2-manifold (without boundary).

Triangulation: A triangulation (simplicial decomposition) of a topological space X is a simplicial complex Γ such that X and $||\Gamma||$ are homeomorphic.

Ball complex: A finite family \mathcal{C} of topological balls (homeomorphic images of Euclidean unit balls) in a Hausdorff space, the *underlying space* $||\mathcal{C}||$ of \mathcal{C} , whose relative interiors partition $||\mathcal{C}||$ in such a way that the boundary of each ball in \mathcal{C} is the union of other balls in \mathcal{C} . The *dimension* of \mathcal{C} is the maximum of the dimensions of the balls in \mathcal{C} .

Embedding: For a ball complex \mathcal{C} , a continuous mapping $\gamma : ||\mathcal{C}|| \hookrightarrow \mathbb{R}^n$ that is a homeomorphism of $||\mathcal{C}||$ onto its image. \mathcal{C} is said to be *embedded* in \mathbb{R}^n .

Polyhedral embedding: For a ball complex \mathcal{C} , an embedding γ that maps each ball in \mathcal{C} onto a convex polytope.

Immersion: For a ball complex \mathcal{C} , a continuous mapping $\gamma : ||\mathcal{C}|| \hookrightarrow \mathbb{R}^n$ that is locally injective (hence the image may have self-intersections). \mathcal{C} is said to be *immersed* in \mathbb{R}^n .

Polyhedral immersion: For a ball complex \mathcal{C} , an immersion γ that maps each ball in \mathcal{C} onto a convex polytope.

Map on a surface: An embedded finite graph M (without loops or multiple edges) on a compact 2-manifold (surface) S such that two conditions are satisfied: The closures of the connected components of $S \setminus M$, the *faces* of M , are closed

2-cells (closed topological disks), and each vertex of M has valency at least 3. (Note that some authors use a broader definition of maps; e.g., see [CM80].)

Polyhedral map: A map M on S such that the intersection of any two distinct faces is either empty, a common vertex, or a common edge.

Figure 21.1.1 shows a polyhedral map on a surface of genus 3, known as **Dyck's regular map**. We will discuss this map further in Sections 21.4 and 21.5.

Type: A map M on S is of type $\{p, q\}$ if all its faces are topological p -gons such that q meet at each vertex. The symbol $\{p, q\}$ is the **Schläfli symbol** for M .

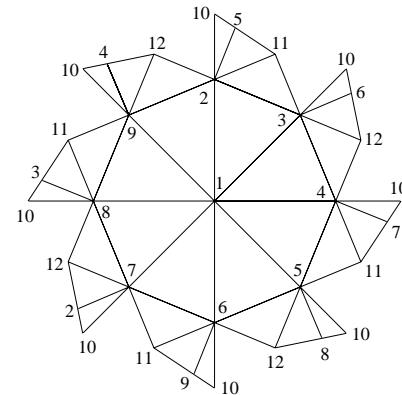


FIGURE 21.1.1

Dyck's regular map, of type $\{3, 8\}$.

Vertices with the same label are identified.

BASIC RESULTS

Simplicial complexes are important in topology, geometry, and combinatorics. Each abstract simplicial d -complex Δ with n vertices is isomorphic to the face poset of a geometric simplicial complex Γ in \mathbb{R}^{2d+1} that is obtained as the image under a projection (Schlegel diagram—see [Chapter 16](#)) of a simplicial d -subcomplex in the boundary complex of the cyclic convex $(2d+2)$ -polytope $C(n, 2d+2)$ with n vertices; see [Grü67] or Chapter 16 of this Handbook.

Let \mathcal{C} be a ball complex and $P(\mathcal{C})$ the associated poset (i.e., \mathcal{C} ordered by inclusion). Let $\Delta(P(\mathcal{C}))$ denote the **order complex** of $P(\mathcal{C})$; that is, the simplicial complex whose vertex set is \mathcal{C} and whose k -faces are the k -chains $x_0 < x_1 < \dots < x_k$ in $P(\mathcal{C})$. Then $||\mathcal{C}||$ and $||\Delta(P(\mathcal{C}))||$ are homeomorphic. This means that the poset $P(\mathcal{C})$ already carries complete topological information about $||\mathcal{C}||$. See [Bjö95] or [BW93], as well as [Chapter 18](#), for further information.

Each polyhedral d -complex is a d -dimensional ball complex. The set \mathcal{C} of vertices, edges, and faces of a map M on a 2-manifold S is a 2-dimensional ball complex. In particular, a map M is a polyhedral map if and only if the intersection of any two elements of \mathcal{C} is empty or an element of \mathcal{C} . A map is usually identified with its poset of vertices, edges, and faces, ordered by inclusion. If M is a polyhedral map, then this poset is a lattice when augmented by \emptyset and S as smallest and largest elements. The dual lattice (obtained by reversing the order) again gives a polyhedral map, the **dual map**, on the same 2-manifold S . Note that in the context of polyhedral maps, the qualification “polyhedral” does not mean that it can be realized as a polyhedral complex. However, a polyhedral 2-manifold can always be regarded as a polyhedral map.

An important problem is the following:

PROBLEM 21.1.1 *General Embeddability Problem*

When is a given finite poset isomorphic to the face poset of some polyhedral complex in a given space \mathbb{R}^n ? When can a ball complex be polyhedrally embedded or polyhedrally immersed in \mathbb{R}^n ?

These questions are different from the embeddability problems that are discussed in piecewise-linear topology, because simplicial subdivisions are excluded. A complete answer is available only for the face posets of spherical maps:

THEOREM 21.1.2 *Steinitz's Theorem*

Each polyhedral map M on the 2-sphere is isomorphic to the boundary complex of a convex 3-polytope. Equivalently, a finite graph is the edge graph of a convex 3-polytope if and only if it is planar and 3-connected (it has at least 4 vertices and the removal of any 2 vertices leaves a connected graph).

Very little is known about polyhedral embeddings of orientable polyhedral maps of positive genus g . There are some general necessary combinatorial conditions for the existence of polyhedral embeddings in n -space \mathbb{R}^n [BGH91]. Given a simplicial polyhedral map of genus g it is generally difficult to decide whether or not it admits a polyhedral embedding in 3-space \mathbb{R}^3 . For each $g \geq 6$, there are examples of simplicial polyhedral maps that cannot be embedded in \mathbb{R}^3 [BO00]. Each nonorientable closed surface can be immersed but not embedded in \mathbb{R}^3 . However, the Möbius strip and therefore each nonorientable surface can be triangulated in such a way that the resulting simplicial polyhedral map cannot be polyhedrally immersed in \mathbb{R}^3 [Br83]. On the other hand, each triangulation of the torus or the real projective plane \mathbb{RP}^2 can be polyhedrally embedded in \mathbb{R}^4 [BS95].

Another important type of problem asks for topological properties of the space of all polyhedral embeddings, or of all convex d -polytopes, with a given face lattice. This is the *realization space* for this lattice. Every convex 3-polytope has an open ball as its realization space. However, the realization spaces of convex 4-polytopes can be arbitrarily complicated; see the “Universality Theorem” by Richter-Gebert [Ric96] in Chapter 16 of this Handbook.

For further embeddability results in higher dimensions, as well as for a discussion of some related problems such as the polytopality problems and isotopy problems, see [Zie95, BLS⁺99, BW93]. For a computational approach to the embeddability problem in terms of oriented matroids, see Bokowski and Sturmfels [BS89], as well as [Chapter 6](#) of this Handbook. We shall revisit the embeddability problem in Sections 21.2 and 21.5 for interesting special classes of polyhedral maps.

Many interesting maps M on compact surfaces S have a Schläfli symbol $\{p, q\}$; for examples, see [Section 21.4](#). These maps can then be obtained from the regular tessellation $\{p, q\}$ of the 2-sphere, the Euclidean plane, or the hyperbolic plane by making identifications. Trivially, $qf_0 = 2f_1 = pf_2$. Also, if the Euler characteristic χ of S is negative and m denotes the number of flags (incident triples consisting of a vertex, an edge, and a face) of M , then

$$\chi = f_0 - f_1 + f_2 = \frac{m}{2} \left(\frac{1}{q} - \frac{1}{2} + \frac{1}{p} \right) \leq -\frac{m}{84}, \quad (21.1.1)$$

and equality holds on the right-hand side if and only if M is of type $\{3, 7\}$ or $\{7, 3\}$.

21.2 EXTREMAL PROPERTIES

There is a natural interest in polyhedral maps and polyhedra defined by certain minimality properties. For relations with the famous Map Color Theorem, which gives the minimum genus of a surface on which the complete graph K_n can be embedded, see Ringel [Rin74] and Barnette [Bar83]. See also Brehm and Wills [BW93].

GLOSSARY

f-vector: For a map M , the vector $f(M) = (f_0, f_1, f_2)$, where f_0, f_1, f_2 are the numbers of vertices, edges, and faces of M , respectively.

Weakly neighborly: A polyhedral map is weakly neighborly (a *wnp map*) if any two vertices lie in a common face.

Neighborly: A map is neighborly if any two vertices are joined by an edge.

Nonconvex vertex: A vertex x of a polyhedral 2-manifold M in \mathbb{R}^3 is a *convex vertex* if at least one of the two components into which M divides a small convex neighborhood of x in \mathbb{R}^3 is convex; otherwise, x is nonconvex.

Tight polyhedral 2-manifold: A polyhedral 2-manifold M embedded in \mathbb{R}^3 such that every hyperplane strictly supporting M locally at a point supports M globally.

BASIC RESULTS

THEOREM 21.2.1

Let M be a polyhedral map of Euler characteristic χ with f-vector (f_0, f_1, f_2) . Then

$$f_0 \geq \lceil (7 + \sqrt{49 - 24\chi})/2 \rceil. \quad (21.2.1)$$

Here, $\lceil t \rceil$ denotes the smallest integer greater than or equal to t . This lower bound is known as the **Heawood bound** and is an easy consequence of Euler's formula $f_0 - f_1 + f_2 = \chi$ ($= 2 - 2g$ if M is orientable of genus g).

THEOREM 21.2.2

Except for the nonorientable 2-manifolds with $\chi = 0$ (Klein bottle) or $\chi = -1$ and the orientable 2-manifold of genus $g = 2$ ($\chi = -2$), each 2-manifold admits a triangulation for which the lower bound (21.2.1) is attained.

This is closely related to the Map Color Theorem. The same lower bound (21.2.1) holds for the number f_2 of faces of M , since the dual of M is a polyhedral map with the same Euler characteristic and with f-vector (f_2, f_1, f_0) .

The exact minimum for the number f_1 of edges of a polyhedral map is known for only some manifolds. Let $E_+(\chi)$ or $E_-(\chi)$, respectively, denote the smallest number f_1 such that there is a polyhedral map with f_1 edges on the orientable 2-manifold, or on the nonorientable 2-manifold, respectively, of Euler characteristic χ . The known values of $E_+(\chi)$ and $E_-(\chi)$ are listed in [Table 21.2.1](#); undecided cases

are left blank. The polyhedral maps that attain the minimal values $E_+(2)$, $E_+(-8)$, $E_-(0)$, and $E_-(-6)$ are uniquely determined.

TABLE 21.2.1 The known values of $E_+(\chi)$ and $E_-(\chi)$.

χ	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8	-26
$E_+(\chi)$	6	—	18	—	27	—	33	—	38	—	40	78
$E_-(\chi)$	—	15	18	23	26	30	33	35	36	40	42	

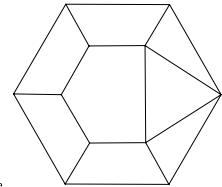


FIGURE 21.2.1

A self-dual polyhedral map on \mathbb{RP}^2 with the minimum number (15) of edges.

For a map on \mathbb{RP}^2 with 15 edges, see Figure 21.2.1. For the unique polyhedral map with 40 edges on the orientable 2-manifold of genus 5 ($\chi = -8$), see Figure 21.2.2 (and [Br90a]). This map is weakly neighborly and self-dual, and has a cyclic group of automorphisms acting regularly on the set of vertices and on the set of faces. Maps with the latter property are currently being investigated systematically.

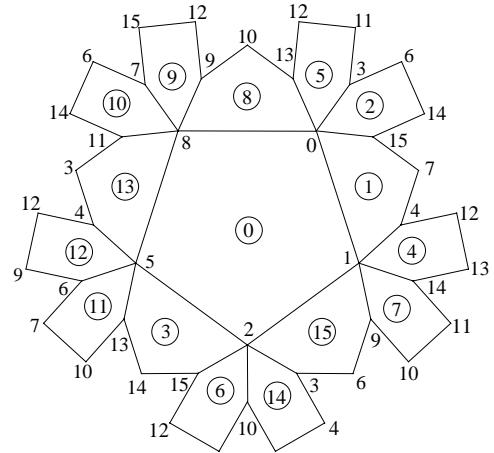


FIGURE 21.2.2

The unique polyhedral map of genus 5 with the minimum number (40) of edges.

A general bound for the number f_1 of edges is given by:

THEOREM 21.2.3 [Br90a]

$$f_1 \geq -\chi + \min\{y \in \mathbb{N} \mid y(\sqrt{2y} - 6) \geq -8\chi \text{ and } y \geq 8\},$$

where \mathbb{N} is the set of natural numbers.

If M is a polyhedral map on a surface S , then a new polyhedral map M' on S can be obtained from M by the following operation, called **face splitting**. A new edge xy is added across a face of M , where x and y are points on edges of M that are not contained in a common edge. The new vertices x and y of M' may be vertices of M , or one or both may be relative interior points of edges of M . The dual operation is called **vertex splitting**. On the sphere S^2 , the (boundary complex of the) tetrahedron is the only polyhedral map that is minimal with respect to face splitting. On the real projective plane \mathbb{RP}^2 , there are exactly 16 polyhedral maps that are minimal with respect to face splitting [Bar91], and exactly 7 that are minimal with respect to both face splitting and vertex splitting. These are exactly the polyhedral maps on \mathbb{RP}^2 with 15 edges, which is the minimum number of edges for \mathbb{RP}^2 . For an example, see Figure 21.2.1.

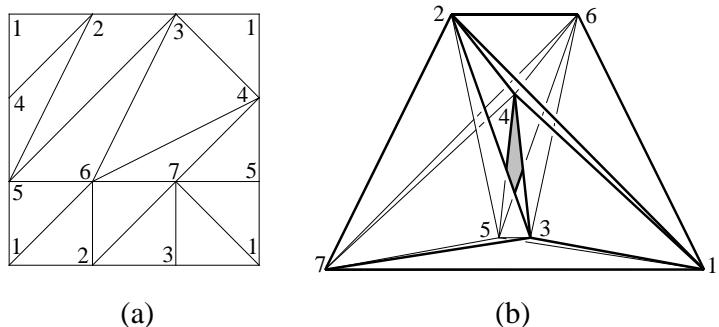
For neighborly polyhedral maps we always have equality in (21.2.1). Weakly neighborly polyhedral maps (wnp maps) are a generalization of neighborly polyhedral maps. On the 2-sphere, the only wnp maps are the (boundary complexes of the) pyramids and the triangular prism. Every other 2-manifold admits only finitely many combinatorially distinct wnp maps. Moreover,

$$\limsup_{\chi \rightarrow \infty} V_{\max}(\chi) \cdot |2\chi|^{-2/3} \leq 1,$$

where $V_{\max}(\chi)$ denotes the maximum number of vertices of a wnp map of Euler characteristic χ ; see [BA86], which also discusses further equalities and inequalities for general polyhedral maps. For several 2-manifolds, all wnp maps have been determined. For example, on the torus there are exactly five wnp maps, and three of them are geometrically realizable as polyhedra in \mathbb{R}^3 .

In some instances, the combinatorial lower bound (21.2.1) can also be attained geometrically by (necessarily orientable) polyhedra in \mathbb{R}^3 . Trivially, the tetrahedron minimizes $f_0 (= 4)$ for $g = 0$. For $g = 1$ there is a polyhedron with $f_0 = 7$ known as the **Császár torus**; see Figure 21.2.3. A pair of congruent copies of the torus shown in Figure 21.2.3b can be linked (if the coordinates orthogonal to the plane of projection are sufficiently small). Polyhedra that have the minimum number of vertices have also been found for $g = 2$ (the exceptional case), 3, or 4, with 10, 10, or 11 vertices, respectively.

FIGURE 21.2.3
(a) The unique 7-vertex triangulation of the torus and
(b) a symmetric realization as a polyhedron.



The minimum number of vertices for polyhedral maps that admit polyhedral immersions in \mathbb{R}^3 is 9 for both the real projective plane \mathbb{RP}^2 [Br90b] and the Klein bottle. The lower bound for \mathbb{RP}^2 follows directly from the fact that each immersion of \mathbb{RP}^2 in \mathbb{R}^3 has (generically) a triple point (like the classical Boy surface).

There are also some surprising results for higher genus. For example, for each $q \geq 3$ there exists a polyhedral map M_q of type $\{4, q\}$ with $f_0 = 2^q$ and $g = 2^{q-3}(q - 4) + 1$ such that M_q and its dual have polyhedral embeddings in \mathbb{R}^3 [MSW83]. These polyhedra are combinatorially regular in the sense of Section 21.5. Note that $f_0 = O(g/\log g)$. Thus for sufficiently large genus, M_q has more handles than vertices, and its dual has more handles than faces.

Every polyhedral 2-manifold in \mathbb{R}^3 of genus $g \geq 1$ contains at least 5 nonconvex vertices. This bound is attained for each $g \geq 1$. For tight polyhedral 2-manifolds, the lower bound for the number of nonconvex vertices is larger and depends on g . For a survey on tight polyhedral submanifolds see [Küh95].

21.3 EBERHARD'S THEOREM AND RELATED RESULTS

Eberhard's theorem is one of the oldest nontrivial results about convex polyhedra. The standard reference is Grünbaum [Grü67, Grü70]. For recent developments see also Jendrol [Jen93].

GLOSSARY

- p-sequence:*** For a polyhedral map M , the sequence $p(M) = (p_k(M))_{k \geq 3}$, where $p_k = p_k(M)$ is the number of k -gonal faces of M .
- v-sequence:*** For a polyhedral map M , the sequence $v(M) = (v_k(M))_{k \geq 3}$, where $v_k = v_k(M)$ is the number of vertices of M of degree k .

EBERHARD-TYPE RESULTS

Significant results are known for the general problem of determining what kind of polygons, and how many of each kind, may be combined to form the faces of a polyhedral map M on an orientable surface of genus g . These refine results (for $d = 3$) about the boundary complex and the number of i -dimensional faces ($i = 0, \dots, d - 1$) of a convex d -polytope [Grü67, Zie95]; see Chapter 18.

If M is a polyhedral map of genus g with f -vector (f_0, f_1, f_2) , then

$$\sum_{k \geq 3} p_k = f_2, \quad \sum_{k \geq 3} v_k = f_0, \quad \sum_{k \geq 3} kp_k = 2f_1 = \sum_{k \geq 3} kv_k. \quad (21.3.1)$$

Further, Euler's formula $f_0 - f_1 + f_2 = 2(1 - g)$ implies the equations

$$\sum_{k \geq 3} (6 - k)p_k + 2 \sum_{k \geq 3} (3 - k)v_k = 12(1 - g) \quad (21.3.2)$$

and

$$\sum_{k \geq 3} (4 - k)(p_k + v_k) = 8(1 - g). \quad (21.3.3)$$

These equations contain no information about p_6, v_3 and p_4, v_4 , respectively.

Eberhard-type results deal with the problem of determining which pairs $(p_k)_{k \geq 3}$ and $(v_k)_{k \geq 3}$ of sequences of nonnegative integers can occur as p -sequences $p(M)$ and v -sequences $v(M)$ of polyhedral maps M of a given genus g . The above equations yield simple necessary conditions. As a consequence of Steinitz's theorem (Section 21.1), the problem for $g = 0$ is equivalent to a similar such problem for convex 3-polytopes [Grü67, Grü70]. The classical theorem of Eberhard says the following:

THEOREM 21.3.1 *Eberhard's Theorem*

For each sequence $(p_k \mid 3 \leq k \neq 6)$ of nonnegative integers satisfying

$$\sum_{k \geq 3} (6 - k)p_k = 12,$$

there exist values of p_6 such that the sequence $(p_k)_{k \geq 3}$ is the p -sequence of a spherical polyhedral map all of whose vertices have degree 3, or, equivalently, of a convex 3-polytope that is simple (has vertices only of degree 3).

This is the case $g = 0$ and $v_3 = f_0$, $v_k = 0$ ($k \geq 4$).

More general results have been established [Jen93]. Given two sequences $p' = (p_k \mid 3 \leq k \neq 6)$ and $v' = (v_k \mid k > 3)$ of nonnegative integers such that the equation (21.3.2) is satisfied for a given genus g , let $E(p', v'; g)$ denote the set of integers $p_6 \geq 0$ such that $(p_k)_{k \geq 3}$ and $(v_k)_{k \geq 3}$, with $v_3 := (\sum_{k \geq 3} kp_k - \sum_{k \geq 4} kv_k)/3$ determined by (21.3.1), are the p -sequences and v -sequences, respectively, of a polyhedral map of genus g . For all but two admissible triples (p', v', g) , the set $E(p', v'; g)$ is known up to a finite number of elements. For example, for $g = 0$, the set $E(p', v'; 0)$ is nonempty if and only if $\sum_{k \not\equiv 0 \pmod{3}} v_k \neq 1$ or $p_k \neq 0$ for at least one odd k . In particular, for each such nonempty set, there exists a constant c depending on (p', v') such that $E(p', v'; 0) = \{j \mid c \leq j\}$, $\{j \mid c \leq j \equiv 0 \pmod{2}\}$, or $\{j \mid c \leq j \equiv 1 \pmod{2}\}$. Similarly, for each triple with $g \geq 2$, there is a constant c depending on (p', v', g) such that $E(p', v'; g) = \{j \mid c \leq j\}$. There are analogous results for sequences $(p_k \mid 3 \leq k \neq 4)$ and $(v_k \mid 3 \leq k \neq 4)$ that satisfy the equation (21.3.3) or other related equations.

For $g = 1$ there is also a more geometric Eberhard-type result available, which requires the polyhedral map M to be polyhedrally embedded in \mathbb{R}^3 :

THEOREM 21.3.2 [Gri83]

Let s , p_k ($k \geq 3, k \neq 6$) be nonnegative integers. Then there exists a toroidal polyhedral 2-manifold M in \mathbb{R}^3 with $p_k(M) = p_k$ ($k \neq 6$) and $\sum_{k \geq 3} (k - 3)v_k(M) = s$ if and only if $\sum_{k \geq 3} (6 - k)p_k = 2s$ and $s \geq 6$.

Also, for toroidal polyhedral 2-manifolds in \mathbb{R}^3 (as well as for convex 3-polytopes), the exact range of possible f -vectors is known [Grü67, BW93].

THEOREM 21.3.3

A polyhedral embedding in \mathbb{R}^3 of some torus with f -vector (f_0, f_1, f_2) exists if and only if $f_0 - f_1 + f_2 = 0$, $f_2(11 - f_2)/2 \leq f_0 \leq 2f_2$, $f_0(11 - f_0)/2 \leq f_2 \leq 2f_0$, and $2f_1 - 3f_0 \geq 6$.

For generalizations of Eberhard's theorem to tilings of the Euclidean plane, see also [GS87].

21.4 REGULAR MAPS

Regular maps are topological analogues of the ordinary regular polyhedra and star-polyhedra on surfaces. Historically they became important in the context of transformations of algebraic equations and representations of algebraic curves in homogeneous complex variables. There is a large body of literature on regular maps and their groups. The classical text is Coxeter and Moser [CM80]. For a recent text see McMullen and Schulte [MS02].

GLOSSARY

(Combinatorial) automorphism: An incidence-preserving bijection (of the set of vertices, edges, and faces) of a map M on a surface S to itself. The (**combinatorial automorphism**) **group** $A(M)$ of M is the group of all such bijections. It can be “realized” by a group of homeomorphisms of S .

Regular map: A map M on S whose group $A(M)$ is transitive on the flags (incident triples consisting of a vertex, an edge, and a face) of M .

GENERAL RESULTS

Each regular map M is of type $\{p, q\}$ for some finite p and q . Its group $A(M)$ is transitive on the vertices, the edges, and the faces of M . In general, the Schläfli symbol $\{p, q\}$ does not determine M uniquely. The group $A(M)$ is generated by involutions ρ_0, ρ_1, ρ_2 such that the **standard relations**

$$\rho_0^2 = \rho_1^2 = \rho_2^2 = (\rho_0\rho_1)^p = (\rho_1\rho_2)^q = (\rho_0\rho_2)^2 = 1$$

hold, but in general there are also further independent relations. Any triangle in the “barycentric subdivision” (order complex) of M is a fundamental region for $A(M)$ on the underlying surface S ; see [Section 21.1](#). For any fixed such triangle, we can take for ρ_i the “combinatorial reflection” in its side opposite to the vertex that corresponds to an i -dimensional element of M . The set of standard relations gives a presentation for the symmetry group of the regular tessellation $\{p, q\}$ on the 2-sphere, in the Euclidean plane, or in the hyperbolic plane, whichever is the universal covering of M . See [Figure 21.5.1](#) (a) for a conformal (hyperbolic) drawing of the Dyck map (shown also in [Figure 21.1.1](#)) with a fundamental region shaded. The identifications on the boundary of the drawing are indicated by letters.

For orientable surfaces S , the regular maps are known for genus $g \leq 6$. Up to isomorphism, if $g = 0$, there are just the Platonic solids (or regular spherical tessellations) $\{3, 3\}$, $\{3, 4\}$, $\{4, 3\}$, $\{3, 5\}$, and $\{5, 3\}$. For $g = 1$, there are three infinite families of torus maps of type $\{3, 6\}$, $\{6, 3\}$, and $\{4, 4\}$, each a quotient of the corresponding Euclidean universal covering tessellations $\{3, 6\}$, $\{6, 3\}$, and $\{4, 4\}$, respectively. For $g \geq 2$, the universal covering tessellation $\{p, q\}$ is hyperbolic and there are only finitely many regular maps on a surface of genus g . The latter follows from the **Hurwitz formula** $|A(M)| \leq 84|\chi|$ (or from the inequality 21.1.1), where χ is the Euler characteristic of S . Each regular map on a nonorientable surface is

doubly covered by a regular map of the same type on an orientable surface, and this covering map is unique [Wil78].

Generally speaking, given M , the topology of S is reflected in the relations that have to be added to the standard relations to obtain a presentation for $A(M)$. Conversely, many interesting regular maps can be constructed by adding certain kinds of extra relations for the group. Two examples are the regular maps $\{p, q\}_r$ and $\{p, q|r\}$ obtained by adding the extra relations $(\rho_0\rho_1\rho_2)^r = 1$ or $(\rho_0\rho_1\rho_2\rho_1)^r = 1$, respectively. Often these are “infinite maps” on noncompact surfaces, but there are also many (finite) maps on compact surfaces. The Dyck map $\{3, 8\}_6$ and the famous **Klein map** $\{3, 7\}_8$ (with group $PGL(2, 7)$) are both of genus 3 and of the first kind, while the so-called regular skew polyhedra in Euclidean 3-space or 4-space are of the second kind. For more details and further interesting classes of regular maps, see [CM80, MS02] and Chapter 19 of this Handbook. In Section 21.5 we shall discuss polyhedral embeddings of regular maps in ordinary 3-space.

The rotation subgroup (orientation preserving subgroup) of the group of an orientable regular map (of type $\{3, 7\}$ or $\{7, 3\}$) that achieves equality in the Hurwitz formula is also called a **Hurwitz group**. The Klein map is the regular map of smallest genus whose rotation subgroup is a Hurwitz group [Con90].

21.5 SYMMETRIC POLYHEDRA

Traditionally, much of the appeal of polyhedral 2-manifolds comes from their combinatorial or geometric symmetry properties. For surveys on symmetric polyhedra in \mathbb{R}^3 see Schulte and Wills [SW91], Bokowski and Wills [BW88], and Brehm and Wills [BW93].

GLOSSARY

Combinatorially regular: A polyhedral 2-manifold (or polyhedron) P is combinatorially regular if its combinatorial automorphism group $A(P)$ is flag-transitive (or, equivalently, if the underlying polyhedral map is a regular map).

Equivelar: A polyhedral 2-manifold (or polyhedron) P is equivelar of type $\{p, q\}$ if all its 2-faces are convex p -gons and all its vertices are q -valent.

GENERAL RESULTS

See Section 21.4 for results about regular maps. Up to isomorphism, the Platonic solids are the only combinatorially regular polyhedra of genus 0. For the torus, each regular map that is a polyhedral map also admits an embedding in \mathbb{R}^3 as a combinatorially regular polyhedron. Much less is known for maps of genus $g \geq 2$. Two infinite sequences of combinatorially regular polyhedra have been discovered, one consisting of polyhedra of type $\{4, q\}$ ($q \geq 3$) and the other of their duals of type $\{q, 4\}$. These are polyhedral embeddings of the maps M_q and their duals mentioned in Section 21.2. Several famous regular maps have also been realized as polyhedra, including Klein’s $\{3, 7\}_8$, Dyck’s $\{3, 8\}_6$, and Coxeter’s $\{4, 6|3\}$, $\{6, 4|3\}$, $\{4, 8|3\}$, and $\{8, 4|3\}$ [SW85, SW91, BS89]. However, a complete classification of

combinatorially regular polyhedra is not within reach at present. See Figure 21.5.1 for an illustration of a polyhedral realization of Dyck's regular map $\{3, 8\}_6$ shown in [Figure 21.1.1](#). (a) shows a conformal drawing of the Dyck map, with a fundamental region shaded, while (b) shows a maximally symmetric polyhedral realization.

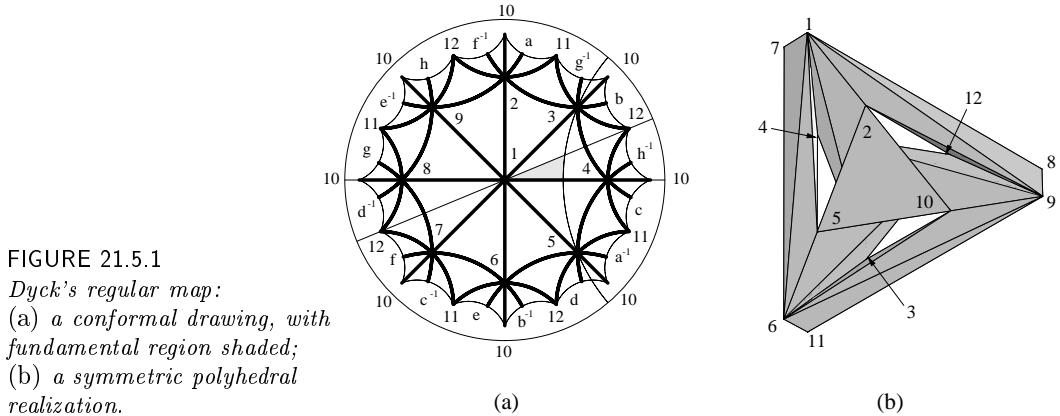


FIGURE 21.5.1

Dyck's regular map:
 (a) a conformal drawing, with
 fundamental region shaded;
 (b) a symmetric polyhedral
 realization.

(a)

(b)

For a more general concept of polyhedra in \mathbb{R}^3 or higher-dimensional spaces, as well as an enumeration of the corresponding regular polyhedra, see [Chapter 19](#) of this Handbook. The latter also contains a depiction of the polyhedral realization of $\{4, 8|3\}$.

Equivelarity is a local regularity condition. Each combinatorially regular polyhedron in \mathbb{R}^3 is equivellar. However, there are many other equivellar polyhedra. For sufficiently large genus g , for example, there are equivellar polyhedra for each of the types $\{3, q\}$ with $q = 7, 8, 9$; $\{4, q\}$ with $q = 5, 6$; and $\{q, 4\}$ with $q = 5, 6$ [BW93].

The symmetry group of a polyhedron can be much smaller than the combinatorial automorphism group of the underlying polyhedral map. In particular, the five Platonic solids are the only polyhedra in \mathbb{R}^3 with a flag-transitive symmetry group. However, even for higher genus (namely, for $g = 1, 3, 5, 7, 11$, and 19), polyhedra with a vertex-transitive symmetry group are known. Such a polyhedral torus is shown in Figure 21.5.2.

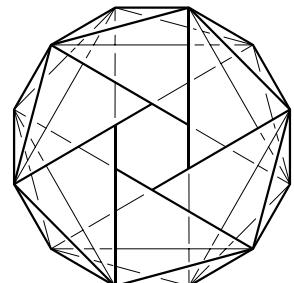


FIGURE 21.5.2

A vertex-transitive polyhedral torus.

Finally, if we relax the requirement that a polyhedron be free of self-intersections and allow more general “polyhedral realizations” of maps (for instance, polyhedral

immersions), then there is much more flexibility in the construction of “polyhedra” with high symmetry properties. The most famous examples are the Kepler-Poinsot star-polyhedra, but there are also many others. For more details see [SW91, BW88, BW93, MS02] and [Chapter 19](#) of this Handbook.

21.6 SOURCES AND RELATED MATERIAL

SURVEYS

- [Bar83]: A text about colorings of maps and polyhedra.
- [Bjö95]: A survey on topological methods in combinatorics.
- [BLS⁺99]: A monograph on oriented matroids.
- [BS89]: A text about computational aspects of geometric realizability.
- [BW93]: A survey on polyhedral manifolds in 2 and higher dimensions.
- [Con90]: A survey on Hurwitz groups.
- [Cox73]: A monograph on regular polytopes, regular tessellations, and reflection groups.
- [CM80]: A monograph on discrete groups and their presentations.
- [GT87]: A text about maps on surfaces.
- [Grü67]: A monograph on convex polytopes. The second edition is a reprint of the original one, updated with extensive notes about recent developments.
- [Grü70]: A survey on convex polytopes complementing the exposition in the original (1967) edition of [Grü67].
- [GS87]: A monograph on plane tilings and patterns.
- [Küh95]: A survey on tight polyhedral manifolds.
- [Moi77]: A text about geometric topology in low dimensions.
- [MS02]: A monograph on abstract regular polytopes and their groups.
- [Rin74]: A text about maps on surfaces and the Map Color Theorem.
- [SW91]: A survey on combinatorially regular polyhedra in 3-space.
- [Zie95]: A graduate textbook on convex polytopes.

RELATED CHAPTERS

- [Chapter 3: Tilings](#)
- [Chapter 6: Oriented matroids](#)
- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 17: Subdivisions and triangulations of polytopes](#)
- [Chapter 18: Face numbers of polytopes and complexes](#)
- [Chapter 19: Symmetry of polytopes and polyhedra](#)

REFERENCES

- [Bar83] D.W. Barnette. *Map Coloring, Polyhedra, and the Four-Color Problem*. Math. Assoc. America, Washington, 1983.
- [Bar91] D.W. Barnette. The minimal projective plane polyhedral maps. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, pages 63–70, volume 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, Amer. Math. Soc., Providence, 1991.
- [BGH91] D.W. Barnette, P. Gritzmann, and R. Höhne. On valences of polyhedra. *J. Combin. Theory Ser. A*, 58:279–300, 1991.
- [Bjö95] A. Björner. Topological methods. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, pages 1819–1872. Elsevier, Amsterdam, 1995.
- [BLS⁺99] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G.M. Ziegler. *Oriented Matroids*. Volume 46 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993; second ed. 1999.
- [BO00] J. Bokowski and A.G. de Oliveira. On the generation of oriented matroids. *Discrete Comput. Geom.*, 24:197–208, 2000.
- [BS89] J. Bokowski and B. Sturmfels. *Computational Synthetic Geometry*. Volume 1355 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1989.
- [BW88] J. Bokowski and J.M. Wills. Regular polyhedra with hidden symmetries. *Math. Intelligencer*, 10:27–32, 1988.
- [Br83] U. Brehm. A nonpolyhedral triangulated Möbius strip. *Proc. Amer. Math. Soc.*, 89:519–522, 1983.
- [Br90a] U. Brehm. Polyhedral maps with few edges. In R. Bodendiek and R. Henn, editors, *Topics in Combinatorics and Graph Theory*, pages 153–162. Physica Verlag, Heidelberg, 1990.
- [Br90b] U. Brehm. How to build minimal polyhedral models of the Boy surface. *Math. Intelligencer*, 12:51–56, 1990.
- [BA86] U. Brehm and A. Altshuler. On weakly neighborly polyhedral maps of arbitrary genus. *Israel J. Math.*, 53:137–157, 1986.
- [BS95] U. Brehm and G. Schild. Realizability of the torus and the projective plane in R^4 . *Israel J. Math.*, 91:249–251, 1995.
- [BW93] U. Brehm and J.M. Wills. Polyhedral manifolds. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 535–554. North-Holland, Amsterdam, 1993.
- [Con90] M. Conder. Hurwitz groups: A brief survey. *Bull. Amer. Math. Soc.*, 23:359–370, 1990.
- [Cox73] H.S.M. Coxeter. *Regular Polytopes* (3rd edition). Dover, New York, 1973.
- [CM80] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups* (4th edition). Springer-Verlag, Berlin, 1980.
- [Gri83] P. Gritzmann. The toroidal analogue of Eberhard’s theorem. *Mathematika*, 30:274–290, 1983.
- [GT87] J.L. Gross and T.W. Tucker. *Topological Graph Theory*. Wiley, New York, 1987.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Interscience, London, 1967; second edition edited by V. Kaibel, V. Klee, and G.M. Ziegler, volume 221 of *Graduate Texts in Math.*, Springer-Verlag, New York, 2003.

- [Grü70] B. Grünbaum. Polytopes, graphs, and complexes. *Bull. Amer. Math. Soc.*, 76:1131–1201, 1970.
- [GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns*. Freeman, New York, 1987.
- [Jen93] S. Jendrol. On face-vectors and vertex-vectors of polyhedral maps on orientable 2-manifolds. *Math. Slovaca*, 43:393–416, 1993.
- [Küh95] W. Kühnel. *Tight Polyhedral Submanifolds and Tight Triangulations*. Volume 1612 of *Lecture Notes in Math.*, Springer-Verlag, New York, 1995.
- [Moi77] E.E. Moise. *Geometric Topology in Dimensions 2 and 3*. Volume 47 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1977.
- [MSW83] P. McMullen, Ch. Schulz, and J.M. Wills. Polyhedral manifolds in E^3 with unusually large genus. *Israel J. Math.*, 46:127–144, 1983.
- [MS02] P. McMullen and E. Schulte. *Abstract Regular Polytopes*. Volume 92 of *Encyclopedia Math. Appl.*, Cambridge University Press, 2002.
- [Ric96] J. Richter-Gebert. *Realization Spaces of Polytopes*. Volume 1643 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1996.
- [Rin74] G. Ringel. *Map Color Theorem*. Springer-Verlag, Berlin, 1974.
- [SW85] E. Schulte and J.M. Wills. A polyhedral realization of Felix Klein’s map $\{3, 7\}_8$ on a Riemann surface of genus 3. *J. London Math. Soc.*, 32:539–547, 1985.
- [SW91] E. Schulte and J.M. Wills. Combinatorially regular polyhedra in three-space. In K.H. Hofmann and R. Wille, editors, *Symmetry of Discrete Mathematical Structures and Their Symmetry Groups*, pages 49–88. Heldermann Verlag, Berlin, 1991.
- [Wil78] S.E. Wilson. Non-orientable regular maps. *Ars Combin.*, 5:213–218, 1978.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

22 CONVEX HULL COMPUTATIONS

Raimund Seidel

INTRODUCTION

The “convex hull problem” is a catch-all phrase for computing various descriptions of a polytope that is either specified as the convex hull of a finite point set in \mathbb{R}^d or as the intersection of a finite number of halfspaces. We first define the various problems and discuss their mutual relationships (Section 22.1). We discuss the very special case of the irredundancy problem in Section 22.2. We consider general dimension d in Section 22.3 and describe the most common general algorithmic approaches along with the best run-time bounds achieved so far. In Section 22.4 we consider separately the case of small dimensions $d = 2, 3, 4, 5$. Finally, Section 22.5 addresses various issues related to the convex hull problem.

22.1 DESCRIBING CONVEX POLYTOPES AND POLYHEDRA

“Computing the convex hull” is a phrase whose meaning varies with the context. Consequently there has been confusion regarding the applicability and efficiency of various “convex hull algorithms.” We therefore first discuss the different versions of the “convex hull problem” along with versions of the “halfspace intersection problem” and how they are related via polarity.

CONVEX HULLS

The generic convex hull problem can be stated as follows: Given a finite set $S \subset \mathbb{R}^d$, compute a description of $P = \text{conv}S$, the **polytope** formed by the convex hull of S .

A convex polytope P can be described in many ways. In our context the most important descriptions are those listed below.

GLOSSARY

(See [Chapter 16](#) for basic concepts and results of polytope theory.)

Vertex description: The set of all vertices of P (specified by their coordinates).

Facet description: The set of all facets of P (specified by their defining linear inequalities).

Double description: The set of vertices of P , the set of facets of P , and the incidence relation between the vertices and the facets (specified by an incidence matrix).

Lattice description: The face lattice of P (specified by its *Hasse diagram* (cf.

below), with vertex and facet nodes augmented by coordinates and defining linear inequalities, respectively).

Boundary description: A triangulation of the boundary of P (specified by a simplicial complex, with vertices and maximal simplices augmented by coordinates and defining normalized linear inequalities, respectively).

Hasse diagram: A directed graph of an order relation that joins nodes a to b iff $a \leq b$ and there are no elements between a and b in the sense that if $a \leq c \leq b$ then either $c = a$ or $c = b$. For the face lattice the order relation is containment.

The five descriptions above assume that P is full-dimensional. If it is not, then a specification of the smallest affine subspace containing P has to be added to all but the vertex description.

These five descriptions make explicit to varying degrees the geometric information carried by polytope P and the combinatorial information of its facial structure. The vertex description and the facet description each carry only rudimentary geometric information about P . We therefore call them *purely geometric descriptions*. The other three descriptions we call *combinatorial* since they also carry more or less complete combinatorial information about the face structure of P . As a matter of fact, these three descriptions are equivalent in the sense that one can be computed from the other by purely combinatorial means, i.e., without the use of arithmetic operations on real numbers.

Which description is to be computed depends on the application at hand. It is important to keep in mind, however, that these descriptions can differ drastically in terms of their sizes (see [Section 22.3](#)).

INTERSECTION OF HALFSPACES

Closely related to the convex hull problem is the *halfspace intersection* problem: Given a finite set H of halfspaces in \mathbb{R}^d , compute a description of the *polyhedron* $Q = \bigcap H$.

Convex polyhedra are more general objects than convex polytopes in that they need not be bounded. Consequently their descriptions are slightly more complicated. Every polyhedron Q admits a “factorization” $Q = L + C + R$, where L is a linear subspace orthogonal to C and R , the set C is a convex cone, and R is a convex polytope. The “vertex description” of Q then consists of a minimal set of vectors spanning L , the set of extreme rays of C , and the set of vertices of R . Our other four description methods for convex polytopes have to be adjusted accordingly in order to apply to polyhedra. Also, the triangulations appearing in the boundary description need to allow for unbounded simplices (this concept makes sense if one views a k -simplex as an intersection of $k+1$ halfspaces).

Because polyhedra are more general than polytopes, all statements about the size differences among the various descriptions of the latter apply also to the former.

POLARITY

The relationship between computing convex hulls and computing the intersection of halfspaces arises because of *polarity* (Section 16.1.2). Let S be a finite set in \mathbb{R}^d and let H_S be the set of halfspaces $\{h_p \mid p \in S\}$, with $h_p = \{x \mid \langle x, p \rangle \leq 1\}$. Let

$P = \text{conv}S$ and let $Q = \bigcap H_S$. Polarity yields a 1-1 correspondence between the k -faces of Q and the $(d-k)$ -faces of P that admit supporting hyperplanes having P and the origin strictly on the same side. In particular, if the origin is contained in the relative interior of P , then the face lattices of P and Q are anti-isomorphic.

It is thus easy to reduce a convex hull problem to a halfspace intersection problem: First translate S by $-\sum_{p \in S} p / |S|$ to insure that the origin is contained in the relative interior of P , and compute $Q = \bigcap H_S$ for the resulting H_S . The polytope Q is then the polar P^Δ of P , and, assuming that P is full-dimensional, we have straightforward correspondences between the vertex description of Q and the facet description of P , between the facet description of Q and the vertex description of P , between the double descriptions of Q and of P (reverse the roles of vertices and facets), and between the lattice descriptions of Q and P (reverse the order of the lattice). Note that there is *no* correspondence between the boundary descriptions. If P has dimension $l < d$ then $Q = Q' \times L$, where polytope Q' has dimension l and L is a linear subspace of dimension $d-l$. The indicated correspondences then hold between P and Q' .

Reducing a halfspace intersection problem to a convex hull problem is more difficult. Polarity assumes all halfspaces to be describable as $\{x \mid \langle a, x \rangle \leq 1\}$, which means they must strictly contain the origin. In general not all halfspaces in a set H will be of such a form. In order to achieve this form the origin must be translated to a point r that is contained in the interior of $Q = \bigcap H$. Determining such a point r requires solving a linear program. Moreover, such an r does not exist if Q is empty, in which case the halfspace intersection problem has a trivial solution, or if Q is not full-dimensional, in which case one has to perform some sort of dimension reduction.

In general, halfspace intersection appears to be a slightly more general and versatile problem, especially in a homogenized formulation, which very elegantly avoids various special cases (see, e.g., [MRTT53]). Nevertheless, we will concentrate exclusively on the convex hull problem. The stated results can be translated *mutatis mutandis* to the halfspace intersection problem. In many cases the algorithms can be “dualized” to apply directly to the halfspace intersection problem, or the algorithms were originally stated for the halfspace intersection problem and were “dualized” to the convex hull problem.

22.2 THE IRREDUNDANCY PROBLEM

GLOSSARY

Irredundancy problem: Given a set S of n points in \mathbb{R}^d , compute the vertex description of $P = \text{conv}S$.

$\lambda(n, d)$: The time to solve a linear programming problem in d variables with n constraints. $O(n)$ for fixed d (see [Chapter 45](#)).

This problem seeks to compute all points in S that are irredundant, in the sense that they cannot be represented as a convex combination of the remaining points in S . The equivalent polar formulation requires computation of the facet

description of $Q = \bigcap H$, given a set H of n halfspaces in \mathbb{R}^d . We will follow the primal formulation.

The flavor of this version of the convex hull problem is very different from the other versions. Testing whether a point $p \in S$ is irredundant amounts to solving a linear programming problem in d variables with $n - 1$ constraints. The straightforward method of successively testing points for irredundancy results in an algorithm with running time $O(n\lambda(n - 1, d))$, which for fixed dimension d is $O(n^2)$.

Clarkson [Cla94] and independently Ottmann et al. [OSS95] have ingeniously improved this method so that every linear program involves only at most V constraints, where V is the number of vertices of P , i.e., the output size. The resulting running time is $O(n\lambda(V, d))$, which for fixed d is $O(nV)$.

In each of these two methods the n linear programs that occur are closely related to each other. This can be exploited, at least theoretically, by using data structures for so-called linear programming queries [Mat93, Cha96a, Ram00]. This was first done by Matoušek for the naive method [Mat93], and then by Chan for the improved method [Cha96], resulting for fixed $d > 3$ in an asymptotic time bound of

$$O(n \log^{d+2} V + (nV)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n).$$

Finally, note that for the small-dimensional case $d = 2, 3$ there are even algorithms with running time $O(n \log V)$ (see [Chapter 38](#)), which can be shown to be asymptotically worst-case optimal [KS86].

22.3 COMPUTING COMBINATORIAL DESCRIPTIONS

GLOSSARY

Facet enumeration problem: Compute the facet description of $P = \text{conv}S$, given S .

Vertex enumeration problem: Compute the vertex description of $Q = \bigcap H$, given H .

The facet and vertex enumeration problems are classical and were already considered as early as 1824 by Fourier (see [Sch86, pp. 209–225] for a survey). Interestingly, no *efficient* algorithm is known that solves these enumeration problems without also computing, besides the desired purely geometric description, some combinatorial description of the polyhedron involved. Consequently we now concentrate on computing combinatorial descriptions.

THE SIZES OF COMBINATORIAL DESCRIPTIONS

It is important to understand how the three combinatorial descriptions differ in terms of their sizes. Let S be a set of n points in \mathbb{R}^d and let $P = \text{conv}S$. Assume that P is a d -polytope and that it has m facets. As a consequence of McMullen’s Upper Bound Theorem ([Chapter 16](#)) and of polarity, the following inequalities hold

between n and m and are tight:

$$n \leq \mu(d, m) \quad \text{and} \quad m \leq \mu(d, n),$$

where

$$\mu(d, x) = f_{d-1}(C_d(x)) = \binom{x - \lceil d/2 \rceil}{\lfloor d/2 \rfloor} + \binom{x - 1 - \lceil (d-1)/2 \rceil}{\lfloor (d-1)/2 \rfloor},$$

which is $\Theta(x^{\lfloor d/2 \rfloor})$ for fixed d .

For the sake of definiteness let us define the sizes of the various descriptions as follows. For the double description of P it is the number of vertex-facet incidences, for the lattice description it is the total number of faces (of all dimensions) of P , and for the boundary description it is the number of $(d-1)$ -simplices in the boundary triangulation.

Note that for the double and the lattice descriptions the sizes are completely determined by P , whereas the size of a boundary description depends on the boundary triangulation that is actually used. The sizes of those triangulations for a given P can vary quite drastically, even if, as we assume from now on, all vertices of the triangulation must be from S .

These size measures are only crude approximations of the space required to store such descriptions in memory (in particular, in case of the lattice description the edges of the Hasse diagram are completely ignored). However, these approximations suffice to convey the possible similarities and differences between the sizes of the different descriptions.

For such a comparison between the description sizes of $P = \text{conv}S$ consider Table 22.3.1, whose columns deal with three cases. The first column lists worst-case upper bounds in terms of n and d . The second columns lists upper bounds in terms of m and d under the assumption that S is in nondegenerate position, i.e., no $d+1$ points in S lie in a common hyperplane, which means that P must be simplicial. Note that in this case there is a unique boundary description. Finally, the third column lists asymptotic bounds (d fixed) for *products of cyclic polytopes* $CC_d(n)$, a certain class of highly degenerate polytopes described in [ABS97]. (See Section 13.1 for a discussion of cyclic polytopes.) In this third table column, $\delta = \lfloor \sqrt{d/2} \rfloor$

TABLE 22.3.1 Polytope description sizes.

DESCRIPTION	WORST CASE	NONDEGENERATE	DEGENERATE CLASS $CC_d(n)$
Double	$d \cdot \mu(d, n)$	$d \cdot m$	$\Theta(n \cdot m^{1-1/\delta})$
Lattice	$2^d \cdot \mu(d, n)$	$2^d \cdot m$	$\Theta((n+m)^\delta)$
Boundary	$\mu(d, n)$	m	$\Omega((n+m)^\delta)$

The bounds in the table are based on the fact that all description sizes are maximized when P is a cyclic polytope, that each facet of a simplicial d -polytope contains 2^d faces, and that the Upper Bound Theorem also applies to simplicial spheres. The lower bound on the size of the boundary description of $CC_d(n)$ applies no matter which triangulation of the boundary is actually used.

The implication of this table is that in the worst case and also in the nondegenerate case all three combinatorial descriptions of P have approximately the same size. If d is considered constant, then the sizes are $\Theta(n^{\lfloor d/2 \rfloor})$ in the worst case, where n is the number of points in S (i.e., n is the input size), and the description sizes are $\Theta(m)$ in the nondegenerate case, where m is the number of facets of P (in a way the output size). The third column of the table, however, shows that in the general case the double description of a polytope P may be substantially more compact than the lattice description or the boundary description.

MAIN RESULTS AND OPEN PROBLEMS

The main positive results are that in the sense of asymptotic worst case complexity the convex hull problem has been solved completely, and that in the case of nondegenerate input, each of the three combinatorial descriptions can be found in time polynomial in the size of the input and the size of the output. In the case of general input this has only been shown for the lattice and for a boundary description, whereas it is unknown whether this is also possible for the double description.

In the following let $P = \text{conv}S$ be a d -polytope, and $|S| = n$.

THEOREM 22.3.1 Chazelle [Cha93]

If the dimension d is considered constant, then given S , each of the three combinatorial descriptions of $P = \text{conv}S$ can be computed in time $O(n \log n + n^{\lfloor d/2 \rfloor})$ using space $O(n^{\lfloor d/2 \rfloor})$. This is asymptotically worst-case optimal.

THEOREM 22.3.2 Avis-Fukuda [AF92]

Given S , a boundary description of $P = \text{conv}S$ can be computed in time $O(dnM)$ using space $O(dn)$, where M is the size of the boundary description produced.

If S is nondegenerate, then each of the three combinatorial descriptions of P can be computed in time $O(d^{O(1)}nM)$, where M is the size of the respective description.

THEOREM 22.3.3 Swart [Swa85] and Chand-Kapur [CK70]

Given S , the lattice description of $P = \text{conv}S$ can be computed in time and space polynomial in d , n , and the size of the output.

OPEN PROBLEM 22.3.4

Is there an algorithm that, given S , computes the double description of $P = \text{conv}S$ in time polynomial in d , n , and the size of the double description?

The algorithm in Chazelle's theorem appears to be of theoretical interest only. The algorithm of Avis-Fukuda is quite practical, the algorithms of Swart and of Chand and Kapur are less so because of the potentially large space requirements. (See [Chapter 52](#) for descriptions of available code.) The running times of the last two algorithms admit some theoretical improvements, as will be discussed in the following sections.

Almost all algorithms that have been published for solving the different versions of the convex hull problem and the halfspace intersection problem appear to be variations of three general methods: incremental, graph traversal, and divide-and-conquer. We discuss the incremental and the graph traversal methods in the next two subsections. Divide-and-conquer has proven useful only for very small

dimension, and we will discuss it in that context in Section 22.4. Methods that fall outside this threefold classification are discussed in Subsection 22.3.3.

22.3.1 THE INCREMENTAL METHOD

The incremental method puts the points in S in some order p_1, \dots, p_n and then successively computes a description of $P_i = \text{conv}S_i$ from the description of P_{i-1} and p_i , where $S_i = \{p_1, \dots, p_i\}$.

Before discussing details it should be noted that no matter how the incremental method is implemented, it has a serious shortcoming in that the intermediate polytopes P_i may have many more facets than the final $P_n = P$ (see, e.g., [ABS97]). Thus the description sizes of the intermediate polytopes may be much larger than the size of the description of the final result, and hence this method cannot have running time that depends reasonably on the output size.

This is not necessarily just the result of an unfortunate choice of the insertion order, since Bremner [Bre99] has shown that if S is the vertex set of the aforementioned product of cyclic polytopes $CC_d(n)$, then P_{n-1} has $\Omega(m^{\lfloor \sqrt{d/2} \rfloor - 1})$ facets no matter which insertion order is used, where m is the number of facets of $P_n = P$.

We first present a selection of algorithms implementing the incremental method and list their asymptotic worst-case or expected running times for fixed d (Table 22.3.2). All these algorithms compute boundary descriptions, except for [Sei81] (see also [Ede87, Section 8.4]), which can also be made to compute a lattice description, and [MRTT53], which computes a double description.

TABLE 22.3.2 Sample of incremental algorithms.

ALGORITHM	TIME	BOUND TYPE
Kallay [PS85, Section 3.4.2]	$n^{\lfloor d/2 \rfloor + 1}$	worst-case
Seidel [Sei81]	$n \log n + n^{\lceil d/2 \rceil}$	worst-case
Chazelle [Cha93]	$n \log n + n^{\lfloor d/2 \rfloor}$	worst-case
Clarkson-Shor [CS89]	$n \log n + n^{\lfloor d/2 \rfloor}$	expected
Clarkson et al. [CMS93]	$n \log n + n^{\lfloor d/2 \rfloor}$	expected
Motzkin et al. [MRTT53]	$n^{3\lfloor d/2 \rfloor + 1}$	worst-case

We now concentrate on how P_{i-1} and P_i differ. For the sake of simplicity we will first assume that S is nondegenerate and hence all involved polytopes are simplicial. Moreover we will ignore how the insertion method starts and assume that P_{i-1} and P_i are full-dimensional. We say that a facet of P_{i-1} is *visible* (from p_i) if its supporting hyperplane separates P_{i-1} and p_i . Otherwise the facet is *obscured*.

The facet set of P_i consists of “old facets,” namely all obscured facets of P_{i-1} , and “new facets,” namely facets of the form $\text{conv}(R \cup \{p_i\})$, where R is a “horizon” ridge of P_{i-1} , i.e., R is contained in a visible and in an obscured facet of P_{i-1} .

Updating P_{i-1} to P_i thus requires solving three subproblems: finding (and deleting) all visible facets of P_{i-1} ; finding all horizon ridges; forming all new facets. The various incremental algorithms only differ in how they solve those subproblems, and they differ in the type of insertion order used.

Visible facets. The simplest way of finding the visible facets is simply to check each facet of P_{i-1} . This is done in Kallay’s “beneath-beyond” method [PS85, Section 3.4.2] and in the “double description method” of Motzkin et al. [MRTT53]. Since P_i may have $\Theta(i^{\lfloor d/2 \rfloor})$ facets such an approach automatically leads to a sub-optimal overall running time of $\Omega(n^{\lfloor d/2 \rfloor + 1})$ in the worst case.

Another way is to maintain “conflict lists” between facets and not yet inserted points. In the worst case this is no better than the previous method. However, if the insertion order is a random permutation of the points in S , then in expectation this method works in $O(n^{\lfloor d/2 \rfloor})$ time [CS89].

The last method requires the maintenance of a *facet graph*, whose nodes are the facets and whose arcs connect facets if they share a common ridge. The visible facets form a connected subgraph of this facet graph. Thus they can be determined by graph search, such as depth-first search. This takes time proportional to the number of visible facets, which means that in the amortized sense this takes no time since all those visible facets will be deleted. This graph search requires that one starting visible facet be known. Such an initial visible facet can be determined relatively efficiently by a special choice of the insertion order, as in [Sei81], by maintaining “canonical visible facets,” as in [CS89] and [CMS93], or by linear programming, as in [Sei91].

Horizon ridges. Determining the horizon ridges is trivial if the facet graph is used, since those ridges correspond to arcs connecting visible and obscured facets. Otherwise one has to use data structuring techniques to determine which of the ridges incident to the visible facets are incident to exactly one visible facet.

New facets. After the horizon ridges are determined, the new facets are easily constructed in time proportional to their number. Keeping this number small is one of the main difficulties of making the insertion method efficient. In the worst case there may be as many as $\mu(d-1, i-1) = \Theta(i^{\lfloor (d-1)/2 \rfloor})$ such new facets. For even d this is $\Theta(i^{\lfloor d/2 \rfloor - 1})$, which is the main reason why it was relatively easy to obtain an asymptotically worst-case optimal running time of $O(n^{\lfloor d/2 \rfloor})$ for even d [Sei81]. For general d , using a random insertion order [CS89, CMS93, Sei91] appears to be the only known way to keep this number low, at least in terms of expectation. Chazelle’s celebrated deterministic algorithm [Cha93] applies derandomization and thus in effect “simulates” random insertion order so that the number of new facets is not only small in the expected sense but also in the worst case.

Finally, if a facet graph is used, then the arcs corresponding to the ridges between the new facets need to be generated, which can be done via data structuring techniques, as in [Sei91], or by graph traversal techniques, as in [CS89, CMS93]. We should mention that if we remove the nondegeneracy assumption this problem of determining the new ridges seems to become very difficult.

Degenerate input. So far we have assumed that the input set S be nondegenerate. If this is not the case, then this can be simulated using perturbation techniques [Sei96]. This way the algorithms produce a boundary description from which a lattice description or a double description could be computed in $O(n^{\lfloor d/2 \rfloor})$ worst-case time.

The algorithm of Seidel [Sei81] (see also [Ede87, Section 8.4]) also works with degenerate input and then produces a lattice description. Most interesting, though, in the case of degeneracy is the so-called double description algorithm of Motzkin et al. [MRTT53].

THE DOUBLE DESCRIPTION METHOD

Although it is one of the oldest published incremental algorithms, this method has received little attention in the computational geometry community. This method maintains only the double descriptions of the polytopes P_i . It makes no assumptions about nondegeneracy. In fact, despite its poor worst-case complexity, empirically this method works well for degenerate inputs, where all other methods seem to fail, running out of time or space.

The algorithm determines the visible facets by simply checking all facets of P_{i-1} . The interesting point is how it determines the horizon ridges, from which the new facets are then constructed. In contrast to the other methods it does not maintain ridges, since, as we already mentioned, determining the new ridges created during an insertion is difficult. The double description method simply considers each pair of visible and obscured facets of P_{i-1} and checks whether their intersection A forms a horizon ridge. This is achieved by testing whether the vertex set in A is contained in some other facet of P_{i-1} . If it is, then A is not a ridge and hence not a horizon ridge.

A straightforward implementation of this idea will require $\Theta(i^{3\lfloor d/2 \rfloor})$ time in the worst case to discover all horizon ridges of P_{i-1} , resulting in a high worst-case overall running time. Although a number of heuristics have been proposed to speed up this process (see [Zie94, p. 48]), experiments show that this method is unbearably slow in the nondegenerate case when compared to other algorithms. However, in the case of degenerate input it still appears to be the method of choice with the new primal-dual approach (Section 22.3.3) as a possible contender.

Finally, we should mention that convex hull algorithms based on so-called Fourier-Motzkin elimination are nothing but incremental algorithms dressed up in an algebraic formulation.

22.3.2 THE GRAPH TRAVERSAL METHOD

This method attempts to traverse the facet graph of polytope $P = \text{conv}S$ in an organized fashion. The basic step is: given a facet F of P and a ridge R contained in F , find the other facet F' of P that also contains R . Geometrically this amounts to determining the point $p \in S$ such that the hyperplane spanned by R and p maximizes the angle to F . In analogy to a 3D physical realization this operation is therefore known as a “gift-wrapping step,” and these algorithms are known as **gift-wrapping algorithms**. In the polar context of intersecting halfspaces, this step corresponds to moving along an edge from one vertex to another and is equivalent to a pivoting step of the simplex algorithm for linear programming. Thus these algorithms are also known as **pivoting algorithms**.

The basic outline of the graph traversal method is as follows: Find some initial facet of $P = \text{conv}S$ and the ridges that it contains. As long as there is an **open ridge** R , i.e., one for which only one containing facet F is known, perform a gift-wrapping step to discover the other facet F' containing R and determine the ridges that F' contains.

This general method faces three problems:

- (a) How does one maintain the set of open ridges?
- (b) How can the ridges of the new facet F' be quickly discovered?

- (c) How can an individual gift-wrapping step be performed quickly?

THE NONDEGENERATE CASE

Let us again first assume that the input set S is in nondegenerate position. This trivializes problem (b) since every facet is a $(d-1)$ -simplex and each of the d subsets with $d-1$ of its d vertices will span a ridge.

The most straightforward way to deal with problem (a) is to use some sort of dictionary data structure to store the set of open ridges. The most straightforward way to deal with (c) is to scan through all the points in S to find the best candidate, leading to work proportional to n per discovered facet. This straightforward method has been proposed many times (see [Sch86, p. 224] and [Chv83, p. 282] for references) and has running time $O(d^2nM)$ using $O(d(M+n))$ space, where M is the number of facets of P .

The gift-wrapping steps can be performed faster if a special data structure (for the dual of ray-shooting queries) is used. This was developed by Chan [Cha96], who achieved for fixed $d > 3$ an asymptotic time bound of

$$O(n \log M + (nM)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n).$$

Avis and Fukuda [AF92] proposed an ingenious way to deal with problem (a) so that no storage space is needed. They pointed out that there is a way of defining a canonical spanning tree T of the facet graph of polytope P so that the arcs of T can be recognized locally. Gift-wrapping steps are then performed only over ridges corresponding to arcs of T . Doing this in the form of a depth-first search traversal of T avoids the use of any extra storage space. Facets can be output as soon as they are discovered. Their algorithm is eminently practical and has a running time of $O(dnM)$ using only $O(dn)$ space.

In theory the gift-wrapping step improvement of Chan also could be applied to the algorithm of Avis and Fukuda. However, this appears to be of little practical relevance.

A completely different way of simultaneously addressing problems (a) and (c) was suggested by Seidel [Sei86a]. He proposed to try to discover the facets in an order corresponding to a straight-line shelling of P . In many cases gift-wrapping steps over several currently open ridges would yield the same new facet F' . However, in that case the entire vertex set of F' is known already and the expensive scan to solve problem (c) is not necessary. The facets of P for which this trick is not applicable can be discovered in advance by linear programming. This “shelling algorithm” has running time $O(n\lambda(n-1, d-1) + d^3 M \log n)$, where $\lambda(n-1, d-1)$ is again the time necessary to solve a linear program with $n-1$ constraints in $d-1$ variables. From the way a shelling proceeds one can prove that the space requirement for storing the open ridges is somewhat lower than in an ordinary gift-wrapping algorithm.

The linear programs that need to be solved are similar to the ones in the irredundancy problem of Section 22.2. Again improvements can be achieved by applying linear programming queries ([Mat93]), and the $n\lambda(n-1, d-1)$ factor can be improved to $n^{2-2/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n$.

THE GENERAL CASE

There are two ways to approach the general case where P is not simplicial. The first is again to apply perturbations in order to simulate nondegeneracy of S . This

way all previously mentioned algorithms still apply, however they now compute a boundary description of P . The parameter M is now the size of the triangulation that happens to be constructed. Moreover, the perturbed computations slow down the running times by a polynomial factor in d .

The second way to deal with the general case is to generalize the algorithms so that they compute the lattice description of P . The main obstacle that must be overcome in the degenerate case is problem (b), the discovery of the ridges of a new facet F' . The obvious way to address this problem is to view the construction of F' as a recursive subproblem one dimension down. Some care must be taken however that in the many recursions small-dimensional faces are not reconstructed too often. This method was proposed by Chand and Kapur [CK70] and their algorithm was later improved and analyzed by Swart [Swa85] who showed a running time of $O(d^2nK_1 + d^3K_2 \log K_0)$, where K_i is the number of directed $(i+1)$ -vertex paths in the Hasse diagram of the face lattice of P .

Rote [Rot92] generalized the algorithm of Avis and Fukuda to produce the lattice description using little storage space. Its running time is $O(dK_{d+1}n)$ and it appears to be not as relevant in practice as the original algorithm.

Finally, Seidel [Sei86b] generalized his shelling algorithm to produce the lattice description in time $O(n\lambda(n-1, d-1) + K_2(d^2 + \log K_0))$. Because of the recursive nature of straight-line shellings, this generalization avoids reconstruction of small-dimensional faces. Again the improvement via linear programming queries applies.

22.3.3 OTHER METHODS

THE BRUTE-FORCE APPROACH

Let S be a set of n points in \mathbb{R}^d and let $P = \text{conv}S$. Assume w.l.o.g. that the origin is contained in the interior of P (otherwise apply a translation) and assume that S is irredundant in the sense that every point in S is a vertex of P (otherwise apply the results of Section 22.2).

A set $T \subset S$ spans a face of P iff there is a halfspace that has T on its boundary and $S \setminus T$ in its interior. Algebraically this can be tested by determining

$$y_T = \max\{y \in \mathbb{R} \mid \exists x \in \mathbb{R}^d : \forall p \in T : \langle x, p \rangle = 1 \text{ and } \forall p \in S \setminus T : \langle x, p \rangle + y \leq 1\},$$

which can be computed via linear programming, and checking that $y_T > 0$.

This characterization immediately yields a straightforward algorithm with running time $O(2^n \lambda(n, d))$ for generating all faces and also the lattice description of P : Simply test each subset of S whether it spans a face of P . This brute-force approach can be substantially improved by applying backtrack-search techniques ([Bal61],[FLM97]). Fukuda et al. [FLM97] even achieve a running time of $O(nK_0\lambda(n, d))$ this way, using just $O(dn)$ space. Unfortunately this backtrack-search approach does not seem to yield an efficient method to compute the double description of P .

THE PRIMAL-DUAL METHOD

Let S be a set of n points in \mathbb{R}^d , let $P = \text{conv}S$, and let \mathcal{F} be the set of facets of P . Determining \mathcal{F} from S is difficult if P is degenerate in the sense that it is not simplicial, i.e., its facets are not all simplices. However, in this case determining

S from \mathcal{F} may not be so difficult. The primal-dual method [BFM98] of Bremner, Fukuda, and Marzetta tries to exploit this possibility, despite the fact that \mathcal{F} is unknown and S is the input.

The basic idea of their algorithm is as follows: For a facet $F \in \mathcal{F}$, let H_F be the halfspace that has F on its boundary and contains P , and for $\mathcal{G} \subset \mathcal{F}$ let $H_{\mathcal{G}} = \{H_G | G \in \mathcal{G}\}$. Assume some $\mathcal{G} \subset \mathcal{F}$ is known already. Enumerate the vertices of the polyhedron $P_{\mathcal{G}} = \bigcap H_{\mathcal{G}} \supset P$. If all the vertices found are points in S and if $P_{\mathcal{G}}$ is bounded, then it must be the case that $P_{\mathcal{G}} = P$ and $\mathcal{G} = \mathcal{F}$ and all facets of P have been found, and we are done. If this is not the case (and this can be determined after at most $n+1$ vertices of $P_{\mathcal{G}}$ have been enumerated), then it is easy to find a point $v \in P_{\mathcal{G}} \setminus P$ (either a vertex not in S or a point on an extreme ray of $P_{\mathcal{G}}$). But now clearly $\mathcal{G} \neq \mathcal{F}$. Moreover it is easy to find a facet $G \in \mathcal{F} \setminus \mathcal{G}$ (or rather the halfspace H_G) that separates v from P . This amounts to performing the initial facet finding step of the gift-wrapping algorithm and can be done (without linear programming!) in $O(d^2n)$ time. Now add G to \mathcal{G} and repeat.

The method suggests that the complexity of computing the facet description of a polytope P from its vertex description is related to the complexity of computing the vertex description from the facet description. It is difficult to make this theoretical statement precise without introducing assumptions about the intermediate polyhedra $P_{\mathcal{G}}$. However, on the practical side, the authors of [BFM98] present experimental evidence showing that the primal-dual method outperforms other algorithms in certain “degenerate” cases.

22.4 THE CASE OF SMALL DIMENSION

Convex hull computations in very small dimension are special. We have strong geometric intuitions about 2D and 3D space (and via Schlegel diagrams even about 4-polytopes). Moreover the situation is simpler in the case $d = 2, 3$ since our five polytope descriptions cannot differ much in terms of their sizes (they are all within a constant factor of each other), which means there is little need for keeping an exact distinction. Algorithmically, small dimensions are special in that besides the incremental and the graph traversal method, divide-and-conquer methods have also been brought to fruition.

THE 2-DIMENSIONAL CASE

The planar convex hull problem has drawn considerable attention and many different algorithmic paradigms have been tried (see textbooks such as [PS85] or [O'R98]). The graph traversal method was rediscovered and is known in the planar case as the **Jarvis march** with running time $O(nM)$, and the incremental method was rediscovered and is known in a rather different guise as the **Graham scan** with running time $O(n \log n)$ (as usual n and M are the sizes of the input and output, respectively). It was easy and natural to apply the divide-and-conquer paradigm to obtain further $O(n \log n)$ time algorithms. By giving this paradigm the extra twist of “marriage-before-conquest” it was possible even to obtain an $O(n \log M)$ algorithm, which was also shown to be worst-case optimal in the algebraic computation tree model of computation [KS86]. This algorithm required the use of

2D linear programming. Much later Chan, Snoeyink, and Yap [CSY97] showed how to avoid this and substantially simplified the algorithm in way that allowed its generalization to higher dimensions. Later Chan [Cha96] showed quite surprisingly that by using simple data structures and the method of guessing the output size by repeated squaring, the Jarvis march algorithm can be sped up to also run in time $O(n \log M)$.

THE 3-DIMENSIONAL CASE

In 3 dimensions the output size M is $O(n)$ in the worst case. However, the straightforward implementations of the standard incremental and the graph traversal methods only yield algorithms with worst-case running time $O(n^2)$. In this context the use of the divide-and-conquer paradigm was decisive in obtaining $O(n \log n)$ running time, which was achieved by Preparata and Hong (see [PS85, Section 3.4.4]; for a more detailed account, [Ede87, Section 8.5]). This running time was later matched in the expected sense by the randomized incremental algorithm of Clarkson and Shor [CS89], who also gave another randomized algorithm with expected performance $O(n \log M)$.

The question whether this optimal output-size sensitive bound could also be achieved deterministically was open for a long time. Edelsbrunner and Shi [ES91] first generalized the “marriage-before-conquest” method of [KS86] but achieved only a running time of $O(n \log^2 M)$. Eventually Chazelle and Matoušek [CM92] succeeded in derandomizing the randomized algorithm of Clarkson and Shor and obtained, at least theoretically, this optimal $O(n \log M)$ time bound. Later Chan [Cha96] showed that there is a relatively simple algorithm for achieving this bound, again by the method of speeding up the gift-wrapping method using data structures and guessing the output size by repeated squaring.

THE CASE $d = 4, 5$

In this case the sizes of the combinatorial descriptions may be as large as $\Theta(n^2)$. All the methods and bounds mentioned in Section 22.3 apply. In addition there are methods for computing a boundary description based on sophisticated divide-and-conquer and some additional pruning mechanisms. Worst-case time bounds of $O((n + M) \log^{d-2} M)$ were achieved by Chan, Snoeyink, and Yap [CSY97] for $d = 4$, and by Amato and Ramos [AR96] for $d = 4, 5$. The latter paper also states that their bound applies to computing the lattice description in the case $d = 4$.

22.5 RELATED TOPICS

There has been some work on determining the intrinsic computational complexity of versions of the convex hull problem. The strongest results at this point are:

1. For fixed $d \geq 2$ the time necessary to determine whether exactly V of n points in \mathbb{R}^d are extreme is $\Omega(n \log V)$ in the algebraic computation tree model [KS86]. This is asymptotically best possible for $d = 2$.

- For fixed $d \geq 2$ the time necessary to determine whether the convex hull of n points in \mathbb{R}^d has exactly M facets is $\Omega(n^{\lceil d/2 \rceil - 1} + n \log n)$ in a specialized but realistic model of computation [E99]. This is asymptotically best possible for odd $d > 1$.

The expected sizes of convex hulls of point sets drawn according to some statistical distribution are typically much smaller than the worst-case sizes. Constructing such convex hulls has been explicitly studied by several authors (see, e.g., [DT81, Dwy91, BGJR91]). One should also mention in this context the randomized incremental algorithm [CS89]. With input set $S \subset \mathbb{R}^d$ its expected running time for constructing a boundary description is

$$O\left(\sum_{d+1 < r \leq n} df_r(S)/r + \sum_{d+1 \leq r < n} d^2 nf_r(S)/r^2\right),$$

where $f_r(S)$ is the expected size of the boundary description of the convex hull of a random subset of S of size r . For many distributions f_r is sufficiently sublinear so that this randomized incremental algorithm has $O(n)$ expected running time.

The problem of maintaining convex hulls under insertions and deletions of points has been addressed also. In higher dimensions randomized incremental algorithms have been adapted by several authors to process updates [Mul94, Sch91, CMS93]. However, the analyses are all based on some probabilistic model of which updates actually occur. More satisfactory solutions have only been obtained in the planar case. Solutions with $O(\log n)$ update time were obtained for the insertions only case (see [PS85, Section 3.3.6]) and also for the deletions only case [HS92]. For the general dynamic case $O(\log^2 n)$ update times were achieved early on [OvL81, Gow80], and only very recently they were improved to $O(\log n)$ in [Cha01, BJ02].

For some time there was hope that additional input information might help compute convex hulls. Although this is true in the planar case, where having points presorted or having them given along a nonintersecting polygonal line [Mel87] leads to linear-time algorithms, it has been shown [Sei85] that for dimension $d \geq 3$ such additional information does not help. Having a 3D set S presorted or even knowing a nonself-intersecting polyhedral surface whose vertex set is S does not in general make it easier to find the convex hull of S .

There have been some attempts to generalize the convex hull construction problem so that the input S does not consist of points but of more general objects such as algebraically described regions in the plane [BK91, NY98], balls in \mathbb{R}^d [BCD⁺92], ellipsoids in \mathbb{R}^3 [Wol02], or sets of polyhedra [FLL01].

Finally, parallel algorithms for the convex hull problem have been developed; see [Chapter 42](#).

22.6 SOURCES AND RELATED MATERIALS

FURTHER READING

[Zie94]: A modern account of polytope theory.

[MR80]: A survey of vertex enumeration methods from the dual standpoint.

RELATED CHAPTERS

[Chapter 16: Basic properties of convex polytopes](#)

[Chapter 18: Face numbers of polytopes and complexes](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 45: Linear programming](#)

REFERENCES

- [ABS97] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Comput. Geom. Theory Appl.*, 7:265–301, 1997.
- [AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.
- [AR96] N.M. Amato and E.A. Ramos. On computing Voronoi diagrams by divide-prune-and-conquer. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 166–175, 1996.
- [BK91] C.L. Bajaj and M.-S. Kim. Convex hulls of objects bounded by algebraic curves. *Algorithmica*, 6:533–553, 1991.
- [Bal61] M.L. Balinski. An algorithm for finding all vertices of convex polyhedral sets. *SIAM J. Appl. Math.*, 9:72–81, 1961.
- [BCD⁺92] J.-D. Boissonnat, A. Cérézo, O. Devillers, J. Duquesne, and M. Yvinec. An algorithm for constructing the convex hull of a set of spheres in dimension d . In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 269–273, 1992.
- [BGJR91] K.H. Borgwardt, N. Gaffke, M. Jünger, and G. Reinelt. Computing the convex hull in the Euclidean plane in linear expected time. In P. Gritzmann and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Math. and Theoret. Comput. Sci.*, pages 91–107. Amer. Math. Soc., Providence, 1991.
- [Bre99] D. Bremner. Incremental convex hull algorithms are not output sensitive. *Discrete Comput. Geom.*, 21:57–68, 1999.
- [BFM98] D. Bremner, K. Fukuda, and A. Marzetta. Primal-dual methods for vertex and facet enumeration. *Discrete Comput. Geom.*, 20:333–357, 1998.

- [BJ02] G.S. Brodal and R. Jacob. Dynamic planar convex hull. *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, pages 617–626, 2002.
- [Cha93] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.
- [Cha96] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete Comput. Geom.*, 16:369–387, 1996.
- [Cha96a] T.M. Chan. Fixed-dimensional linear programming queries made easy. In *Proc. 12th Annu. ACM Symp. Comput. Geom.*, pages 284–290, 1996.
- [Cha01] T.M. Chan. Dynamic planar convex hull operations in near-logarithmic time. *J. Assoc. Comput. Mach.*, 48:1–12, 2001.
- [Chv83] V. Chvátal. *Linear Programming*. W.H. Freeman, New York, 1983.
- [CK70] D.R. Chand and S.S. Kapur. An algorithm for convex polytopes. *J. Assoc. Comput. Mach.*, 17:78–86, 1970.
- [Cla94] K.L. Clarkson. More output-sensitive geometric algorithms. In *Proc. 35th Annu. IEEE Symp. Found. Comput. Sci.*, pages 695–702, 1994.
- [CM92] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. Tech. Rep., Dept. Comput. Sci., Princeton Univ. Press, 1992.
- [CMS93] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3:185–212, 1993.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [CSY97] T.M. Chan, J. Snoeyink, and C.K. Yap. Primal dividing and dual pruning: Output-sensitive construction of four-dimensional polytopes and three-dimensional Voronoi diagrams. *Discrete Comput. Geom.*, 18:433–454, 1997.
- [DT81] L. Devroye and G.T. Toussaint. A note on linear expected time algorithms for finding convex hulls. *Computing*, 26:361–366, 1981.
- [Dwy91] R. Dwyer. Convex hulls of samples from spherically symmetric distributions. *Discrete Appl. Math.*, 31:113–132, 1991.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monogr. Theoret. Comput. Sci.* Springer-Verlag, Heidelberg, 1987.
- [ES91] H. Edelsbrunner and W. Shi. An $O(n \log^2 h)$ time algorithm for the three-dimensional convex hull problem. *SIAM J. Comput.*, 20:259–277, 1991.
- [E99] J. Erickson. New lower bounds for convex hull problems in odd dimensions. *SIAM J. Comput.*, 28:1198–1214, 1999.
- [FLL01] K. Fukuda, T.M. Liebling, and C. Lütolf. Extended convex hull. *Comput. Geom. Theory Appl.*, 20:13–23, 2001.
- [FLM97] K. Fukuda, T.M. Liebling, and F. Margot. Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Comput. Geom. Theory Appl.*, 8:1–12, 1997.
- [Gow80] I.G. Gowda. *Dynamic problems in computational geometry*. M.Sc. thesis, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1980.
- [HS92] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32:249–267, 1992.
- [KS86] D.G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, 15:287–299, 1986.

- [Mat93] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.
- [Mel87] A. Melkman. On-line construction of the convex hull of a simple polyline. *Inform. Process. Lett.*, 25:11–12, 1987.
- [MR80] T.H. Matteiss and D. Rubin. A survey and comparison of methods for finding all vertices of convex polyhedral sets. *Math. Oper. Res.*, 5:167–185, 1980.
- [MRTT53] T.S. Motzkin, H. Raiffa, G.L. Thompson, and R.M. Thrall. The double description method. In H.W. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games II*, volume 8 of *Ann. of Math. Stud.*, pages 51–73. Princeton University Press, 1953.
- [Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, 1994.
- [NY98] F. Nielsen and M. Yvinec. Output-sensitive convex hull algorithms of planar convex objects. *Comp. Geom. Theory Appl.* 8:39–66, 1998.
- [O'R98] J. O'Rourke. *Computational Geometry in C*. Second Edition. Cambridge University Press, 1998.
- [OSS95] T.A. Ottmann, S. Schuierer, and S. Soundaralakshmi. Enumerating extreme points in higher dimensions. In *Proc. 12th Sympos. Theoretical Aspects of Comput. Sci.*, Springer Lect. Notes in Comput. Sci., volume 900, 562–570, 1995.
- [OvL81] M.H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23:166–204, 1981.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [Ram00] E.A. Ramos. Linear optimization queries revisited. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 176–181, 2000.
- [Rot92] G. Rote. Degenerate convex hulls in high dimensions without extra storage. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 26–32, 1992.
- [Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience, New York, 1986.
- [Sch91] O. Schwarzkopf. Dynamic maintenance of geometric structures made easy. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 197–206, 1991.
- [Sei81] R. Seidel. *A convex hull algorithm optimal for point sets in even dimensions*. M.Sc. thesis, Dept. Comput. Sci., Univ. British Columbia, Vancouver, 1981. Report 81/14.
- [Sei85] R. Seidel. A method for proving lower bounds for certain geometric problems. In G.T. Toussaint, editor, *Computational Geometry*, pages 319–334. North-Holland, Amsterdam, 1985.
- [Sei86a] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 404–413, 1986.
- [Sei86b] R. Seidel. *Output-size sensitive algorithms for constructive problems in computational geometry*. Ph.D. thesis, Dept. Comput. Sci., Cornell Univ., Ithaca, 1986. Tech. Rep. TR 86-784.
- [Sei91] R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.
- [Sei96] R. Seidel. The meaning and nature of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1996.
- [Swa85] G.F. Swart. Finding the convex hull facet by facet. *J. Algorithms*, 6:17–48, 1985.

- [Wol02] N. Wolpert. *An exact and efficient approach for computing a cell in an arrangement of quadrics*. Ph.D. thesis, FR Informatik, Univ. des Saarlandes, Saarbrücken, 2002.
- [Zie94] G.M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Math.* Springer-Verlag, New York, 1994.

23 VORONOI DIAGRAMS AND DELAUNAY TRIANGULATIONS

Steven Fortune

INTRODUCTION

The Voronoi diagram of a set of sites partitions space into regions, one per site; the region for a site s consists of all points closer to s than to any other site. The dual of the Voronoi diagram, the Delaunay triangulation, is the unique triangulation such that the circumsphere of every simplex contains no sites in its interior. Voronoi diagrams and Delaunay triangulations have been rediscovered or applied in many areas of mathematics and the natural sciences; they are central topics in computational geometry, with hundreds of papers discussing algorithms and extensions.

Section 23.1 discusses the definition and basic properties in the usual case of point sites in \mathbb{R}^d with the Euclidean metric, while Section 23.2 gives basic algorithms. Some of the many extensions obtained by varying metric, sites, environment, and constraints are discussed in Section 23.3. Section 23.4 finishes with some interesting and nonobvious structural properties of Voronoi diagrams and Delaunay triangulations.

GLOSSARY

Site: A defining object for a Voronoi diagram or Delaunay triangulation. Also generator, source, Voronoi point.

Voronoi face: The set of points for which a single site is closest (or more generally a set of sites is closest). Also Voronoi region, Voronoi cell.

Voronoi diagram: The set of all Voronoi faces. Also Thiessen diagram, Wigner-Seitz diagram, Blum transform, Dirichlet tessellation.

Delaunay triangulation: The unique triangulation of a set of sites such that the circumsphere of each full-dimensional simplex has no sites in its interior.

23.1 POINT SITES IN THE EUCLIDEAN METRIC

See [Aur91, Ede87, For95] for more details and proofs of material in this section.

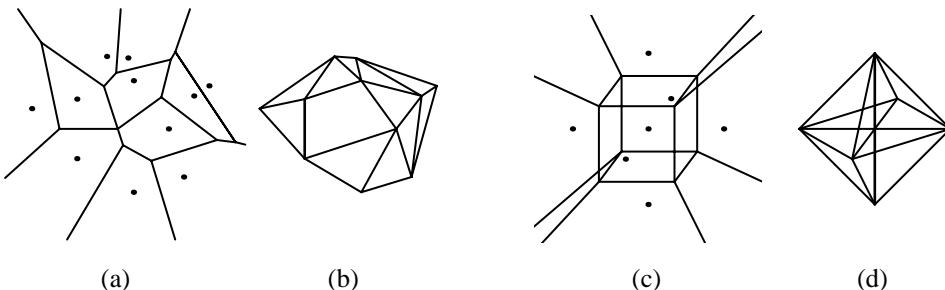
GLOSSARY

Sites: Points in a finite set S in \mathbb{R}^d .

Voronoi face of a site s : The set of all points of \mathbb{R}^d strictly closer to the

FIGURE 23.1.1

Voronoi diagram and Delaunay triangulation of the same set of sites in two dimensions (a,b) and three dimensions (c,d).



site $s \in S$ than to any other site in S . The Voronoi face of a site is always a nonempty, open, convex, full-dimensional subset of \mathbb{R}^d .

Voronoi face $V(T)$ of a subset T : For T a nonempty subset of S , the set of points of \mathbb{R}^d equidistant from all members of T and closer to any member of T than to any member of $S \setminus T$.

Voronoi diagram of S : The collection of all nonempty Voronoi faces $V(T)$, for $T \subseteq S$. The Voronoi diagram forms a cell complex partitioning \mathbb{R}^d . In two dimensions (Figure 23.1.1(a)), the Voronoi face of a site is the interior of a convex, possibly infinite polygon; its boundary consists of **Voronoi edges** (1-dimensional faces) equidistant from two sites and **Voronoi vertices** (0-dimensional faces) equidistant from at least three sites. Figure 23.1.1(c) shows a Voronoi diagram in three dimensions.

Delaunay face $D(T)$ of a subset T : The Delaunay face $D(T)$ is defined for a subset T of S whenever there is a sphere through all the sites of T with all other sites exterior (equivalently, whenever $V(T)$ is not empty). Then $D(T)$ is the (relative) interior of the convex hull of T . For example, in two dimensions (Figure 23.1.1(b)), a **Delaunay triangle** is formed by three sites whose circumcircle is empty and a **Delaunay edge** connects two sites that have an empty circumcircle (in fact, infinitely many empty circumcircles).

Delaunay triangulation of S : The collection of all Delaunay faces. The Delaunay triangulation forms a cell complex partitioning the convex hull of S .

There is an obvious one-one correspondence between the Voronoi diagram and the Delaunay triangulation; it maps the Voronoi face $V(T)$ to the Delaunay face $D(T)$. This correspondence has the property that the sum of the dimensions of $V(T)$ and $D(T)$ is always d . Thus, in two dimensions, $V(T)$ is a Voronoi vertex iff $D(T)$ is an open polygonal region; $V(T)$ is an edge iff $D(T)$ is; $V(T)$ is an open polygonal region iff $D(T)$ is a vertex, i.e., a site. In fact, the 1-1 correspondence is a duality between cell complexes, reversing face ordering: for subsets $T, T' \subseteq S$, $V(T')$ is a face of $V(T)$ iff $D(T')$ is a face of $D(T)$.

The set of sites $S \subset \mathbb{R}^d$ is in **general position** (or is **nondegenerate**) if no $d+2$ points lie on a common d -sphere and no $k+2$ points lie on a common k -flat, for $k < d$. If S is in general position, then the Delaunay triangulation of S is a simplicial complex, and every vertex of the Voronoi diagram is incident to $d+1$ edges in the Delaunay triangulation. If S is not in general position, then Delaunay faces need

not be simplices; for example, the four cocircular sites in [Figure 23.1.1\(b\)](#) form a Delaunay quadrilateral. A **completion** of a Delaunay triangulation is obtained by splitting nonsimplicial faces into simplices without adding new vertices.

RELATION TO CONVEXITY

There is an intimate connection between Delaunay triangulations in \mathbb{R}^d and convex hulls in \mathbb{R}^{d+1} , and between Voronoi diagrams in \mathbb{R}^d and halfspace intersections in \mathbb{R}^{d+1} . To see the connections, consider the special case of $d = 2$. Identify \mathbb{R}^2 with the plane spanned by the first two coordinate axes of \mathbb{R}^3 , and call the third coordinate direction the *vertical* direction.

The **lifting map** $\lambda : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is defined by $\lambda(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$; $\Lambda = \lambda(\mathbb{R}^2)$ is a paraboloid of revolution about the vertical axis. See [Figure 23.1.2\(a\)](#). Let H be the convex hull of the lifted sites $\lambda(S)$.

The Delaunay triangulation of S is exactly the orthogonal projection into \mathbb{R}^2 of the lower faces of H (a face is *lower* if it has a supporting plane with inward normal having positive vertical coordinate). To see this informally, suppose that triangle $\lambda(s)\lambda(t)\lambda(u)$ is a lower facet of H , and that plane P passes through $\lambda(s)\lambda(t)\lambda(u)$. The intersection of P with Λ is an ellipse that projects orthogonally to a circle in \mathbb{R}^2 ([Figure 23.1.2\(a\)](#)). Since all other lifted sites are above the plane, all other unlifted sites are outside the circle, and stu is a Delaunay triangle. The opposite direction, that a Delaunay triangle is a lower facet, is similar.

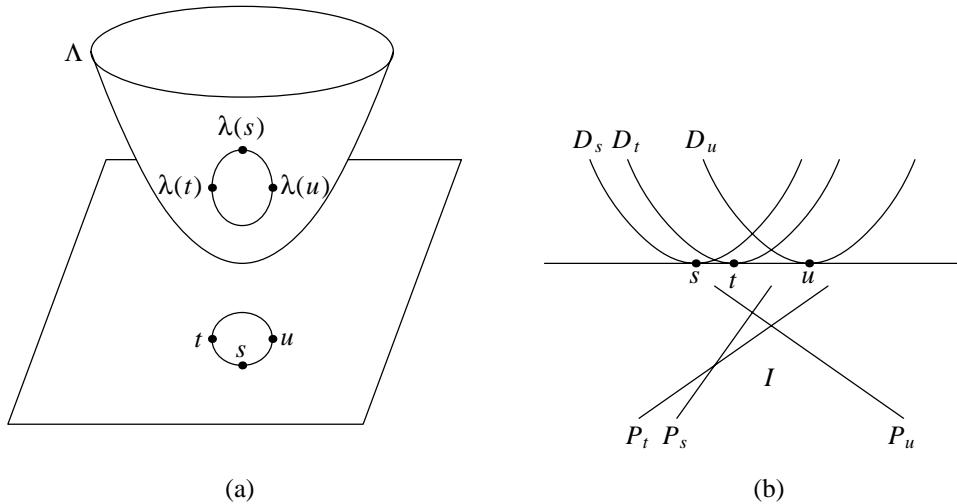
For Voronoi diagrams, assign to each site $s = (s_1, s_2)$ the plane

$$P_s = \{(x_1, x_2, x_3) : x_3 = -2x_1s_1 + s_1^2 - 2x_2s_2 + s_2^2\}.$$

Let I be the intersection of the lower halfspaces of the planes P_s . The Voronoi diagram is exactly the orthogonal projection into \mathbb{R}^2 of the upper faces of I . To

FIGURE 23.1.2

(a) *The intersection of a plane with Λ is an ellipse that projects to a circle;* (b) *on any vertical line, the surfaces $\{D_s\}$ appear in the same order as the planes $\{P_s\}$.*



see this informally, consider the surfaces

$$D_s = \{(x_1, x_2, x_3) : x_3 = ((x_1 - s_1)^2 + (x_2 - s_2)^2)\}$$

(see [Figure 23.1.2](#)). Viewed as a function from \mathbb{R}^2 into R , D_s gives the squared distance to site s . Furthermore, P_s and D_s differ only by the quadratic term $x_1^2 + x_2^2$, which is independent of s . Hence a point $x \in \mathbb{R}^2$ is in the Voronoi cell of site t iff on the vertical line through x , D_t is lowest among all surfaces $\{D_s\}$. This happens exactly if, on the same line, P_t is lowest among all planes $\{P_s\}$, i.e., x is in the projection of the upper face of I formed by P_t .

COMBINATORIAL COMPLEXITY

In dimension 2, a Voronoi diagram of $n \geq 3$ sites has at most $2n - 5$ vertices and $3n - 6$ edges (and the Delaunay triangulation has at most as many triangles and edges, respectively).

In dimension $d \geq 3$ the Voronoi diagram and Delaunay triangulation can have $\Theta(n^{\lceil d/2 \rceil})$ faces. Exact bounds can be given using results from convex polytope theory (section 15). For n sites in d dimensions, the maximum number of Voronoi k -dimensional faces, $k < d$, is $f_{n-k}(C_{d+1}(n)) - \delta_{0k}$, where $C_{d+1}(n)$ is the $d+1$ -dimensional cyclic polytope, f_{n-k} gives the number of $n-k$ dimensional faces (see [Section 18.3](#) and [Theorem 18.3.4](#)), and $\delta_{0k} = 1$ if $k = 0$ and 0 otherwise.

For a simple lower bound example in dimension 3, choose $n/2$ distinct point sites on each of two noncoplanar line segments l and l' . Then there is an empty sphere through each quadruple of sites (a, a', b, b') with a, a' adjacent on l and b, b' adjacent on l' . Since there are $\Omega(n^2)$ such quadruples, there are as many Delaunay tetrahedra (and Voronoi vertices).

If point sites are chosen uniformly at random from inside a sphere, then the expected number of faces is linear in the number of sites. In dimension 2, the expected number of Delaunay triangles is $2n$; in dimension 3, the expected number of Delaunay tetrahedra is $\sim 6.77n$; in dimension 4, the expected number of Delaunay 4-simplices is $\sim 31.78n$ [[Dwy91](#)]. Similar bounds probably hold for other distributions, but proofs are lacking.

Subquadratic bounds on the complexity of the Delaunay triangulation of point sites in \mathbb{R}^3 can be obtained in a few cases. The *spread (of points)* of a set of points is the ratio between largest and smallest interpoint distances. A point set in \mathbb{R}^3 of size n with spread Δ can have at most $O(\Delta^3)$ Delaunay tetrahedra, for all $\Delta = O(\sqrt{n})$ [[Eri02](#)]. Thus if the point set is dense, i.e., has spread $O(n^{1/3})$, there are only $O(n)$ tetrahedra. Points chosen on a surface can also have subquadratic complexity. For example, if the surface is sufficiently continuous and satisfies mild genericity conditions, then any (ϵ, κ) -sample of n points on the surface has complexity $O(n \log n)$ [[ABL03](#)]. A set of points is an (ϵ, κ) -sample if for any point in the set, there is at least one and at most κ other points in the set within geodesic distance ϵ measured on the surface.

23.2 BASIC ALGORITHMS

[Table 23.2.1](#) lists basic algorithms that compute the Delaunay triangulation of n point sites in \mathbb{R}^d using the Euclidean metric. Using the connection with con-

vexity, any $(d+1)$ -dimensional convex hull algorithm can be used to compute a d -dimensional Delaunay triangulation; in fact the divide-and-conquer, incremental, and gift-wrapping algorithms are specialized convex hull algorithms. Running times are given both for worst-case inputs, and for inputs chosen uniformly at random inside a sphere, with expectation taken over input distribution. The Voronoi diagram can be obtained in linear time from the Delaunay triangulation, using the one-one correspondence between their faces. See [Aur91, Ede87, For95] for more citations. Chapter 64 lists available implementations of Voronoi diagram algorithms.

TABLE 23.2.1 Delaunay Triangulation algorithms in the Euclidean metric for point sites.

ALGORITHM	DIM	WORST CASE	UNIFORM
Flipping	2	$O(n^2)$	
Plane sweep	2	$O(n \log n)$	
Divide-and-conquer	2	$O(n \log n)$	$O(n)$
Randomized incremental	2	$O(n \log n)$	
Randomized incremental	≥ 3	$O(n^{\lceil d/2 \rceil})$	$O(n \log n)$
Gift-wrapping	≥ 2	$O(n^{\lceil d/2 \rceil} + 1)$	$O(n)$

THE RANDOMIZED INCREMENTAL ALGORITHM

The incremental algorithm adds sites one by one, updating the Delaunay triangulation after each addition. The update consists of discovering all Delaunay faces whose circumspheres contain the new site. These faces are deleted and the empty region is partitioned into new faces, each of which has the new site as a vertex. See Figure 23.2.1. An efficient algorithm requires a good data structure for finding the faces to be deleted. Then the running time is determined by the total number of face updates, which depends upon site insertion order. The bounds given in Table 23.2.1 are the expected running time of an algorithm that makes a random choice of insertion order, with each insertion permutation equally likely; the bounds for the worst-case insertion order are about a factor of n worse. (For uniform data there is a double expectation, over both insertion order and input distribution.) With additional algorithmic complexity, it is possible to obtain deterministic algorithms with the same worst-case running times [Cha91].

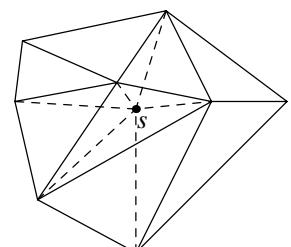


FIGURE 23.2.1

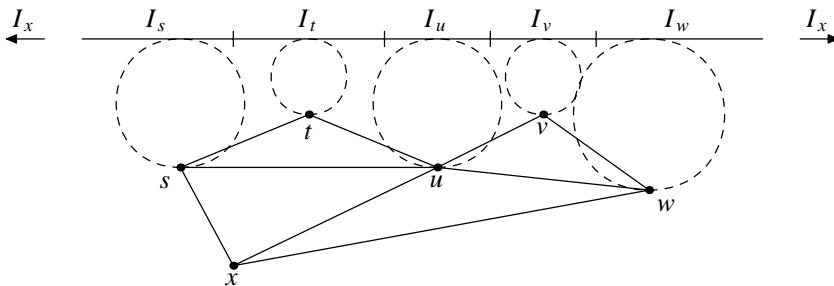
The addition of site s deletes four triangles and adds six (shown dashed).

THE PLANE SWEEP ALGORITHM

The plane sweep algorithm computes a planar Delaunay triangulation using a horizontal line that sweeps upward across the plane. The algorithm discovers a Delaunay triangle when the sweepline passes through the topmost point of its circumcircle; in Figure 23.2.2, the Delaunay triangles shown have already been discovered. A sweepline data structure stores an ordered list of sites; the entry for site s corresponds to an interval I_s on the sweepline where each maximal empty circle with topmost point in I_s touches site s . The sweepline moves in discrete steps only when the ordered list changes. This happens when a new site is encountered or when a new Delaunay triangle is discovered (at the topmost point of the circumcircle of three sites that are consecutive on the sweepline list). A priority queue is needed to determine the next sweepline move. The running time of the algorithm is $O(n \log n)$ since the sweepline moves $O(n)$ times—once per site and once per triangle—and it costs time $O(\log n)$ per move to maintain the priority queue and sweepline data structure.

FIGURE 23.2.2

The sweepline list is x, s, t, u, v, w, x . The next Delaunay triangle is tuv .



OTHER ALGORITHMS

The divide-and-conquer algorithm uses a splitting line to partition the point set into two equal halves, recursively computes the Delaunay triangulation of each half, and then merges the two subtriangulations in linear time. The gift-wrapping algorithm is a specialization of the convex-hull gift-wrapping algorithm ([Chapter 19](#)) to Delaunay triangulations. Output-sensitive algorithms, with running time approximately proportional to the actual number of Delaunay facets, have remained elusive [CSY95]. If the sites form the vertices of a convex polygon, then the Voronoi diagram can be computed in linear time [AGS89].

Graphics hardware, in particular Z-buffers, allow efficient practical computation of fixed-resolution approximate Voronoi diagrams for quite general sites and distance functions [HCK99].

23.3 EXTENSIONS

GLOSSARY

Order- k Voronoi diagram: The order- k Voronoi diagram partitions \mathbb{R}^d on the basis of the first k closest sites (without distinguishing order among them).

Furthest site Voronoi diagram: The furthest site Voronoi diagram partitions \mathbb{R}^d on the basis of the furthest site, or equivalently, the closest $n-1$ of n sites.

Constrained Delaunay triangulation: Constrained Delaunay triangulations are defined relative to a set of **constraint facets** that restrict visibility. The constrained Delaunay triangulation of a set of sites has the property that for every simplex, the interior of the simplex circumsphere contains no site visible from the interior of the simplex.

Conforming Delaunay triangulation: Fix a set of noncrossing **constraint facets** E and a set of point sites S . A conforming Delaunay triangulation is the Delaunay triangulation of a set of sites $S' \supseteq S$ so that every facet in E is the union of Delaunay faces of S' .

Power or Laguerre diagram: A Voronoi diagram for sites s_i with weights w_i where the distance from a point x is measured along a tangent to the sphere of radius $\sqrt{w_i}$ centered on s_i .

HIGHER-ORDER VORONOI DIAGRAMS

The order- k Voronoi diagram can be obtained as an appropriate projection of the k -level of an arrangement of hyperplanes (see [Ede87, For93] and Section 24.2 of this Handbook); it can also be obtained as the orthogonal projection of an intersection polytope [AS92]. In dimension 2, the order- k Voronoi diagram has $O(k(n - k))$ faces. In dimensions $d \geq 3$, the sum of the number of faces of the order- j diagrams, $j \leq k$, is $O(n^{\lceil d/2 \rceil} k^{\lfloor d/2 \rfloor + 1})$ [CS89]; finding good bounds for fixed k remains an open problem. See Table 23.3.1 for algorithm bounds.

TABLE 23.3.1 Algorithms for order- k Voronoi diagrams of point sites in the Euclidean metric.

PROBLEM	DIM	TIME
Furthest site	2	$O(n \log n)$
Furthest site	≥ 3	$O(n^{\lceil d/2 \rceil})$
Order- k	2	$O(k(n - k) \log n + n \log^3 n)$
Order- j , $1 \leq j \leq k$	≥ 3	$O(n^{\lceil d/2 \rceil} k^{\lfloor d/2 \rfloor + 1})$

CONSTRAINED DELAUNAY TRIANGULATIONS

Let S be a set of n point sites in \mathbb{R}^2 and E a set of noncrossing *constraint* edges with endpoints in S . A point $p \in \mathbb{R}^2$ is *visible* from a site s if the open segment ps does not intersect any edge of E . The *constrained Delaunay triangulation* (of S with respect to E) is a triangulation of S extending the edges in E so that the circumcircle of every triangle contains no site that is visible from the interior of the triangle. In \mathbb{R}^2 the constrained Delaunay triangulation always exists; it is as close as possible to the true Delaunay triangulation, subject to the constraint that the edges in E must be used. See also [Section 24.2](#).

The *bounded distance* from a site to a point is Euclidean distance if the point is visible, and infinite otherwise; the *bounded Voronoi diagram* of S using E is defined using bounded distance. The bounded Voronoi diagram is dual to a subgraph of the constrained Delaunay triangulation.

Both the constrained Delaunay triangulation and the bounded Voronoi diagram can be computed in time $O(n \log n)$ using either divide-and-conquer or the sweepline paradigm. If the sites and constraint edges are the vertices and edges of a simple polygon, respectively, then the constrained Delaunay triangulation can be computed in linear time [KL93].

The constrained Delaunay triangulation can be generalized to dimension $d > 2$. Let S be a set of point sites in \mathbb{R}^d and E a set of $d-1$ -dimensional closed simplicial *constraint facets* with vertices in S that are noncrossing (the intersection of two constraint simplices is either empty or a face of both). A *constrained Delaunay triangulation* (of S with respect to E) is a triangulation such that constraint facets are triangulation facets and such that, for every d -simplex, the interior of the circumsphere of the simplex contains no site visible from the interior of the simplex. In dimension $d > 2$, a constrained Delaunay triangulation does not always exist; for example, the Schönhardt polyhedron in dimension 3 cannot be triangulated without extra Steiner points (see [Section 25.5](#)). Shewchuk [She98] gives a sufficient condition for the existence of constrained Delaunay triangulations: the constraint simplices must be *ridge-protected*, that is, each j -face of a constraint simplex, $j \leq d-2$, must have a closed circumsphere not containing any sites. Shewchuk [She00] gives an algorithm that will construct the constrained Delaunay triangulation when it exists, in time $O(ns)$, where n is the number of sites and s is the number of simplices in the output. He also gives a potentially simpler algorithm [She03] that transforms an unconstrained Delaunay triangulation into a constrained Delaunay triangulation by incrementally inserting constraint facets.

CONFORMING DELAUNAY TRIANGULATIONS

Let S be a set of point sites in \mathbb{R}^d and E a set of noncrossing j -dimensional *constraint simplices*, $j < d$. A *conforming Delaunay triangulation* of E is the Delaunay triangulation of a set of sites $S' \supseteq S$ so that every simplex in E is the union of faces of Delaunay simplices of S' . In \mathbb{R}^2 , Edelsbrunner and Tan [ET93] give an algorithm for conforming Delaunay triangulations, where the cardinality of S' is $O(n^3)$, n the cardinality of S . In \mathbb{R}^3 , algorithms that result in a finite set S' are known [CVY02], without explicit bounds on its cardinality.

OTHER DISTANCE MEASURES

Table 23.3.2 lists Voronoi diagram algorithms where “distance” is altered. The distance from a site s_i to a point x can be a function of the Euclidean distance $e(s_i, x)$ and a site-specific real weight w_i .

TABLE 23.3.2 Algorithms for point sites in \mathbb{R}^2 , other distance measures.

PROBLEM	DISTANCE TO x	TIME
Additive weights	$w_i + e(s_i, x)$	$O(n \log n)$
Multiplicative weights	$w_i e(s_i, x)$	$O(n^2)$
Laguerre or power	$\sqrt{e(s_i, x)^2 - w_i}$	$O(n \log n)$
L_p	$\ s_i - x\ _p$	$O(n \log n)$
Skew	$e(s_i, x) + \kappa \Delta_y(s_i, x)$	$O(n \log n)$
Convex distance function		$O(n \log n)$
Abstract	axiomatic	$O(n \log n)$
Simple polygon	geodesic	$O(n \log^2 n)$
Crystal growth	$w_i \cdot SP(s_i, x)$	$O(n^3 + nS \log S)$
Anisotropic	local metric tensor	$O(n^{2+\epsilon})$

The seemingly peculiar ***power distance*** [Aur87] is the distance from x to the sphere of radius $\sqrt{w_i}$ about s_i along a line tangent to the sphere. Many of the basic Voronoi diagram algorithms extend immediately to the power distance, even in higher dimension.

A (***polygonal convex distance function***) [CD85] is defined by a convex polygon C with the origin in its interior. The distance from x to y is the real $r \geq 0$ so that the boundary of $rC + x$ contains y . Polygonal convex distance functions generalize the L_1 and L_∞ metrics (C is a diamond or square, respectively); a polygonal convex distance function is a metric exactly if C is symmetric about the origin.

The (***skew distance***) [Aur99] between two points is the Euclidean distance plus a constant times the difference in y -coordinate. It can be viewed as a measure of the difficulty of motion on a plane that has been rotated in three dimensions about the x -axis.

An ***abstract*** Voronoi diagram [KMM93] is defined by the “bisectors” between pairs of sites, which must satisfy special properties.

The ***geodesic*** distance inside an environment of polygonal obstacles is the length of the shortest path that avoids obstacle interiors. Recent progress using the geodesic metric appears in [HS99].

The ***crystal growth*** Voronoi diagram [SD91] models crystal growth where each crystal has a different growth rate. The distance from a site s_i to a point x in the Voronoi face of s_i is $w_i \cdot SP(s_i, x)$, where w_i is a weight and $SP(s_i, x)$ is the shortest path distance lying entirely within the Voronoi face of s_i . The parameter S in the running time measures the time to approximate bisectors numerically.

An ***anisotropic*** Voronoi diagram [LS03] requires a metric tensor at each site to specify how distance is measured from that site. The anisotropic Voronoi diagram generalizes the multiplicatively weighted diagram; both have the property that the

region of a site may be disconnected or not simply-connected.

OTHER SITES

Many classes of sites besides points have been used to define Voronoi diagram and Voronoi-diagram-like objects. For example, the Voronoi diagram of a set of disjoint circles in the plane is just an additively-weighted point-site Voronoi diagram.

The Voronoi diagram of a set of n line segment sites in R^2 can be computed in time $O(n \log n)$ using the sweepline method or the divide-and-conquer method. The divide-and-conquer algorithm extends to circular-arc segments as well. The well-known medial axis of a polygon or polygonal region can be obtained from the Voronoi diagram of its constituent line segments. The medial axis of a simple polygon can be found in linear time, using the linear-time triangulation algorithm [AGS89].

The *straight skeleton* of a simple polygon [AA⁺95] is structurally similar to the medial axis, though it is not strictly a Voronoi diagram. It is defined as the trace of the vertices of the polygon, as the polygon is shrunk by translating each edge inward at a constant rate. Unlike the medial axis, it has only polygonal edges. Several algorithms achieve time and space bounds of roughly $O(nr)$, r the number of reflex vertices; a subquadratic worst-case bound is known [EE99].

The worst-case combinatorial and algorithmic complexity of Voronoi diagrams of general sites in three dimensions is not well understood. For many sites and metrics in R^3 , roughly cubic upper bounds on the combinatorial complexity of the Voronoi diagram can be obtained using the general theory of lower envelopes of trivariate functions (see [Chapter 24](#) on arrangements). Known lower bounds are roughly quadratic, and upper bounds are conjectured to be quadratic.

A specific long-standing open problem is to give tight bounds on the combinatorial complexity of the Voronoi diagram of a set of n lines in R^3 using the Euclidean metric. Roughly quadratic upper bounds are known if the lines have only a constant number of orientations [KS03]. The boundary of the union of infinite cylinders of fixed radius is also known to have roughly quadratic complexity [AS00]; this boundary can be viewed as the level set of the line Voronoi diagram at fixed distance.

Dwyer [Dwy97] shows that the expected complexity of the Euclidean Voronoi diagram of n k -flats in R^d is $\theta(n^{d/(d-k)})$, as long as $d \geq 3$ and $0 \leq k < d$. The flats are assumed to be drawn independently from the uniform distribution on k -flats intersecting the unit ball. Thus the expected complexity of the Voronoi diagram of a set of n lines in R^3 is $O(n^{3/2})$.

Voronoi diagrams in R^3 can be defined by convex distance functions, as in the plane. If the distance function is determined by a convex polytope with a constant number of facets, then the Voronoi diagram of a set of disjoint polyhedra has combinatorial complexity roughly quadratic in the total number of vertices of all polytopes [KS04].

KINETIC VORONOI DIAGRAMS

Consider a set of n moving point sites in R^d , where the position of each site is a continuous function of a real parameter t , representing time. In general the

Voronoi diagram of the points will vary continuously with t , without any change to its combinatorial structure; however at certain discrete values t_i , $i = 1, \dots$, the combinatorial structure will change. A *kinetic Voronoi diagram algorithm* determines times that the structure changes and at each change updates a data structure representation of the Voronoi diagram. An algorithm is known [AG⁺98] that requires linear space and $O(\log n)$ time per structural change. See [Chapter 50](#).

A long-standing open problem is to give tight bounds on the total number of changes to the Voronoi diagram when each site moves along a line at unit speed. The known upper bound is roughly cubic and the lower bound is roughly quadratic. This problem is clearly related to the problem of bounding the complexity of the Voronoi diagram of lines in R^3 , just mentioned above.

OTHER SURFACES

The Delaunay triangulation of a set of points on the surface of a sphere S^d has the same combinatorial structure as the convex hull of the set of points, viewed as sitting in R^{d+1} . On a closed Riemannian manifold, the Delaunay triangulation of a set of sites exists and has properties similar to the Euclidean case, as long as the set of sites is sufficiently dense [LL00, GM01].

MOTION PLANNING

The motion planning problem is to find a collision-free path for a robot in an environment filled with obstacles. The Voronoi diagram of the obstacles is quite useful, since it gives a lower-dimensional skeleton of maximal clearance from the obstacles. In many cases the shape of the robot can be used to define an appropriate metric for the Voronoi diagram. See [Section 47.2](#) for more on the use of Voronoi diagrams in motion planning.

SURFACE RECONSTRUCTION

The *surface reconstruction problem* is to construct an approximation to a two-dimensional surface embedded in R^3 , given a set of points sampled from the surface. A whole class of surface reconstruction methods are based on the computation of Voronoi diagrams [AB99, DG03] (see [Chapters 29](#) on reconstruction and 54 on surface simplification).

IMPLEMENTATIONS

There are a number of available high-quality implementations of algorithms that compute Delaunay triangulations and Voronoi diagrams of point sites in the Euclidean metric. These can be obtained from the web and from the algorithms libraries CGAL and LEDA (see [Chapters 64](#) and [65](#) on implementations). It is typically challenging to implement algorithms for sites other than points or metrics other than the Euclidean metric, largely because of issues of numerical robustness. See [Bur96, Hel01, SIII00] for approaches for line segment sites in the plane.

23.4 IMPORTANT PROPERTIES

ROUNDNESS

The Delaunay triangulation is “round,” that is, skinny simplices are avoided. This can be formalized in two dimensions by Lawson’s classic result: over all possible triangulations, the Delaunay triangulation maximizes the minimum angle of any triangle. No generalization using angles is known in higher dimension. However, define the *enclosing radius* of a simplex as the minimum radius of an enclosing sphere. In any dimension and over all possible triangulations of a point set, the Delaunay triangulation minimizes the maximum enclosing radius of any simplex [Raj94]. Also see [Section 25.4](#) on mesh generation.

OPTIMALITY

Fix a set S of sites in \mathbb{R}^d . For a triangulation T of S with simplices t_1, \dots, t_n , define

$$\begin{aligned} v_i &= \text{sum of squared vertex norms of } t_i \\ c_i &= \text{squared norm of barycenter of } t_i \\ a_i &= \text{volume of } t_i \\ s_i &= \text{sum of squared edge lengths of } t_i \\ r_i &= \text{circumradius of } t_i. \end{aligned}$$

Over all triangulations T of S , the Delaunay triangulation attains the unique minimum of the following functions, where κ is any positive real [Mus97]:

$$\begin{aligned} V(T) &= \sum_i v_i a_i \\ C(T) &= \sum_i c_i a_i \\ H(T) &= \sum_i s_i / a_i && d = 2 \text{ only} \\ R(T, \kappa) &= \sum_i r_i^\kappa && d = 2 \text{ only}. \end{aligned}$$

VISIBILITY DEPTH ORDERING

Choose a viewpoint v and a family of disjoint convex objects in \mathbb{R}^d . Object A is *in front of* object B from v if there is a ray starting at v that intersects A and then B in that order. Though an arbitrary family can have cycles in the “in front of” relation, the relation is acyclic for the faces of the Delaunay triangulation, for any viewpoint and any dimension [Ede90].

An application comes from computer graphics. The *painter's algorithm* renders 3D objects in back to front order, with later objects simply overpainting the image space occupied by earlier objects. A valid rendering order always exists if the “in front of” relation is acyclic, as is the case if the objects are Delaunay tetrahedra, or a subset of a set of Delaunay tetrahedra.

SUBGRAPH RELATIONSHIPS

The edges of a Delaunay triangulation form a graph DT whose vertices are the sites. In any dimension, the following subgraph relations hold:

$$EMST \subseteq RNG \subseteq GG \subseteq DT$$

where $EMST$ is the Euclidean minimum spanning tree, RNG is the relative neighborhood graph, and GG is the Gabriel graph. See [Section 51.2](#) on pattern recognition.

DILATION

A geometrically embedded graph G has *dilation* c if for any two vertices, the shortest path distance along the edges of G is at most c times the Euclidean distance between the vertices. In \mathbb{R}^2 , the edge set of the Delaunay triangulation has dilation at most ~ 2.42 ; with an equilateral-triangle convex distance function, the dilation is at most 2.

INTERPOLATION

Suppose each point site $s_i \in S \subset \mathbb{R}^d$ has an associated function value f_i . For $p \in \mathbb{R}^d$ define $\lambda_i(p)$ as the proportion of the area of s_i 's Voronoi cell that would be removed if p were added as a site. Then the *natural neighbor* interpolant $f(p) = \sum \lambda_i(p)f_i$ is C^0 , and C^1 except at sites. This construction can be generalized to give a C^k interpolant, any fixed k [HS02].

Alternatively, for a triangulation of S in \mathbb{R}^2 , consider the piecewise linear surface defined by linear interpolation over each triangle. Over all possible triangulations, the Delaunay triangulation minimizes the roughness of the resulting surface, where *roughness* is the square of the L_2 norm of the gradient of the surface, integrated over the triangulation [Rip90].

23.5 SOURCES AND RELATED MATERIAL

FURTHER READING

[Aur91, For95, AK00]: Survey papers that cover many aspects of Delaunay triangulations and Voronoi diagrams.

[OBS00]: A book entirely devoted to Voronoi diagrams, with an extensive discussion of applications.

[Ede87, PS85, dBK97]: Basic references for geometric algorithms.

www.voronoi.com, www.ics.uci.edu/~eppstein/gina/voronoi.html: Web sites devoted to Voronoi diagrams.

RELATED CHAPTERS

- [Chapter 22: Convex hull computations](#)
 - [Chapter 24: Arrangements](#)
 - [Chapter 25: Triangulations](#)
 - [Chapter 47: Algorithmic motion planning](#)
 - [Chapter 51: Pattern recognition](#)
-

REFERENCES

- [AA⁺95] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner. A novel type of skeleton for polygons. *J. Univer. Comput. Sci.*, 1:752–761, 1995.
- [Aur99] O. Aichholzer, F. Aurenhammer, D.Z. Chen, D.T. Lee, and E. Papadopoulou. Skew Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, 9:235–247, 1999.
- [AGS89] A. Aggarwal, L.J. Guibas, J.B. Saxe, and P.W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4:591–604, 1989.
- [AS00] P.K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: the union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24:645–685, 2000.
- [AG⁺98] G. Albers, L.J. Guibas, J.S.B. Mitchell, and T. Roos. Voronoi diagrams of moving points. *Internat. J. Comput. Geom. Appl.*, 8:365–379, 1998.
- [AB99] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22:481–504, 1999.
- [ABL03] D. Attali, J.-D. Boissonnat, and A. Lieutier. Complexity of the Delaunay triangulation of points on surfaces: the smooth case. *Proc. 19th. Annu. ACM Sympos. Comput. Geom.*, pages 201–210, 2003.
- [AS92] F. Aurenhammer and O. Schwarzkopf. A simple randomized incremental algorithm for computing higher order Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, 2:363–381, 1992.
- [Aur87] F. Aurenhammer. Power diagrams: properties, algorithms, and applications. *SIAM J. Comput.*, 16:78–96, 1987.
- [Aur91] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.
- [AK00] F. Aurenhammer and R. Klein. Voronoi diagrams. In J. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 201–290, Elsevier North-Holland, Amsterdam, 2000.
- [dBK97] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, New York, 1997.

- [Bur96] C. Burnikel. *Exact computation of Voronoi diagrams and line segment intersections*. Ph.D. Thesis, Universität des Saarlandes, 1996.
- [CD85] L.P. Chew and R.L. Drysdale. Voronoi diagrams based on convex distance functions. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 234–244, 1985.
- [Cha91] B. Chazelle. An optimal convex hull algorithm and new results on cuttings. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 29–38, 1991.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [CSY95] T.M. Chan, J. Snoeyink, and C.K. Yap. Output-sensitive construction of polytopes in four dimensions and clipped Voronoi diagrams in three. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 282–291, 1995.
- [CVY02] D. Cohen-Steiner, É. Colin de Verdière, and M. Yvinec. Conforming Delaunay triangulations in 3D. *Proc. 18th Annu. Sympos. Comput. Geom.*, pages 199–208, 2002.
- [DG03] T.K. Dey and S. Goswami. Tight Cocone: a watertight surface reconstructor. *Proc. 8th Annu. ACM Sympos. Solid Modeling Appl.*, 2003, pages 127–134.
- [Dwy91] R. Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.*, 6:343–367, 1991.
- [Dwy97] R. Dwyer. Voronoi diagrams of random lines and flats. *Discrete Comput. Geom.*, 17:123–136, 1997.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin, 1987.
- [Ede90] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimensions. *Combinatorica*, 10:251–260, 1990.
- [ET93] H. Edelsbrunner and T.-S. Tan. An upper bound for conforming Delaunay triangulations. *Discrete Comput. Geom.*, 10:197–213, 1993.
- [EE99] D. Eppstein and J. Erickson. Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. *Discrete Comput. Geom.*, 22:569–592, 1999.
- [Eri02] J. Erickson. Dense point sets have sparse Delaunay triangulations. *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 125–134, 2002.
- [For93] S.J. Fortune. Progress in computational geometry. In R. Martin, editor, *Directions in Geometric Computing*, pages 81–128. Information Geometers, Winchester, 1993.
- [For95] S.J. Fortune. Voronoi diagrams and Delaunay triangulations. In F. Hwang and D.Z. Du, editors, *Computing in Euclidean Geometry* (Second Edition), pages 225–265. World Scientific, Singapore, 1995.
- [GM01] C.I. Grima and A. Márquez. *Computational Geometry on Surfaces*. Kluwer Academic, Dordrecht, 2001.
- [HS99] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 26:2215–2256, 1999.
- [HS02] H. Hiyoshi and K. Sugihara. Improving continuity of Voronoi-based interpolation over Delaunay spheres. *Comp. Geom. Theory Appl.*, 22:167–183, 2002.
- [Hel01] M. Held. VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Comp. Geom. Theory Appl.*, 18:95–123, 2001.

- [HCK99] K.E. Hoff III, T. Culver, J. Keyser, M.C. Lin, and D. Manocha. Fast computation of generalized Voronoi diagrams using graphics hardware. *Proc. ACM Conf. SIGGRAPH 99*, pages 277–286, 1999.
- [KL93] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 124–132, 1993.
- [KMM93] R. Klein, K. Mehlhorn, and S. Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory Appl.*, 3:157–184, 1993.
- [KS03] V. Koltun and M. Sharir. Three dimensional Euclidean Voronoi diagrams of lines with a fixed number of orientations. *SIAM J. Computing*, 32:616–642, 2003.
- [KS04] V. Koltun and M. Sharir. Polyhedral Voronoi diagrams of polyhedra in three dimensions. *Discrete Comput. Geom.*, 31:83–124, 2004.
- [LS03] F. Labelle and J.R. Shewchuk. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. *Proc. 19th. Ann. ACM Sympos. Comput. Geom.*, pages 191–200, 2003.
- [LL00] G. Leibon and D. Letscher. Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 341–349, 2000.
- [Mus97] O.R. Musin. Properties of the Delaunay triangulation. *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 424–426, 1997.
- [OBS00] A. Okabe, B. Boots, K. Sugihara, and S.N. Chio. *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*, second edition. Wiley, Chichester, 2000.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry*. Springer-Verlag, New York, 1985.
- [Raj94] V.T. Rajan. Optimality of the Delaunay triangulation in R^d . *Discrete Comput. Geom.*, 12:189–202, 1994.
- [Rip90] S. Rippa. Minimal roughness property of the Delaunay triangulation. *Comput. Aided Design*, 7:489–497, 1990.
- [SD91] B. Schaudt and R.L. Drysdale. Multiplicatively weighted crystal growth Voronoi diagram. In *Proc. 7th. Annu. ACM Sympos. Comput. Geom.*, pages 214–223, 1991.
- [She98] J.R. Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations. *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 76–85, 1998.
- [She00] J.R. Shewchuk. Sweep algorithms for constructing higher-dimensional constrained Delaunay triangulations. *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 2000.
- [She03] J.R. Shewchuk. Updating and constructing constrained Delaunay and constrained regular triangulations by flips. *Proc. 19th. Annu. ACM Sympos. Comput. Geom.*, pages 181–190, 2003.
- [SII00] K. Sugihara, M. Iri, H. Inagaki, and T. Imai. Topology-oriented implementation—An approach to robust geometric algorithms. *Algorithmica*, 27:5–20, 2000.

24 ARRANGEMENTS

Dan Halperin

INTRODUCTION

Given a finite collection \mathcal{S} of geometric objects such as hyperplanes or spheres in \mathbb{R}^d , the *arrangement* $\mathcal{A}(\mathcal{S})$ is the decomposition of \mathbb{R}^d into connected open cells of dimensions $0, 1, \dots, d$ induced by \mathcal{S} . Besides being interesting in their own right, arrangements of hyperplanes have served as a unifying structure for many problems in discrete and computational geometry. With the recent advances in the study of arrangements of curved (algebraic) surfaces, arrangements have emerged as the underlying structure of geometric problems in a variety of “physical world” application domains such as robot motion planning and computer vision. This chapter is devoted to arrangements of hyperplanes and of curved surfaces in low-dimensional Euclidean space, with an emphasis on combinatorics and algorithms.

In the first section we introduce basic terminology and combinatorics of arrangements. In Section 24.2 we describe substructures in arrangements and their combinatorial complexity. Section 24.3 deals with data structures for representing arrangements and with special refinements of arrangements. The following two sections focus on algorithms: algorithms for constructing full arrangements are described in Section 24.4, and algorithms for constructing substructures in Section 24.5. Situations where arrangements have lower complexity than the general worst-case bounds are presented in Section 24.6. In Section 24.7 we discuss the relation between arrangements and other structures. Several applications of arrangements are reviewed in Section 24.8. Section 24.9 deals with robustness issues when implementing algorithms and data structures for arrangements and Section 24.10 surveys software implementation.

24.1 BASICS

In this section we review basic terminology and combinatorics of arrangements, first for arrangements of hyperplanes and then for arrangements of curves and surfaces.

24.1.1 ARRANGEMENTS OF HYPERPLANES

GLOSSARY

Arrangement of hyperplanes: Let \mathcal{H} be a finite set of hyperplanes in \mathbb{R}^d . The hyperplanes in \mathcal{H} induce a decomposition of \mathbb{R}^d (into connected open cells), the arrangement $\mathcal{A}(\mathcal{H})$. A d -dimensional cell in $\mathcal{A}(\mathcal{H})$ is a maximal connected region

of \mathbb{R}^d not intersected by any hyperplane in \mathcal{H} ; any k -dimensional cell in $\mathcal{A}(\mathcal{H})$, for $0 \leq k \leq d - 1$, is a maximal connected region in the intersection of a subset of the hyperplanes in \mathcal{H} that is not intersected by any other hyperplane in \mathcal{H} . It follows that any cell in an arrangement of hyperplanes is convex.

Simple arrangement: An arrangement $\mathcal{A}(\mathcal{H})$ of a set \mathcal{H} of n hyperplanes in \mathbb{R}^d , with $n \geq d$, is called simple if every d hyperplanes in \mathcal{H} meet in a single point and if any $d + 1$ hyperplanes have no point in common.

Vertex, edge, face, facet: 0, 1, 2, and $(d-1)$ -dimensional cell of the arrangement, respectively. (What we call *cells* here are in some texts referred to as *faces*.)

k -cell : A k -dimensional cell in the arrangement.

Combinatorial complexity of an arrangement: The overall number of cells of all dimensions in the arrangement.

EXAMPLE: AN ARRANGEMENT OF LINES

Let \mathcal{L} be a finite set of lines in the plane, and let $\mathcal{A}(\mathcal{L})$ be a simple arrangement induced by \mathcal{L} . A 0-dimensional cell (a vertex) is the intersection point of two lines in \mathcal{L} ; a 1-dimensional cell (an edge) is a maximal connected portion of a line in \mathcal{L} that is not intersected by any other line in \mathcal{L} ; and a 2-dimensional cell (a face) is a maximal connected region of \mathbb{R}^2 not intersected by any line in \mathcal{L} . See Figure 24.1.1.

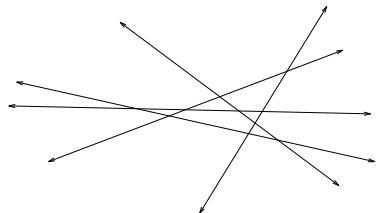


FIGURE 24.1.1

A simple arrangement of 5 lines.
It has 10 vertices, 25 edges (10 of which are unbounded),
and 16 faces (10 of which are unbounded).

COUNTING CELLS

A fundamental question in the study of arrangements is how complex a certain arrangement (or portion of it) can be. Answering this question is often a prerequisite to the analysis of algorithms on arrangements.

THEOREM 24.1.1

Let \mathcal{H} be a set of hyperplanes in \mathbb{R}^d . The maximum number of k -dimensional cells in the arrangement $\mathcal{A}(\mathcal{H})$, for $0 \leq k \leq d$, is

$$\sum_{i=0}^k \binom{d-i}{k-i} \binom{n}{d-i}.$$

The maximum is attained exactly when $\mathcal{A}(\mathcal{H})$ is simple.

We assume henceforth that the dimension d is a (small) constant. With few exceptions, we will not discuss *exact* combinatorial complexity bounds, as in the theorem above, but rather use the big-O notation. Theorem 24.1.1 implies the following:

COROLLARY 24.1.2

The maximum combinatorial complexity of an arrangement of n hyperplanes in \mathbb{R}^d is $O(n^d)$. If the arrangement is simple its complexity is $\Theta(n^d)$. In these bounds the constant of proportionality depends on d .

24.1.2 ARRANGEMENTS OF CURVES AND SURFACES

We now introduce more general arrangements, allowing for objects that are non-linear and/or bounded. We distinguish between planar arrangements and arrangements in three or higher dimensions. For planar arrangements we require only that the objects defining the arrangement be x -monotone Jordan arcs with a constant maximum number of intersections per pair. For arrangements of surfaces in three or higher dimensions we require that the surfaces be algebraic of constant maximum degree (a more precise definition is given below). This requirement simplifies the analysis and computation of such arrangements, and it does not seem to be too restrictive, as in most applications the arrangements that arise are of low-degree algebraic surfaces.

In both cases we typically assume that the objects (curves or surfaces) are in general position. This is a generalization to the current setting of the simplicity assumption for hyperplanes made above. (This assumption is reconsidered in Section 24.9.) All the other definitions in the Glossary carry over to arrangements of curves and surfaces.

PLANAR ARRANGEMENTS

Let $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ be a collection of Jordan arcs in the xy -plane, such that each arc is **x -monotone** (i.e., every line parallel to the y -axis intersects an arc in at most one point) and each pair of arcs in \mathcal{C} intersect in at most s points for some fixed constant s . The arrangement $\mathcal{A}(\mathcal{C})$ is the decomposition of the plane into open cells of dimensions 0, 1, and 2 induced by the arcs in \mathcal{C} . Here, a 0-dimensional cell (a vertex) is either an endpoint of one arc or an intersection point of two arcs. See Figure 24.1.2.

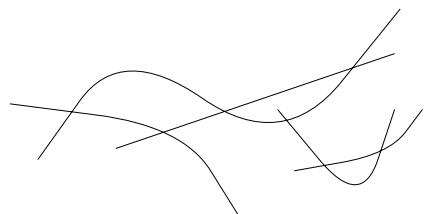


FIGURE 24.1.2

A simple arrangement of 5 bounded arcs, where $s = 2$. It has 17 vertices (10 of which are arc endpoints), 19 edges, and 4 faces (one of which is unbounded).

We assume that the arcs in \mathcal{C} are in **general position**, namely, that each intersection of a pair of arcs in \mathcal{C} is either a common endpoint or a transversal

intersection at a point in the relative interior of both arcs, and that no three arcs intersect at a common point.

THEOREM 24.1.3

If \mathcal{C} is a collection of n Jordan arcs as defined above, then the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{C})$ is $O(n^2)$. There are such arrangements whose complexity is $\Theta(n^2)$. In these bounds the constant of proportionality depends linearly on s .

THREE AND HIGHER DIMENSIONS

We denote the coordinate axes of \mathbb{R}^d by x_1, x_2, \dots, x_d . For a collection $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ of (hyper)surface patches in \mathbb{R}^d we make the following assumptions:

1. Each surface patch is contained in an algebraic surface of constant maximum degree.
2. The boundary of each surface patch is determined by at most some constant number of algebraic surface patches of constant maximum degree each. (Formally, each surface patch is a semialgebraic set of \mathbb{R}^d defined by a Boolean combination of a constant number of d -variate polynomial equalities or inequalities of constant maximum degree each.)
3. Every d surface patches in \mathcal{S} meet in at most s points.
4. Each surface patch is *monotone* in x_1, \dots, x_{d-1} , namely every line parallel to the x_d -axis intersects the surface patch in at most one point.
5. The surface patches in \mathcal{S} are in *general position*.

We use the simplified term *arrangement of surfaces* to refer to arrangements whose defining objects satisfy the assumptions above. A few remarks regarding these assumptions (see [AS00a, Section 2],[Mat02, Section 7.7],[Sha94], for detailed discussions of the required assumptions):

- Assumptions (1) and (2), together with the general position assumption (5), imply that every d -tuple of surfaces meet in at most some constant number of points. One can bound this number using Bézout's Theorem (see [Chapter 33](#)). The bound s on the number of d -tuple intersection points turns out to be a crucial parameter in the combinatorial analysis of substructures in arrangements. Often, one can get a better estimate for s than the bound implied by Bézout's theorem.
- Assumption (4) is used in results cited below. It can however be easily relaxed without affecting these results: If a surface patch does not satisfy this assumption, it can be decomposed into pieces that satisfy the assumption, and by assumptions (1) and (2) the number of these pieces will be bounded by a constant and their boundaries will satisfy assumption (2).
- Assumption (5) often does not affect the worst-case combinatorial bounds obtained for arrangements or their substructures, because it can be shown that the asymptotically highest complexity is obtained when the surfaces are in general position [Sha94]. For algorithms, this assumption is more problematic. There are general relaxation methods but these seem to introduce new difficulties [Sei98] (see also [Section 24.9](#)).

THEOREM 24.1.4

Given a collection \mathcal{S} of n surfaces in \mathbb{R}^d , as defined above, the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{S})$ is $O(n^d)$. There are such arrangements whose complexity is $\Theta(n^d)$. The constant of proportionality in these bounds depends on d and on the maximum algebraic degree of the surfaces and of the polynomials defining their boundaries.

ARRANGEMENTS ON CURVED SURFACES

Although we do not discuss such arrangements directly in this chapter, many of the combinatorial and algorithmic results that we survey carry over to arrangements on curved surfaces with only slight adjustments. Arrangements on spheres are especially prevalent in applications. The ability to analyze or construct arrangements on curved surfaces is implicitly assumed and exploited in the results for arrangements of surfaces in Euclidean space, since we often need to consider the lower-dimensional arrangement induced on a surface by its intersections with all the other surfaces that define the arrangement.

ADDITIONAL TOPICS

We focus in this chapter on simple arrangements. We note, however, that non-simple arrangements raise interesting questions; see, for example, [Szé97]. Another noteworthy topic that we will not cover here is *combinatorial equivalence* of arrangements; see [Chapter 6](#) and [BLW⁺93].

24.2 SUBSTRUCTURES IN ARRANGEMENTS

A substructure in an arrangement (i.e., a portion of an arrangement), rather than the entire arrangement, may be sufficient to solve a problem at hand. Also, the analysis of several algorithms for constructing arrangements relies on combinatorial bounds for substructures. We survey substructures that are known in general to have significantly smaller complexity than that of the entire arrangement. For simplicity, some of the substructures are defined below only for the planar case.

GLOSSARY

Let \mathcal{C} be a collection of n x -monotone Jordan arcs as defined in Section 24.1.

Lower (upper) envelope: For this definition we regard each curve c_i in \mathcal{C} as the graph of a continuous univariate function $c_i(x)$ defined on an interval. The lower envelope Ψ of the collection \mathcal{C} is the pointwise minimum of these functions: $\Psi(x) = \min c_i(x)$, where the minimum is taken over all functions defined at x . (The lower envelope is the 0-level of the arrangement $\mathcal{A}(\mathcal{C})$; see below.) Similarly, the upper envelope of the collection \mathcal{C} is defined as the pointwise maximum of these functions. Lower and upper envelopes are completely symmetric structures, and from this point on we will discuss only lower envelopes.

Minimization diagram of \mathcal{C} : The subdivision of the x -axis into maximal intervals so that on each interval the same subset of functions attains the minimum. In \mathbb{R}^d we regard the surface patches in \mathcal{S} as graphs of functions in the variables x_1, \dots, x_{d-1} , the lower envelope is the pointwise minimum of these functions, and the minimization diagram is the subdivision of \mathbb{R}^{d-1} into maximal connected cells such that over the interior of each cell the lower envelope is attained by a fixed subset of \mathcal{S} .

Zone: For an additional curve γ , the collection of faces of the arrangement $\mathcal{A}(\mathcal{C})$ intersected by γ . See [Figure 24.4.1](#). In earlier works, the zone is sometimes called the **horizon**.

Single cell: In this section, a d -cell in an arrangement in \mathbb{R}^d .

Many cells (m cells): Any m distinct d -cells in an arrangement in \mathbb{R}^d .

Sides and borders: Let e be an edge in an arrangement of lines, and let l be the line containing e . The line l divides the plane into two halfplanes h_1, h_2 . We regard e as two-sided, and denote the two sides by (e, h_1) and (e, h_2) . The edge e is on the boundary of two faces f_1 and f_2 in the arrangement. e is said to be a **1-border** of either face, marked (e, f_1) and (e, f_2) , respectively. Similarly a vertex in a simple arrangement of lines has four sides, and it is a **0-border** of four faces. The definition extends to arrangements of hyperplanes in higher dimensions and to arrangements of curved surfaces.

k -level: We assume here, for simplicity, that the curves are unbounded; the definition can be extended to the case of bounded curves. A point p in the plane is said to be at level k , if there are exactly k curves in \mathcal{C} lying strictly below p (i.e., a relatively open ray emanating from p in the negative y direction intersects exactly k curves in \mathcal{C}). The level of an (open) edge e in $\mathcal{A}(\mathcal{C})$ is the level of any point of e . The k -level of $\mathcal{A}(\mathcal{C})$ is the closure of the union of edges of $\mathcal{A}(\mathcal{C})$ that are at level k ; see [Figure 24.2.1](#). The **at-most- k -level** of $\mathcal{A}(\mathcal{C})$, denoted $(\leq k)$ -level, is the union of points in the plane at level j , for $0 \leq j \leq k$. Different texts use slight variations of the above definitions. In particular, in some texts the ray is directed upwards thus counting the levels from top to bottom. k -levels in arrangements of hyperplanes are closely related (through *duality*, see [Section 24.7](#)) to k -sets in point configurations; see [Chapter 1](#).

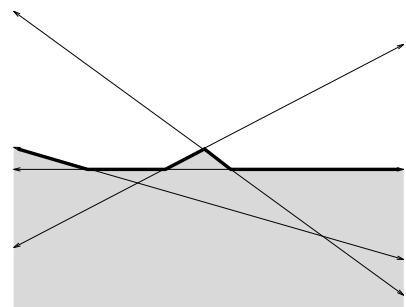


FIGURE 24.2.1

The bold polygonal line is the 2-level of the arrangement of four lines.

The shaded region is the (≤ 2) -level of the arrangement.

Union boundary: If each surface s in an arrangement in \mathbb{R}^d is the boundary of a d -dimensional object, then the boundary of the union of the objects is another interesting substructure. The study of the union boundary has largely been motivated by robot motion planning problems; for details see [Chapter 47](#).

$\alpha(n)$: The extremely slowly growing functional inverse of Ackermann's function.
See [Section 47.4](#).

MEASURING THE COMPLEXITY OF A SUBSTRUCTURE

For an arrangement in \mathbb{R}^d , if a substructure consists of a collection C of d -cells, its combinatorial complexity is defined to be the overall number of cells of any dimension on the boundary of each of the d -cells in C . This means that we count certain cells of the arrangement with multiplicity (as *borders* of the corresponding d -cells). For example, for the zone of a line l in an arrangement of lines, each edge of the arrangement that intersects l will be counted twice. However, since we assume that our arrangements reside in a fixed (low) dimensional space, this only implies a constant multiplicative factor in our count.

The complexity of the lower envelope of an arrangement is defined to be the complexity of its minimization diagram. In three or higher dimensions, this means that we count features that do not appear in the original arrangement. For example, in the lower envelope of a collection of triangles in 3-space, the projection of the edges of two distinct triangles may intersect in the minimization diagram although the two triangles are disjoint in 3-space.

The complexity of a k -level in an arrangement is defined in a similar way to the complexity of an envelope. The complexity of the $(\leq k)$ -level is defined as the overall number of cells of the arrangement that lie in the region of space whose points are at level at-most- k .

COMBINATORIAL COMPLEXITY BOUNDS FOR SUBSTRUCTURES

In the rest of this section we list bounds on the maximum combinatorial complexity of substructures. For lines, hyperplanes, Jordan arcs, and surfaces, these are arranged in Tables 24.2.1, [24.2.2](#), [24.2.3](#), and [24.2.4](#), respectively. Bounds in the tables using ϵ should read “for any $\epsilon > 0$ ” (with the implied constant of proportionality depending on ϵ). In the bounds for k -levels and $(\leq k)$ -levels we assume that $k \geq 1$ (otherwise one should use $k + 1$ instead of k). For each substructure, many special cases of arrangements have been considered and the results are too numerous to cover here. For an extensive recent review of results for k -levels see [Mat02, Chapter 11], for other substructures see [AS00a], [Mat02, Chapter 7].

TABLE 24.2.1 Substructures in arrangements of n lines in the plane.

SUBSTRUCTURE	BOUND	NOTES
Envelope	n edges	
Single face	n edges	
Zone of a line	$\Theta(n)$	See [Ede87] for an exact bound on the number of 0- and 1-borders
m faces	$\Theta(m^{2/3}n^{2/3} + m + n)$	Upper bound [CEG ⁺ 90]; lower bound [Ede87]
k -level	$O(nk^{1/3})$	[Dey98]
	$n2^{\Omega(\sqrt{\log k})}$	[Tot01]
$(\leq k)$ -level	$\Theta(nk)$	[AG86]

TABLE 24.2.2 Substructures in arrangements of n hyperplanes in \mathbb{R}^d .

SUBSTRUCTURE	BOUND	NOTES
Envelope	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	Upper bound theorem [McM70]
Single cell	$\Theta(n^{\lfloor \frac{d}{2} \rfloor})$	Upper bound theorem [McM70]
Zone of a hyperplane	$\Theta(n^{d-1})$	[ESS93]
Zone of p -dimensional algebraic surface (const max deg)	$O(n^{\lfloor (d+p)/2 \rfloor} \log^\gamma n)$	$\gamma = d + p \pmod{2}$ [APS93], the bound is almost tight in the worst case
m cells	$O(m^{1/2} n^{d/2} \log^{\lfloor \frac{d}{2} \rfloor - 1}/2 n)$	Bound is almost tight [AMS94]; see [AA92] for bounds on no. of facets
k -level, $d = 3$	$O(nk^{3/2})$	[SST01]
k -level, $d \geq 4$	$O(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil - \epsilon_d})$	[AAC98], constant $\epsilon_d > 0$
($\leq k$)-level	$\Theta(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil})$	[CS89]

CURVES

For a collection \mathcal{C} of n well-behaved curves as defined in Section 24.1, the complexity bounds for certain substructures involve functions related to *Davenport-Schinzel sequences*. The function $\lambda_s(n)$ is defined as the maximum length of a Davenport-Schinzel sequence of order s on n symbols, and it is almost linear in n for any fixed s . Davenport-Schinzel sequences play a central role in the analysis of substructures of arrangements of curves and surfaces. See [Section 47.4](#) for more details.

THEOREM 24.2.1

For a set \mathcal{C} of n x -monotone Jordan arcs such that each pair intersects in at most s points, the maximum number of intervals in the minimization diagram of \mathcal{C} is $\lambda_{s+2}(n)$. If the curves are unbounded, then the maximum number of intervals is $\lambda_s(n)$.

The connection between a zone and a single cell. As observed in [EGP⁺92], a bound on the complexity of a single cell in general arrangements of arcs implies the same asymptotic bound on the complexity of the zone of an additional well-behaved curve γ in the arrangement; “well-behaved” meaning that γ does not intersect any curve in \mathcal{C} more than some constant number of times. This observation extends to higher dimensions and is exploited in the result for zones in arrangements of surfaces [HS95a].

The results in [Table 24.2.3](#) are for Jordan arcs (bounded curves). There are slightly better bounds in the case of unbounded curves. For subquadratic bounds on k -levels in special arrangements of curves see [TT98], [Cha03], [NPP⁺02]. Improved bounds on the complexity of m faces in special arrangements of curves are given in [AEGS92] for segments, [AAS03] for circles, and [NPP⁺02] for pseudo circles and some other types of curves.

UNION BOUNDARY

For a collection of n Jordan regions (regions bounded by closed Jordan curves) such that each pair of bounding curves intersects at most twice, there are at most $6n - 12$

TABLE 24.2.3 Substructures in arrangements of n Jordan arcs.

SUBSTRUCTURE	BOUND	NOTES
Envelope	$\Theta(\lambda_{s+2}(n))$	See Theorem 24.2.1
Single face, zone m cells	$\Theta(\lambda_{s+2}(n))$ $O(m^{1/2}\lambda_{s+2}(n))$ $\Omega(m^{2/3}n^{2/3})$	[GSS89] [EGP ⁺ 92]
($\leq k$)-level	$\Theta(k^2\lambda_{s+2}(\lfloor \frac{n}{k} \rfloor))$	Lower bound for lines [Sha91]

TABLE 24.2.4 Substructures in arrangements of n surfaces.

OBJECTS	SUBSTRUCTURE	BOUND	NOTES
Surfaces in \mathbb{R}^d	Lower envelope	$O(n^{d-1+\epsilon})$	[HS94],[Sha94]
	Single cell, zone	$O(n^{d-1+\epsilon})$	[Bas98],[HS95a]
	($\leq k$)-level	$O(n^{d-1+\epsilon}k^{1-\epsilon})$	Combining [CS89] and Lower envelopes bound
$(d-1)$ -simplices in \mathbb{R}^d	Lower envelope	$\Theta(n^{d-1}\alpha(n))$	[Ede89]
	Single cell, zone	$O(n^{d-1} \log n)$	[AS94]
	Lower envelope, single cell	$\Theta(n^{\lceil \frac{d}{2} \rceil})$	Linearization

intersection points (for $n \geq 3$) between curves on the union boundary [KLPS86]. This bound is tight in the worst case. For variants and extensions of this result see [EGH⁺89], [PS99], [AEHS01].

Many of the interesting results in this area are for Minkowski sums where one of the operands is convex, motivated primarily by motion planning problems. These results are reviewed in [Chapter 47](#). We mention one exemplary result that (almost) settles a long-standing open problem: the complexity of the union boundary of n congruent infinite cylinders (namely, each cylinder is the Minkowski sum of a line in 3-space and a unit ball) is $O(n^{2+\epsilon})$ [AS00b].

Another family of results is for so-called *fat* objects. For example, a triangle is considered fat if all its angles are at least some fixed constant $\delta > 0$. For such triangles it is shown [MPS⁺94] that they determine at most a linear number of *holes* (namely connected components of the complement of the union) and that their union boundary has near-linear complexity. Typically (but not always) fatness precludes constructions with high union complexity, such as grid-like patterns with complexity $\Omega(n^d)$ in \mathbb{R}^d . For more results in the plane see [AFK⁺92], [ES00], [vK98]. For results in \mathbb{R}^3 and in higher dimensions consult [BSTY98], [PSS03].

ADDITIONAL COMBINATORIAL BOUNDS

The following bounds, while not bounds on the complexity of substructures, are useful in the analysis of algorithms for computing substructures and in obtaining other combinatorial bounds on arrangements.

Sum of squares. Let \mathcal{H} be a collection of n hyperplanes in \mathbb{R}^d . For each d -cell c of the arrangement $\mathcal{A}(\mathcal{H})$, let $f(c)$ denote the number of cells of any dimension on the boundary of c . Aronov et al. [AMS94] show that $\sum_c f^2(c) = O(n^d \log^{\lfloor \frac{d}{2} \rfloor - 1} n)$, where the sum extends over all d -cells of the arrangement. They use it to obtain bounds on the complexity of m cells in the arrangement. An application of the zone theorem [ESS93] implies a related bound: If we denote the number of hyperplanes appearing on the boundary of the cell c by $g(c)$, then $\sum_c f(c)g(c) = O(n^d)$, where the sum extends over all d -cells of the arrangement.

Overlay of envelopes. For two sets A and B of objects in \mathbb{R}^d , the complexity of the *overlay of envelopes* is defined as the complexity of the subdivision of \mathbb{R}^{d-1} induced by superposing the minimization diagram of A on that of B . Given two sets C_1 and C_2 , each of n x -monotone Jordan arcs, such that no pair of (the collection of $2n$) arcs intersects more than s times, the complexity of the overlay is easily seen to be $\Theta(\lambda_{s+2}(n))$. In 3-space, given two sets each of n well-behaved surfaces, the complexity of the overlay is $O(n^{2+\epsilon})$ [ASS96] (a simpler proof of the bound appears in [KS02]). The bound is applied to obtain a simple divide-and-conquer algorithm for computing the envelope in 3-space, and for obtaining bounds on the complexity of *transversals* (see [Chapter 4](#)). The bound in \mathbb{R}^4 is $O(n^{3+\epsilon})$ [KS02].

OPEN PROBLEMS

1. What is the complexity of the k -level in an arrangement of lines in the plane? For the gap between the known lower and upper bounds see [Table 24.2.1](#). This is a long-standing open problem in combinatorial geometry.
2. What is the complexity of m faces in an arrangement of well-behaved Jordan arcs? For lines a tight bound is known, whereas for curves a considerable gap still exists—see [Table 24.2.3](#).
3. What is the complexity of the boundary of the union of n infinite cylinders of different radii in 3-space? If all the radii are the same then the bound is $O(n^{2+\epsilon})$ [AS00b]. Also, what is the complexity of the union of n arbitrary cubes in 3-space? A near-quadratic bound is known only when the cubes are nearly equal [PSS03].

24.3 REPRESENTATIONS AND DECOMPOSITIONS

Before describing algorithms for arrangements in the next sections, we discuss how to represent an arrangement. The appropriate data structure for representing an arrangement depends on its intended use. Two typical ways of using arrangements are: (i) traversing the entire arrangement cell by cell; and (ii) directly accessing certain cells of the arrangement. We will present three structures, each providing a method for traversing the entire arrangement: the *incidence graph*, the *cell-tuple structure*, and the *complete skeleton*. We will then discuss refined representations that further subdivide an arrangement into subcells. These refinements are essential to allow for efficient access to cells of the arrangement. For algebraic geometry-oriented representations and decompositions see [Chapters 33](#) and [47](#).

GLOSSARY

Let \mathcal{S} be a collection of surfaces in \mathbb{R}^d (or curves in \mathbb{R}^2) as defined in Section 24.1, and $\mathcal{A}(\mathcal{S})$ the arrangement induced by \mathcal{S} . Let c_1 be a k_1 -dimensional cell of $\mathcal{A}(\mathcal{S})$ and c_2 a k_2 -dimensional cell of $\mathcal{A}(\mathcal{S})$.

Subcell, supercell: If $k_2 = k_1 + 1$ and c_1 is on the boundary of c_2 , then c_1 is a subcell of c_2 , and c_2 is a supercell of c_1 .

(-1)-dimensional cell, ($d+1$)-dimensional cell: Some representations assume the existence of two additional cells in an arrangement. The unique (-1) -dimensional cell is a subcell of every vertex (0-dimensional cell) in the arrangement, and the unique $(d+1)$ -dimensional cell is a supercell of all the d -dimensional cells in the arrangement.

Incidence: If c_1 is a subcell of c_2 , then c_1 and c_2 are *incident* to one another. We say that c_1 and c_2 define an *incidence*.

24.3.1 REPRESENTATIONS

INCIDENCE GRAPH

The incidence graph (sometimes called the *facial lattice*) of the arrangement $\mathcal{A}(\mathcal{S})$ is a graph $G = (V, E)$ where there is a node in V for every k -cell of $\mathcal{A}(\mathcal{S})$, $-1 \leq k \leq d + 1$, and an arc between two nodes if the corresponding cells are incident to one another (cf. Figure 16.1.3). For an arrangement of n surfaces in \mathbb{R}^d the number of nodes in V is $O(n^d)$ by Theorem 24.1.4. This is also a bound on the number of arcs in E : every cell (besides the (-1) -dimensional cell) in an arrangement $\mathcal{A}(\mathcal{S})$ in general position has at most a constant number of supercells. For an exact bound in the case of hyperplanes, see [Ede87, Section 1.2].

CELL-TUPLE STRUCTURE

While the incidence graph captures all the cells in an arrangement and (as its name implies) their incidence relation, it misses *order* information between cells. For example, there is a natural order among the edges that appear along the boundary of a face in a planar arrangement. This leads to the *cell-tuple structure* [Bri93] which is a generalization to any dimension of the two-dimensional doubly-connected-edge-list (DCEL) [dBvK⁺00] or the similar *quad-edge structure* of Guibas and Stolfi [GS85] and the 3D *facet-edge structure* of Dobkin and Laszlo [DL89]. The cell-tuple structure gives a simple and uniform representation of the incidence and ordering information in the arrangement.

SKELETON

Let \mathcal{H} be a finite set of hyperplanes in \mathbb{R}^d . A *skeleton* in the arrangement $\mathcal{A}(\mathcal{H})$ is a connected subset of edges and vertices of the arrangement. The *complete*

skeleton is the union of all the edges and vertices of the arrangement. Edelsbrunner [Ede87] proposes a representation of the skeleton as a digraph, which allows for a systematic traversal of the entire arrangement (in the case of a complete skeleton) or a substructure of the arrangement. Using a one-dimensional skeleton to represent an arrangement in an arbitrary-dimensional space is a notion that appears also in algebro-geometric representations. There, however, the skeleton, or **roadmap**, is far more complicated (indeed it represents more general arrangements); see [BPR00] and [Chapter 47](#).

24.3.2 DECOMPOSITIONS

A raw arrangement may still be an unwieldy structure as cells may have complicated shapes and many bounding subcells. It is often desirable to decompose the cells of the arrangement into subcomponents so that each subcomponent has a constant descriptive complexity and is homeomorphic to a ball. Besides the obvious convenience that such a decomposition offers (just like a triangulation of a simple polygon), it turns out to be crucial to the design and analysis of randomized algorithms for arrangements, as well as to combinatorial analysis of arrangements.

For a decomposition to be useful, we aim to add as few extra features as possible. The three decompositions described in this section have the property that the complexity of the decomposed arrangement is asymptotically close to (sometimes the same as) that of the original arrangement. (This is still not known for the *vertical decomposition* in higher dimensions—see the open problem below.)

BOTTOM VERTEX DECOMPOSITION OF HYPERPLANE ARRANGEMENTS

Consider an arrangement of lines $\mathcal{A}(\mathcal{L})$ in the plane. For a face f let $v_b = v_b(f)$ be the bottommost vertex of f (the vertex with lowest y coordinate, ties can be broken by the lexicographic ordering of the coordinate vectors of the vertices). Extend an edge from v_b to each vertex on the boundary of f that is not incident to an edge incident to v_b ; see Figure 24.3.1. Repeat for all faces of $\mathcal{A}(\mathcal{L})$ (unbounded faces require special care). The original arrangement, together with the added edges, constitutes the **bottom vertex decomposition** of $\mathcal{A}(\mathcal{L})$, which is a decomposition of $\mathcal{A}(\mathcal{L})$ into triangles. The notion extends to arrangements of hyperplanes in higher dimensions, and it is carried out recursively [Cla88]. The combinatorial complexity of the decomposition is asymptotically the same as that of the original arrangement.

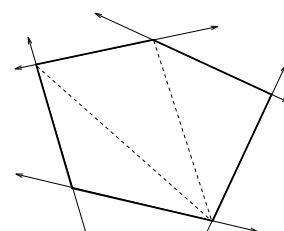


FIGURE 24.3.1

The bottom vertex decomposition of a face in an arrangement of lines.

VERTICAL DECOMPOSITION

The bottom vertex decomposition does not in general extend to arrangements of nonlinear objects. Fortunately there is an alternative, rather simple, decomposition method that applies to almost any reasonable arrangement. This is the *vertical decomposition* or *trapezoidal decomposition*. See Figure 24.3.2. It is optimal for two-dimensional arrangements, namely its complexity is asymptotically the same as that of the underlying arrangement. It is near-optimal in three and four dimensions. In higher dimensions it is still the general decomposition method that is known to have the best (lowest) complexity.

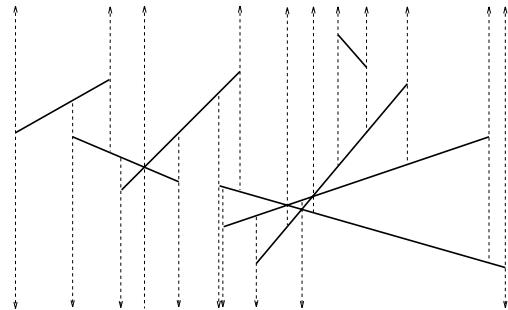


FIGURE 24.3.2

The vertical decomposition of an arrangement of segments: a vertical line segment is extended upward and downward from each vertex of the arrangement until it either hits another segment or extends to infinity.

The extension to higher dimensions is defined recursively and is presented in full generality in [CEGS91]. For details of the extension to three dimensions, see [CEG⁺90] for the case of spheres, and [dBGH96] for the case of triangles. The four-dimensional case is studied in [Kol01a], [Kol01b]. Table 24.3.1 summarizes the bounds on the maximum combinatorial complexity of the vertical decomposition for several types of arrangements and substructures. Certain assumptions that curves and surfaces are “well-behaved” are not detailed.

TABLE 24.3.1 Combinatorial bounds on the maximum complexity of the vertical decomposition of n objects.

OBJECTS	BOUND	NOTES
Curves in \mathbb{R}^2	$\Theta(K)$	K is the complexity of \mathcal{A}
Surfaces in \mathbb{R}^d	$O(n^{2d-4+\epsilon})$	[CEGS91], [Kol01a]
Triangles in \mathbb{R}^3	$\Theta(n^3)$	[dBGH96]
Triangles in \mathbb{R}^3	$O(n^2\alpha(n)\log n + K)$	K is the complexity of \mathcal{A} [Tag96]
Surfaces in \mathbb{R}^3 , single cell	$O(n^{2+\epsilon})$	[SS97]
Surfaces in \mathbb{R}^3 , $(\leq k)$ -level	$O(n^{2+\epsilon}k)$	See [AES99] for refined bounds
Hyperplanes in \mathbb{R}^4	$\Theta(n^4)$	[Kol01b]
Simplices in \mathbb{R}^4	$O(n^4\alpha(n)\log n)$	[Kol01b]

OTHER DECOMPOSITION SCHEMES

Aronov and Sharir devised alternative decomposition methods for arrangements of simplices [AS90], [AS94]. These are more involved than the decompositions described above and we omit their description here. These methods were instrumental in obtaining improved combinatorial bounds and efficient algorithms for arrangements of simplices. A sparse variant of the vertical decomposition is proposed in [SH02] for the case of triangles in 3-space: it produces fewer 3D cells (which are convex but may have many bounding facets) and its computation requires simpler geometric primitives than the standard vertical decomposition—this is advantageous from a practical (implementation) point of view. Yet another decomposition scheme has been devised for surfaces arising in the study of polygonal motion planning in translation and rotation [HS96].

CUTTINGS

All the decompositions described so far have the property that each cell of the decomposition lies fully in a single cell of the arrangement. In various applications this property is not required and other decomposition schemes may be applied, such as *cuttings* ([Chapter 36](#)). Cuttings are the basis of efficient divide-and-conquer algorithms for numerous geometric problems on arrangements and otherwise.

STRUCTURES FOR POINT LOCATION AND RAY SHOOTING

To access certain cells of an arrangement without traversing the entire arrangement, we need more elaborate structures than those described above. See [Chapters 34](#) and 37 for details.

OPEN PROBLEMS

1. Obtain an improved combinatorial bound on the complexity of the vertical decomposition of arrangements of surfaces in five and higher dimensions. Such a result would have a wide-ranging effect on other combinatorial bounds, on algorithms, and on a variety of applications of arrangements.
2. The decompositions described above are asymptotically efficient. However it has been observed that the constant factors in the complexity bounds are highly noticeable in practice. It is desirable to devise alternative *sparser* decompositions, namely decompositions that add fewer extra features such that they will still have some of the favorable properties of say the vertical decomposition. For steps in this direction, cf. [HP00], [SH02].

24.4 ALGORITHMS FOR ARRANGEMENTS

This section covers constructing an arrangement: producing a representation of an arrangement in one of the forms described in the previous section (or in a similar

form). We distinguish between algorithms for the construction of the entire arrangement (surveyed in this section), and algorithms for constructing substructures of an arrangement (in the next section). We start with deterministic algorithms and then describe randomized ones.

MODEL OF COMPUTATION

We assume the standard model in computational geometry: infinite precision real arithmetic [PS85]. For algorithms computing arrangements of curves or surfaces, we further assume that certain operations on a small number of curves or surfaces each take unit time. For algebraic curves or surfaces, the unit cost assumption for these operations is theoretically justified by results on the solution of sets of polynomial equations; see [Chapter 33](#). When implementing algorithms for arrangements some of these assumptions need to be reconsidered—see [Sections 24.9](#) and [24.10](#).

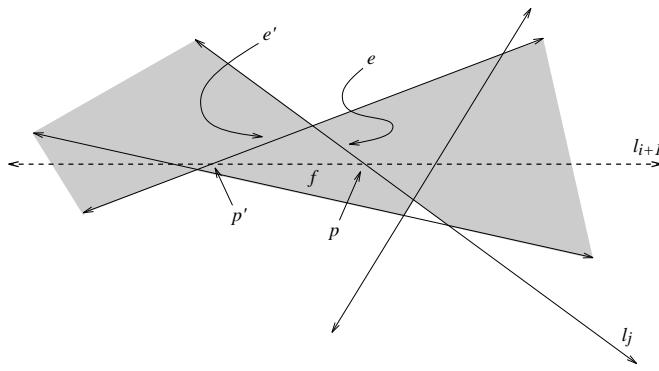
24.4.1 DETERMINISTIC ALGORITHMS

Incremental construction. The incremental algorithm proceeds by adding one object after the other to the arrangement while maintaining (a representation of) the arrangement of the objects added so far. This approach yields an optimal-time algorithm for arrangements of hyperplanes. The analysis of the running time is based on the zone result [ESS93] (Section 24.2). We describe it next for a collection $\mathcal{L} = \{l_1, \dots, l_n\}$ of n lines in the plane, assuming that the arrangement $\mathcal{A}(\mathcal{L})$ is simple.

FIGURE 24.4.1

Adding the line l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$.

The shaded region is the zone of l_{i+1} in the arrangement of the other four lines.



Let \mathcal{L}_i denote the set $\{l_1, \dots, l_i\}$. At stage $i + 1$ we add l_{i+1} to the arrangement $\mathcal{A}(\mathcal{L}_i)$. We maintain the DCEL representation (Section 24.3.1) for $\mathcal{A}(\mathcal{L}_i)$, so that in addition to the incidence information, we also have the order of edges along the boundary of each face. The addition of l_{i+1} is carried out in two steps: (i) we find a point p of intersection between l_{i+1} and an edge of $\mathcal{A}(\mathcal{L}_i)$ and split that edge into

two, and (ii) we walk along l_{i+1} from p to the left (assuming l_{i+1} is not vertical) updating $\mathcal{A}(\mathcal{L}_i)$ as we go; we then walk along l_{i+1} from p to the right completing the construction of $\mathcal{A}(\mathcal{L}_{i+1})$. See Figure 24.4.1.

Finding an edge of $\mathcal{A}(\mathcal{L}_i)$ that l_{i+1} intersects can be done in $O(i)$ time by choosing one line l_j from \mathcal{L}_i and checking all the edges of $\mathcal{A}(\mathcal{L}_i)$ that lie on l_j for intersection with l_{i+1} . This intersection point p lies on an edge e that borders two faces of $\mathcal{A}(\mathcal{L}_i)$. We split e into two edges at p . Next, consider the face f intersected by the part of l_{i+1} to the left of p . Using the order information, we walk along the edges of f away from p and we check for another intersection p' of l_{i+1} with an edge e' on the boundary of f . At the intersection we split e' into two edges, we add an edge to the arrangement for the portion $\overline{pp'}$ of l_{i+1} , and we move to the face on the other (left) side of e' . Once we are done with the faces of $\mathcal{A}(\mathcal{L}_i)$ crossed by l_{i+1} to the left of p , we go back to p and walk to the other side. This way we visit all the faces of the zone of l_{i+1} in $\mathcal{A}(\mathcal{L}_i)$, as well as some of its edges. The amount of time spent is proportional to the number of edges we visit, and hence bounded by the complexity of the zone. The space required for the algorithm is the space to maintain the DCEL structure. The same approach extends to higher dimensions. For details see [Ede87, Chapter 7] (note that the algorithm as described in [Ede87] uses the incidence graph for maintaining the arrangement).

THEOREM 24.4.1

If \mathcal{H} is a set of n hyperplanes in \mathbb{R}^d such that $\mathcal{A}(\mathcal{H})$ is a simple arrangement, then $\mathcal{A}(\mathcal{H})$ can be constructed in $\Theta(n^d)$ time and space.

The time and space required by the algorithm are clearly optimal. However, it turns out that for arrangements of lines one can do better in terms of *working space*. This is explained below in the subsection *topological sweep*. See [Goo93], [HJW90] for parallel algorithms for arrangements of hyperplanes.

The incremental approach can be applied to constructing arrangements of curves, using the vertical decomposition of the arrangement [EGP⁺92]:

THEOREM 24.4.2

Let \mathcal{C} be a set of n Jordan arcs as defined in Section 24.1. The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed in $O(n\lambda_{s+2}(n))$ time using $O(n^2)$ space.

Sweeping over the arrangement. The sweep paradigm, a fundamental paradigm in computational geometry, is also applicable to constructing arrangements. For planar arrangements, its worst-case running time is slightly inferior to that of the incremental construction described above. It is, however, output sensitive.

THEOREM 24.4.3

Let \mathcal{C} be a set of n Jordan arcs as defined in Section 24.1. The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed in $O((n+k)\log n)$ time and $O(n+k)$ space, where k is the number of intersection points in the arrangement.

One can similarly sweep a plane over an arrangement of surfaces in \mathbb{R}^3 . There is an output-sensitive algorithm for constructing the vertical decomposition of an arrangement of n surfaces that runs in time $O(n \log^2 n + V \log n)$, where V is the combinatorial complexity of the vertical decomposition. For details see [SH02].

Topological sweep. Edelsbrunner and Guibas [EG89] devised an algorithm for

constructing an arrangement of lines that requires only linear working storage and still runs in optimal $O(n^2)$ time. Instead of sweeping the arrangement with a straight line, they sweep it with a pseudoline that serves as a “topological wave-front.”

The most efficient deterministic algorithm for computing the intersections in a collection of well-behaved curves is due to Balaban [Bal95]. It runs in $O(n \log n + k)$ time and requires $O(n)$ working storage. The algorithm does *not* construct the arrangement; it only finds the intersection points, unsorted.

24.4.2 RANDOMIZED ALGORITHMS

Most randomized algorithms for arrangements follow one of two paradigms: (i) incremental construction or (ii) divide-and-conquer using random sampling. The randomization in these algorithms is in choices made by the algorithm; for example, the order in which the objects are handled in an incremental construction. In the expected performance bounds, the expectation is with respect to the random choices made by the algorithm. We do not make any assumptions about the distribution of the objects in space. See also [Chapter 40](#).

In constructing a full arrangement, these two paradigms are rather straightforward to apply. Most of these algorithms use an efficient decomposition as discussed in Section 24.3.

Incremental construction. Here the randomization is in the order that the objects defining the arrangement are inserted. For the construction of an arrangement of curves, the algorithm is similar to the deterministic construction mentioned above.

THEOREM 24.4.4 *[Mul93]*

Let \mathcal{C} be a set of n Jordan arcs as defined in Section 24.1. The arrangement $\mathcal{A}(\mathcal{C})$ can be constructed by a randomized incremental algorithm in $O(n \log n + k)$ expected time and $O(n + k)$ expected space, where k is the number of intersection points in the arrangement.

Divide-and-conquer by random sampling. For a set \mathcal{V} of n objects in \mathbb{R}^d the paradigm is: choose a subset \mathcal{R} of the objects at random, construct the arrangement $\mathcal{A}(\mathcal{R})$, decompose it further into constant complexity components (using, for example, one of the methods described in Section 24.3), and recursively construct the portion of the arrangement in each of the resulting components. Then glue all the substructures together into the full arrangement. The theory of random sampling is then used to show that with high probability the size of each subproblem is considerably smaller than that of the original problem, and thus efficient resource bounds can be proved.

The divide-and-conquer counterpart of Theorem 24.4.4 is due to Amato et al. [AGR00]. It has the same running time, and uses slightly more space (or exactly the same space for the case of segments).

The result stated in the following theorem is obtained by applying this paradigm to arrangements of algebraic surfaces and it is based on the vertical decomposition of the arrangement.

THEOREM 24.4.5 [CEGS91], [Kol01a]

Given a collection \mathcal{S} of n algebraic surfaces in \mathbb{R}^d as defined in Section 24.1, a data structure of size $O(n^{2d-4+\epsilon})$ for the arrangement $\mathcal{A}(\mathcal{S})$ can be constructed in $O(n^{2d-4+\epsilon})$ time for any $\epsilon > 0$, so that a point-location query can be answered in $O(\log n)$ time. In these bounds the constant of proportionality depends on ϵ , the dimension d , and the maximum algebraic degree of the surfaces and their boundaries.

If only traversal of the entire arrangement is needed, it is plausible that a simpler structure such as the incidence graph could be constructed using less time and storage space, close to $O(n^d)$ for both. See [Can93],[BPR00] for algebro-geometric methods.

Derandomization. Techniques have been proposed to derandomize many randomized geometric algorithms, often without increase in their asymptotic running time; see [Chapter 40](#). However, in most cases the randomized versions are conceptually much simpler and hence may be better candidates for efficient implementation.

24.4.3 OTHER ALGORITHMIC ISSUES

For algebro-geometric tools, see [Chapter 33](#). See [Chapter 41](#)(and [Section 24.9](#)) for a discussion of precision and degeneracies. Parallel algorithms are discussed in [Chapter 42](#).

24.5 CONSTRUCTING SUBSTRUCTURES

ENVELOPE AND SINGLE CELL IN ARRANGEMENTS OF HYPERPLANES

Computing a single cell or an envelope in an arrangement of hyperplanes is equivalent (through duality) to computing the convex hull of a set of points in \mathbb{R}^d ([Chapter 22](#)).

Using linearization [AM94], we can solve these problems for arrangements of spheres in \mathbb{R}^d . We first transform the spheres into hyperplanes in \mathbb{R}^{d+1} , and then solve the corresponding problems in \mathbb{R}^{d+1} .

LOWER ENVELOPE

The lower envelope of a collection of n well-behaved curves (where each pair intersect in at most s points) can be computed by a simple divide-and-conquer algorithm that runs in time $O(\lambda_{s+2}(n) \log n)$ and requires $O(\lambda_{s+2}(n))$ storage. Hershberger [Her89] devised an improved algorithm that runs in time $O(\lambda_{s+1}(n) \log n)$; in particular, for the case of line segments, it runs in optimal $O(n \log n)$ time. In 3-space, Agarwal et al. [ASS96] showed that a simple divide-and-conquer scheme can be used to compute the envelope of n surfaces in time $O(n^{2+\epsilon})$. This is an application of the bound on the complexity of the overlay of envelopes cited in Section 24.2. Boissonnat and Dobrindt give a randomized incremental algorithm for computing the envelope [BD96]. There are efficient algorithms for computing the envelope of $(d-1)$ -simplices in \mathbb{R}^d (see [EGS89] for the algorithm in 3D which can be efficiently extended to higher dimensions), and an efficient data structure

for point location in the minimization diagram of surfaces in \mathbb{R}^4 . Output-sensitive construction of the envelope of triangles in \mathbb{R}^3 has been mainly studied in relation to hidden-surface-removal (see [dB93]). Partial information of the minimization diagram (vertices, edges and 2-cells) can be computed efficiently for arrangements of surfaces in any fixed dimension [AAS97].

SINGLE CELL AND ZONE

All the results cited below for a single cell in arrangements of bounded objects hold for the zone problem as well (see the remark in Section 24.2 on the connection between the problems).

Computing a single face in an arrangement of n Jordan arcs as defined in Section 24.1 can be accomplished in worst-case near-optimal time: deterministically in $O(\lambda_{s+2}(n) \log^2 n)$ time, and using randomization in $O(\lambda_{s+2}(n) \log n)$ time [SA95].

In three dimensions, Schwarzkopf and Sharir [SS97] give an algorithm with running time $O(n^{2+\epsilon})$ for any $\epsilon > 0$ to compute a single cell in an arrangement of n well-behaved surfaces. Algorithms with improved running time to compute a single cell in 3D arrangements are known for arrangements of surfaces induced by certain motion planning problems [Hal92], [Hal94], and for arrangements of triangles [dBDS95].

LEVELS

In an arrangement of n lines in the plane, the k -level can be computed in $O((n + f) \log n)$ time, where f is the combinatorial complexity of the k -level—the bound is for the algorithm described in [EW86] while using the data structure in [BJ02] which in turn builds on ideas in [Cha01]. For computing the k -level in an arrangement of hyperplanes in \mathbb{R}^d see [AM95],[Cha96].

The $(\leq k)$ -level in arrangements of lines can be computed in worst-case optimal time $O(n \log n + kn)$ [ERvK96]. Algorithms for computing the $(\leq k)$ -level in arrangements of Jordan arcs are described in [AdB⁺98], the $(\leq k)$ -level in arrangements of planes in \mathbb{R}^3 (in optimal $O(n \log n + k^2 n)$ expected time) in [Cha00], and in arrangements of surfaces in \mathbb{R}^3 in [AES99].

UNION BOUNDARY

For a given family of planar regions bounded by well-behaved curves, let $f(m)$ be the maximum complexity of the union boundary of a collection of m objects of the family. Then the union of n such objects can be constructed deterministically in $O(f(n) \log^2 n)$ time or by a randomized incremental algorithm in expected $O(f(n) \log n)$ time [dBDS95]. A slightly faster algorithm for the case of fat triangles is given in [MMP⁺91]. An efficient randomized algorithm for computing the union of convex polytopes in \mathbb{R}^3 is given in [AST97].

MANY CELLS

There are efficient algorithms (deterministic and randomized) for computing a set

of selected faces in arrangements of lines or segments in the plane. These algorithms are nearly worst-case optimal [AMS98]. Algorithms for arrangements of planes are described in [EGS90], and for arrangements of triangles in 3-space in [AS90].

The related issue of computing the incidences between a set of objects (lines, unit circles) and a set of points is dealt with in [Mat93], with results that extend to higher dimensions [AS00a]. Generally, the bounds for the running time are roughly the same as those for the number of incidences. For lower bounds for the related *Hopcroft's problem* see [Eri96], [BK03].

OPEN PROBLEMS

Devise efficient algorithms for computing:

1. The lower envelope of an arrangement of surfaces in five and higher dimensions; for an algorithm that computes partial information see [AAS97].
2. A single cell in an arrangement of surfaces in four and higher dimensions; for a worst-case near-optimal algorithm in three dimensions see [SS97].

24.6 SPARSE ARRANGEMENTS

So far we have discussed arrangements of n objects in \mathbb{R}^d where each object has constant descriptive complexity and the total complexity of the entire arrangement can be $\Omega(n^d)$ in the worst case. In many situations arrangements do not achieve this worst-case complexity, or there are additional parameters that control the complexity of the arrangement. In this section we survey several such situations.

Let \mathcal{C} be a collection of n Jordan arcs, where each pair of arcs in \mathcal{C} intersects at most a constant number of times, and with the additional condition that any vertical line intersects at most k of the curves in \mathcal{C} . In this case the maximum combinatorial complexity of the arrangement $\mathcal{A}(\mathcal{C})$ is $\Theta(nk)$. For an application of this result and for more results on arrangements with low *vertical stabbing number* (the number of objects stabbed by any vertical line) see [dBH⁺97].

A general way to take advantage of reduced complexity of an arrangement is to construct the arrangement using an output-sensitive algorithm. However, by understanding the source of the reduced complexity it may be possible to devise algorithms that perform better than general-purpose output-sensitive algorithms. In several cases this has indeed been achieved. The collection of atom spheres in the geometric model of molecules exhibits sparseness properties that have led to improved combinatorial bounds and relatively simple algorithms. These algorithms have been implemented and perform well in practice [HO98], [HS98]. Another area where results of this nature have been obtained is robot motion planning among *fat obstacles*; see Section 47.3.

ARRANGEMENTS OF CONVEX POLYTOPES

Consider the subdivision of 3-space induced by k convex polytopes with a total of n vertices. To bound the complexity of this arrangement we can regard this

as an arrangement of $O(n)$ triangles in 3-space, implying an upper bound $O(n^3)$. However, the complexity of such an arrangement is shown in [dBH⁺97] to be only $O(nk^2)$. More generally Aronov et al. [ABE91] showed that the complexity of an arrangement of k convex polytopes in \mathbb{R}^d with a total of n facets is $\Theta(n^{\lfloor \frac{d}{2} \rfloor} k^{\lceil \frac{d}{2} \rceil})$.

A useful substructure in an arrangement of convex polytopes is the collection of ***maximally covered cells***, namely cells of the arrangement that are covered by more polytopes than any other cell in their immediate neighborhood [GHH⁺98]. The ability to access these cells efficiently has led to an efficient and practical algorithm to test whether an object consisting of polyhedral parts is interlocked (i.e., cannot be taken apart with two hands).

24.7 RELATION TO OTHER STRUCTURES

Arrangements relate to a variety of additional structures. Since the machinery for analyzing and computing arrangements is rather well developed, problems on related structures are often solved by first constructing (or reasoning about) the corresponding arrangement.

Using *duality* one can transform a set (or *configuration*) of points in \mathbb{R}^d (the primal space) into a set of hyperplanes in \mathbb{R}^d (the dual space) and vice versa. Different duality transforms are advantageous in different situations [O'R98]. Edelsbrunner [Ede87, Chapter 12] describes a collection of problems stated for point configurations and solved by operating on their corresponding dual arrangements. An example is given in the next section. See also [Chapter 1](#).

Plücker coordinates are a tool that enables one to treat k -flats in \mathbb{R}^d as points or hyperplanes in a possibly different (higher) dimensional space. This has been taken advantage of in the study of families of lines in 3-space—see [Chapter 37](#).

Lower envelopes (or more generally k -levels in arrangements) relate to Voronoi diagrams—see [Chapter 23](#).

For the connection of arrangements to polytopes and zonotopes see [Ede87] and Section 16.1.4 of this Handbook. For the connection to oriented matroids see [Chapter 6](#).

24.8 APPLICATIONS

A typical application of arrangements is for solving a problem on related structures. We first transform the original structure (e.g., a point configuration) into an arrangement and then solve the problem on the resulting arrangement. See Section 24.7 above and [Chapters 1, 23, and 37](#).

EXAMPLE: MINIMUM AREA TRIANGLE

Let P be a set of n points in the plane. We wish to find three points of P such that the triangle that they define has minimum area. We use the duality transform that maps a point $p := (a, b)$ to the line $p^* := (y = ax - b)$, and maps a line $l := (y = cx + d)$ to the point $l^* := (c, -d)$. One can show that if we fix two points

$p_i, p_j \in P$, and the line p_k^* has the smallest vertical distance to the intersection point $p_i^* \cap p_j^*$ among all other lines in $P^* = \{p^* | p \in P\}$, then the point p_k defines the minimum area triangle with the fixed points p_i, p_j over all points in $P \setminus \{p_i, p_j\}$. Finding the triple of lines as above (an intersecting pair and the other line closest to the intersection) is easy after constructing the arrangement $\mathcal{A}(P^*)$ (Section 24.4), and can be done in $\Theta(n^2)$ time in total. This is the most efficient algorithm known for this problem [GO95]. The minimum volume simplex defined by $d + 1$ points in a set of n points in \mathbb{R}^d can be found using arrangements of hyperplanes in $\Theta(n^d)$ time.

OTHER APPLICATIONS

Another strand of applications consists of the “robotic” or “physical world” applications [HS95b]. In these problems a continuous space is decomposed into a finite number of cells so that in each cell a certain invariant is maintained. Here, arrangements are used to discretize a continuous space without giving up the completeness or exactness of the solution. An example of an application of this kind solves the following problem: Given a convex polyhedron in 3-space, determine how many combinatorially distinct orthographic and perspective views it induces; see Table 25.6.3. The answer is given using an arrangement of circles on the sphere (for orthographic views) and an arrangement of planes in 3-space (for perspective views) [BD90].

Many developments in the study of arrangements of curves and surfaces have been primarily motivated by problems in robot motion planning ([Chapter 47](#)) and several of its variants ([Chapter 48](#)). For example, the most efficient algorithm known for computing a collision-free path for an arbitrary polygonal robot (not necessarily convex) moving by translation and rotation among polygonal obstacles in the plane is based on computing a single connected component in an arrangement of surfaces in 3-space. The problem of planning a collision-free motion for a robot among obstacles is typically studied in the *configuration space* where every point represents a possible configuration of the robot. The related arrangements are of surfaces that represent all the contact configurations between the boundary of the robot and the boundaries of obstacles and thus partition configuration space into free cells (describing configurations where the robot does not intersect any obstacle) and forbidden cells. Given the initial (free placement) of the robot, we need only explore the cell that contains this initial configuration in the arrangement.

A concept similar to configuration space of motion planning has been applied in assembly planning (Section 48.3). The assembly planning problem is converted into a problem in *motion space* where every point represents an allowed path (motion) of a subcollection of the assembly relative the rest of the assembly [HLW00]. The motion space is partitioned by a collection of constraint surfaces such that for all possible motions inside a cell of the arrangement, the collection of movable subsets of the assembly is invariant.

As mentioned earlier, arrangements on spheres are prevalent in applications. Aside from vision applications, they also occur in: computer-assisted radio-surgery [SAL93], molecular modeling [HS98], assembly planning (Section 48.3), manufacturing [AdB⁺02], and more.

Arrangements have been used to solve problems in many other areas including geometric optimization [AS98], range searching ([Chapter 36](#)), statistical analysis

([Chapter 57](#)), and micro robotics [BDH99], to name a few. More applications can be found in the sources cited below and in several other chapters in this book.

24.9 ROBUSTNESS

Transforming the data structures and algorithms described above into effective computer programs is a difficult task. The typical assumptions of (i) the real RAM model of computation and (ii) general position, are not realistic in practice. This is not only a problem for implementing software for arrangements but rather a general problem in computational geometry (see [Chapter 65](#)). However, it is especially acute in the case of arrangements since here one needs to compute *intersection points* of curves and surfaces and use the computed values in further operations (to distinguish from say convex hull algorithms that only select a subset of the input points).

EXACT COMPUTING

A general paradigm to overcome robustness problems is to compute exactly. For arrangements of linear objects, namely, arrangements of hyperplanes or of simplices, there is a fairly straightforward solution: using arbitrary precision rational arithmetic. This is regularly done by keeping arbitrary long integers for the numerator and denominator of each number. Of course the basic numerical operations now become costly, and a method was devised to reduce the cost of rational arithmetic predicates through the use of *floating point filters* ([Chapter 41](#)) which turn out to be very effective in practice, especially when the input is nondegenerate.

So far filtering has been applied to predicates but not to constructions. Notice that, if one has to produce the *exact* coordinates of an intersection point in an arrangement, there are no shortcuts and exact arithmetic needs to be used.

Matters are more complicated when the objects are not linear, namely when we deal with higher-degree curves and surfaces. First, there is the issue of representation. Consider the following simplest planar arrangement of the line $y = x$ and the circle $x^2 + y^2 = 1$ (both described by equations with integer coefficients). The upper vertex (intersection point) v_1 has coordinates $(\sqrt{2}/2, \sqrt{2}/2)$. This means that we cannot have a simple numerical representation of the vertices of the arrangement. An elegant solution to this problem is provided by special number types (so-called *algebraic number types*; notice though that only a subset of the algebraic numbers is currently supported). The approach is transparent to the user who just has to substitute the standard machine type (e.g., double) for the corresponding novel number type (which is a C++ class). Two software libraries support such number types (called *real* in both): LEDA [MN00] ([Chapter 65](#)) and Core [KLPY99] ([Chapter 41](#)). The ideas behind the solution proposed by both are similar and rely on separation bounds. In terms of arrangements the power that these number types provide is that we can determine the exact topology of the arrangement in all cases including degenerate cases. For example, if another circle passes through the point v_1 we can definitively determine this fact (which in general we cannot with standard machine arithmetic like the C++ double).

While exact computing may seem to be the solution to all problems, the sit-

uation is far from being satisfactory for several reasons: (i) The existing number types considerably slow down the computation compared with standard machine arithmetic. (ii) It is difficult to implement the full fledged number types required for arrangements of curves and surfaces. The state-of-the-art libraries offer the necessary types for arrangements of circles but not even for arrangements of conic arcs. Recently, an alternative approach has been taken to enable the exact construction of arrangements of conic arcs. It is based on using the GCD of the defining polynomials of arrangement vertices [BEH⁺02], [Wei02]. (iii) It still leaves open the question of handling degeneracies (see PERTURBATION below).

The high cost of exact predicates has led researchers to look for alternative algorithmic solutions (for problems where good solutions, in the standard measures of computational geometry, have been known), solutions that use less costly predicates; see, e.g., [BP00].

ROUNDING

In rounding we transform an arbitrary precision arrangement into fixed precision representation. The most intensively studied case is that of planar arrangements of segments. A solution proposed independently by Hobby [Hob99] and by Greene (improving on an earlier method in [GY86]), snaps vertices of the arrangement to centers of pixels in a prespecified grid. The method preserves several topological properties of the original arrangement and indeed expresses the vertices of the arrangement with limited precision numbers (say bounded bit-length integers). A dynamic algorithm is described in [GM98], and an improved algorithm for the case where there are many intersections within a pixel is given in [GGHT97]. Snap rounding has several drawbacks though: a line is substituted by a polyline possibly with many links (a “shortest-path” rounding scheme is proposed in [Mil00] that sometimes introduces fewer links than snap rounding), and a vertex of the arrangement can become very close to a non-incident edge (the latter problem has been overcome in an alternative scheme *iterated snap rounding* which guarantees a large separation between such features of the arrangement but pays in the quality of approximation [HP02]). Furthermore, a pair of input segments may intersect an arbitrarily large number of times in the rounded arrangements. Finally, the 3D version seems to produce a huge number of extra features [For99]: a polyhedral subdivision of complexity n turns into a snapped subdivision of complexity $O(n^4)$.

Effective and consistent rounding of arrangements remains an important and largely open problem. The importance of rounding arrangements stems not only from its being a means to overcome robustness issues, but, not less significantly, from being a way to express the arrangement numerically with reasonable bit-size numbers. Even if highly efficient exact number types are developed, there will still remain the question of numerical representation of the output.

APPROXIMATE ARITHMETIC IN PREDICATE EVALUATION

The behavior of fundamental algorithms for computing line arrangements (both sweep line and incremental) while using limited precision arithmetic is studied in [FM91]. It is shown that the two algorithms can be implemented such that for n lines the maximum error of the coordinates of vertices is $O(n\epsilon)$ where ϵ is the relative

error of the approximate arithmetic used (e.g., floating point). An algorithm for constructing curve arrangements with rounded arithmetic is presented in [Mil89].

PERTURBATION

An arrangement of lines is considered degenerate if it is not simple (Section 24.1). A degeneracy occurs for example when three lines meet at a common point. Intuitively this is a degeneracy since moving the lines slightly will result in a topologically different arrangement. Degeneracies in arrangements pose difficulties for two reasons. First and foremost they incredibly complicate programming (similar reasons led to the general position assumption—developing a theoretical algorithm that handles all possible degenerate cases is also a technically cumbersome and error-prone process). Although it has been proposed that handling degeneracies could be the solution in practice to relax the general position assumption [BMS94], in three and higher dimensions handling all degeneracies in arrangements seems an extremely difficult task. The second difficulty posed by degeneracies is that the numerical computation at or near degeneracies typically requires higher precision and will for example cause floating point filters to fail and resort to exact computing resulting in longer running time.

To overcome the first difficulty, symbolic perturbation schemes have been proposed. They enable a consistent perturbation of the input objects so that all degeneracies are removed. These schemes modify the objects only symbolically and a limiting process is used to define the perturbed objects (corresponding to infinitesimal perturbations) such that all predicates will have non-zero results. They require the usage of exact arithmetic, and a postprocessing stage to determine the structure of the output. For a unifying view of these schemes and a discussion of their properties, see [Sei98].

An alternative approach is to *actually* perturb the objects from their original placement. One would like to perturb the input objects as little as possible so that precision problems are resolved. This approach is viable in situations where the exact placement of the input can be compromised, as is the case in many engineering and scientific applications where the input is inexact due to measurement or modeling errors. An efficient such scheme for arrangements of spheres that model molecules is described in [HS98]; it has been adapted and extended to arrangements of polyhedral surfaces in [Raa99]. It is referred to as *controlled* perturbation since it guarantees that the final arrangement is degeneracy free (and predicates can be safely computed with limited precision arithmetic), to distinguish from heuristic perturbation methods. An in-depth study of the parameters that govern the scheme in the case of planar arrangements of circles is given in [HL03].

OPEN PROBLEM

Devise efficient and consistent rounding schemes for arrangements of curves in the plane and for arrangements in three and higher dimensions.

24.10 SOFTWARE

In spite of the numerous applications of arrangements, robust software for computing and manipulating arrangements is scarce. The reason for this is the difficulties outlined in the previous section. The situation is starting to change, with the increased understanding of the underlying difficulties, the research on overcoming these difficulties that has intensified during the last several years ([Chapter 41](#)), and the appearance of infrastructure for developing such software in the form of computational geometry libraries that emphasize robustness ([Chapter 65](#)).

24.10.1 2D ARRANGEMENTS

LEDA enables the construction of arrangements of segments via a sweep line algorithm. The resulting subdivision is represented as a LEDA graph. Point location based on persistent search trees is supported. The construction is robust through the use of arbitrary precision rationals.

A stand-alone package by Goldwasser supports arrangements of segments (or polygons), and the closely related arrangements of arcs of great circles on a sphere [Gol95]. Although care has been taken to handle degenerate polygons, the software uses standard floating point arithmetic and not exact number types.

Keyser et al. [KCMK99] describe a library for exact manipulation of algebraic curves, one application of which is computing the arrangement induced by such curves. Their method does not however handle degeneracies.

Arrangements of segments as well as of more general types of curves are supported by CGAL as we describe next.

2D ARRANGEMENTS IN CGAL

The most generic arrangement package at the time of the writing is the CGAL 2D arrangements package. The genericity is obtained through the separation of the combinatorial part of the algorithms and the numerical part [FHH⁺00]. (The overall design follows [Ket99].) The combinatorial algorithms are coded assuming that a small set of numerical/geometric operations (predicates and constructions) is supplied by the user for the desired type of curves. These operations are packed in a traits class ([Chapter 65](#)) that is passed as a parameter to the algorithms. The algorithms include the dynamic construction of the arrangement, represented as a doubly-connected-edge-list (DCEL), allowing for insertion and deletion of curves. Alternatively one can construct the arrangement using a sweep line algorithm. Then three algorithms for point location are supported. All algorithms handle arbitrary input, namely they do not assume general position. Several traits classes are supplied with the package for: line segments, circular arcs, canonical parabolas, polylines, and recently a unifying class for conic arcs [Wei02]. The CGAL arrangement package has been used to implement motion planning algorithms [AFH02], [HH02], a rounding scheme [HP02] and more.

An alternative algorithm for constructing arrangements of conic arcs (a static version using a sweep-line algorithm) was developed by Berberich et al. [BEH⁺02]. The more involved case of cubics is treated by Eigenwillig et al. [ESW02].

24.10.2 3D ARRANGEMENTS

Software to construct arrangements of triangles in 3-space exactly, assuming general position, is described in [SH02]. The implementation uses a space sweep algorithm and exact rational arithmetic. The arrangement is represented by its vertical decomposition or a sparser variant called the *partial vertical decomposition*. Several projects are underway whose goal is the construction of arrangements of algebraic surfaces in 3-space.

24.11 SOURCES AND RELATED MATERIAL

FURTHER READING

The study of arrangements through the early 1970s is covered by Grünbaum in [Grü67, Chapter 18], [Grü71], and [Grü72]. See also the monograph by Zaslavsky [Zas75].

In this chapter we have concentrated on more recent results. Details of many of these results can be found in the following books. The book by Edelsbrunner [Ede87] takes the view of “arrangements of hyperplanes” as a unifying theme for a large part of discrete and computational geometry until 1987. Sharir and Agarwal’s book [SA95] is an extensive report on results for arrangements of curves and surfaces. See also the more recent survey [AS00a].

Chapters dedicated to arrangements of hyperplanes in books: Mulmuley emphasizes randomized algorithms [Mul93], O’Rourke discusses basic combinatorics, relations to other structures and applications [O’R98], and Pach and Agarwal [PA95] discuss problems involving arrangements in discrete geometry. Boissonnat and Yvinec [BY98] discuss, in addition to arrangements of hyperplanes, arrangements of segments and of triangles.

Arrangements of hyperplanes and of surfaces are also the topics of chapters in the recently published book by Matoušek [Mat02].

RELATED CHAPTERS

- [Chapter 1: Finite point configurations](#)
- [Chapter 5: Pseudoline arrangements](#)
- [Chapter 6: Oriented matroids](#)
- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 22: Convex hull computations](#)
- [Chapter 23: Voronoi diagrams and Delaunay triangulations](#)
- [Chapter 33: Computational real algebraic geometry](#)
- [Chapter 34: Point location](#)
- [Chapter 36: Range searching](#)
- [Chapter 37: Ray shooting and lines in space](#)
- [Chapter 40: Randomization and derandomization](#)
- [Chapter 41: Robust geometric computation](#)
- [Chapter 42: Parallel algorithms in geometry](#)
- [Chapter 47: Algorithmic motion planning](#)
- [Chapter 48: Robotics](#)
- [Chapter 65: Two computational geometry libraries: LEDA and CGAL](#)

REFERENCES

- [AA92] P.K. Agarwal and B. Aronov. Counting facets and incidences. *Discrete Comput. Geom.*, 7:359–369, 1992.
- [AAC98] P.K. Agarwal, B. Aronov, T.M. Chan, and M. Sharir. On levels in arrangements of lines, segments, planes, and triangles. *Discrete Comput. Geom.*, 19:315–331, 1998.
- [AAS03] P.K. Agarwal, B. Aronov, and M. Sharir. On the complexity of many faces in arrangements of pseudo-segments and of circles. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 1–24. Springer-Verlag, Berlin, 2003.
- [AAS97] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. *SIAM J. Comput.*, 26:1714–1732, 1997.
- [ABE91] B. Aronov, M. Bern, and D. Eppstein. Arrangements of polytopes, 1991. Manuscript.
- [AdB⁺02] H.-K. Ahn, M. de Berg, P. Bose, S.-W. Cheng, D. Halperin, J. Matoušek, and O. Schwarzkopf. Separating an object from its cast. *Comput. Aided Design*, 34:547–559, 2002.
- [AdB⁺98] P.K. Agarwal, M. de Berg, J. Matoušek, and O. Schwarzkopf. Constructing levels in arrangements and higher order Voronoi diagrams. *SIAM J. Comput.*, 27:654–667, 1998.
- [AEGS92] B. Aronov, H. Edelsbrunner, L.J. Guibas, and M. Sharir. The number of edges of many faces in a line segment arrangement. *Combinatorica*, 12:261–274, 1992.
- [AEHS01] B. Aronov, A. Efrat, D. Halperin, and M. Sharir. On the number of regular vertices of the union of Jordan regions. *Discrete Comput. Geom.*, 25:203–220, 2001.
- [AES99] P.K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29:912–953, 1999.
- [AFH02] P.K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of Minkowski sums. *Comput. Geom. Theory Appl.*, 21:39–61, 2002. Special Issue, selected papers from the European Workshop Computational Geometry, Eilat, 2000.
- [AFK⁺92] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Approximate motion planning and the complexity of the boundary of the union of simple geometric figures. *Algorithmica*, 8:391–406, 1992.
- [AG86] N. Alon and E. Győri. The number of small semispaces of a finite set of points in the plane. *J. Combin. Theory Ser. A*, 41:154–157, 1986.
- [AGR00] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Computing the arrangement of curve segments: divide-and-conquer algorithms via sampling. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 705–706, 2000.
- [AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.
- [AM95] P.K. Agarwal and J. Matoušek. Dynamic half-space range reporting and its applications. *Algorithmica*, 13:325–345, 1995.
- [AMS94] B. Aronov, J. Matoušek, and M. Sharir. On the sum of squares of cell complexities in hyperplane arrangements. *J. Combin. Theory Ser. A*, 65:311–321, 1994.
- [AMS98] P.K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. *SIAM J. Comput.*, 27:491–505, 1998.

- [APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9:177–186, 1993.
- [AS90] B. Aronov and M. Sharir. Triangles in space or building (and analyzing) castles in the air. *Combinatorica*, 10:137–173, 1990.
- [AS94] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12:119–150, 1994.
- [AS98] P.K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30:412–458, 1998.
- [AS00a] P.K. Agarwal and M. Sharir. Arrangements and their applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier North-Holland, Amsterdam, 2000.
- [AS00b] P.K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: The union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24:645–685, 2000.
- [ASS96] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.
- [AST97] B. Aronov, M. Sharir, and B. Tagansky. The union of convex polyhedra in three dimensions. *SIAM J. Comput.*, 26:1670–1688, 1997.
- [Bal95] I.J. Balaban. An optimal algorithm for finding segment intersections. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 211–219, 1995.
- [Bas98] S. Basu. On the combinatorial and topological complexity of a single cell. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 606–616, 1998.
- [BD90] K.W. Bowyer and C.R. Dyer. Aspect graphs: An introduction and survey of recent results. *Internat. J. Imaging Systems Technology*, 2:315–328, 1990.
- [BD96] J.-D. Boissonnat and K. Dobrindt. On-line construction of the upper envelope of triangles and surface patches in three dimensions. *Comput. Geom. Theory Appl.*, 5:303–320, 1996.
- [BDH99] K.-F. Böhringer, B.R. Donald, and D. Halperin. On the area bisectors of a polygon. *Discrete Comput. Geom.*, 22:269–285, 1999.
- [BEH⁺02] E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, K. Mehlhorn, and E. Schömer. A computational basis for conic arcs and boolean operations on conic polygons. In *Proc. 10th European Sympos. Algorithms*, volume 2461 of *Lecture Notes in Comput. Sci.*, pages 174–186. Springer-Verlag, Rome, 2002.
- [BJ02] G. Stølting Brodal and R. Jacob. Dynamic planar convex hull. In *Proc. 43rd Annu. Sympos. Found. Comput. Sci.*, pages 617–626, 2002.
- [BK03] P. Brass and C. Knauer. On counting point-hyperplane incidences. *Comput. Geom. Theory Appl.*, 25:13–20, 2003
- [BLW⁺93] A. Björner, M. Las Vergnas, N. White, B. Sturmfels, and G. Ziegler. ERROR: Mis-match parsing authors: A. Björner, M. Las Vergnas, N. White, B. Sturmfels, and G. Ziegler. *Oriented Matroids*, volume 46 of *Encyclopedia of Mathematics*. Cambridge University Press, 1993.
- [BMS94] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994.
- [BP00] J.-D. Boissonnat and F.P. Preparata. Robust plane sweep for intersecting segments. *SIAM J. Comput.*, 29:1401–1421, 2000.
- [BPR00] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. *J. Amer. Math. Soc.*, 13:55–82, 2000.

- [Bri93] E. Brisson. Representing geometric structures in d dimensions: Topology and order. *Discrete Comput. Geom.*, 9:387–426, 1993.
- [BSTY98] J.-D. Boissonnat, M. Sharir, B. Tagansky, and M. Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. *Discrete Comput. Geom.*, 19:473–484, 1998.
- [BY98] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, 1998. Translated by H. Brönnimann.
- [Can93] J.F. Canny. Computing roadmaps in general semialgebraic sets. *Comput. J.*, 36:504–514, 1993.
- [CEG⁺90] K.L. Clarkson, H. Edelsbrunner, L.J. Guibas, M. Sharir, and E. Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete Comput. Geom.*, 5:99–160, 1990.
- [CEGS91] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. A singly-exponential stratification scheme for real semi-algebraic varieties and its applications. *Theoret. Comput. Sci.*, 84:77–105, 1991. An improved bound appears in *Proc. 16th ICALP 1989*, Stresa, *Lecture Notes Comput. Sci.* 372, Springer-Verlag, Berlin.
- [Cha96] T.M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete Comput. Geom.*, 16:369–387, 1996.
- [Cha00] T.M. Chan. Random sampling, halfspace range reporting, and construction of $(\leq k)$ -levels in three dimensions. *SIAM J. Comput.*, 30:561–575, 2000.
- [Cha01] T.M. Chan. Dynamic planar convex hull operations in near-logarithmic amortized time. *J. Assoc. Comput. Mach.*, 48:1–12, 2001.
- [Cha03] T.M. Chan. On levels in arrangements of curves. *Discrete Comput. Geom.*, 3:375–393, 2003.
- [Cla88] K.L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17:830–847, 1988.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [dB93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*, volume 703 of *Lecture Notes Comput. Sci.* Springer-Verlag, Berlin, 1993.
- [dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.
- [dBGH96] M. de Berg, L.J. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. *Discrete Comput. Geom.*, 15:35–61, 1996.
- [dBH⁺97] M. de Berg, D. Halperin, M.H. Overmars, and M. van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *Internat. J. Comput. Geom. Appl.*, 7:175–195, 1997.
- [dBvK⁺00] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*, 2nd edition. Springer-Verlag, Berlin, 2000.
- [Dey98] T.K. Dey. Improved bounds on planar k -sets and related problems. *Discrete Comput. Geom.*, 19:373–382, 1998.
- [DL89] D.P. Dobkin and M.J. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 4:3–32, 1989.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Heidelberg, 1987.

- [Ede89] H. Edelsbrunner. The upper envelope of piecewise linear functions: Tight complexity bounds in higher dimensions. *Discrete Comput. Geom.*, 4:337–343, 1989.
- [EG89] H. Edelsbrunner and L.J. Guibas. Topologically sweeping an arrangement. *J. Comput. Syst. Sci.*, 38:165–194, 1989. Corrigendum in 42:249–251, 1991.
- [EGH⁺89] H. Edelsbrunner, L.J. Guibas, J. Hershberger, J. Pach, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink. Arrangements of Jordan arcs with three intersections per pair. *Discrete Comput. Geom.*, 4:523–539, 1989.
- [EGP⁺92] H. Edelsbrunner, L.J. Guibas, J. Pach, R. Pollack, R. Seidel, and M. Sharir. Arrangements of curves in the plane: Topology, combinatorics, and algorithms. *Theoret. Comput. Sci.*, 92:319–336, 1992.
- [EGS89] H. Edelsbrunner, L.J. Guibas, and M. Sharir. The upper envelope of piecewise linear functions: algorithms and applications. *Discrete Comput. Geom.*, 4:311–336, 1989.
- [EGS90] H. Edelsbrunner, L.J. Guibas, and M. Sharir. The complexity of many cells in arrangements of planes and related problems. *Discrete Comput. Geom.*, 5:197–216, 1990.
- [Eri96] J. Erickson. New lower bounds for Hopcroft’s problem. *Discrete Comput. Geom.*, 16:389–418, 1996.
- [ERvK96] H. Everett, J.-M. Robert and M. van Kreveld. An optimal algorithm for the ($\leq k$)-levels, with applications to separation and transversal problems. *Internat. J. Comput. Geom. Appl.*, 6:247–261, 1996.
- [ES00] A. Efrat and M. Sharir. On the complexity of the union of fat objects in the plane. *Discrete Comput. Geom.*, 23:171–189, 2000.
- [ESS93] H. Edelsbrunner, R. Seidel, and M. Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22:418–429, 1993.
- [ESW02] A. Eigenwillig, E. Schömer, and N. Wolpert. Sweeping arrangements of cubic segments exactly and efficiently. Tech. Rep. ECG-TR-182202-01, INRIA, 2002.
- [EW86] H. Edelsbrunner and E. Welzl. Constructing belts in two-dimensional arrangements with applications. *SIAM J. Comput.*, 15:271–284, 1986.
- [FHH⁺00] E. Flato, D. Halperin, I. Hanniel, O. Nechushtan, and E. Ezra. The design and implementation of planar maps in CGAL. *ACM J. Experimental Algorithms*, 5, 2000. Selected papers from the Workshop on Algorithm Engineering (WAE).
- [FM91] S.J. Fortune and V.J. Milenkovic. Numerical stability of algorithms for line arrangements. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 334–341, 1991.
- [For99] S.J. Fortune. Vertex-rounding a three-dimensional polyhedral subdivision. *Discrete Comput. Geom.*, 22:593–618, 1999.
- [GGHT97] M.T. Goodrich, L.J. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 284–293, 1997.
- [GHH⁺98] L.J. Guibas, D. Halperin, H. Hirukawa, J.-C. Latombe, and R.H. Wilson. Polyhedral assembly partitioning using maximally covered cells in arrangements of convex polytopes. *Internat. J. Comput. Geom. Appl.*, 8:179–200, 1998.
- [GHS01] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 2001.
- [GM98] L.J. Guibas and D. Marimont. Rounding arrangements dynamically. *Internat. J. Comput. Geom. Appl.*, 8:157–176, 1998.

- [GO95] A. Gajentaan and M.H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, 5:165–185, 1995.
- [Gol95] M. Goldwasser. An implementation for maintaining arrangements of polygons. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C32–C33, 1995.
- [Goo93] M.T. Goodrich. Constructing arrangements optimally in parallel. *Discrete Comput. Geom.*, 9:371–385, 1993.
- [Grü67] B. Grünbaum. *Convex Polytopes*. John Wiley & Sons, New York, 1967.
- [Grü71] B. Grünbaum. Arrangements of hyperplanes. *Congr. Numer.*, 3:41–106, 1971.
- [Grü72] B. Grünbaum. *Arrangements and Spreads*. *Regional Conf. Ser. Math.*, Amer. Math. Soc., Providence, 1972.
- [GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.
- [GSS89] L.J. Guibas, M. Sharir, and S. Sifrony. On the general motion planning problem with two degrees of freedom. *Discrete Comput. Geom.*, 4:491–521, 1989.
- [GY86] D.H. Greene and F.F. Yao. Finite-resolution computational geometry. In *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 143–152, 1986.
- [Hal92] D. Halperin. *Algorithmic Motion Planning via Arrangements of Curves and of Surfaces*. Ph.D. thesis, Comput. Sci. Dept., Tel-Aviv Univ., 1992.
- [Hal94] D. Halperin. On the complexity of a single cell in certain arrangements of surfaces related to motion planning. *Discrete Comput. Geom.*, 11:1–34, 1994.
- [Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.
- [HH02] S. Hirsch and D. Halperin. Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane. In *Proc. 5th Workshop Algorithmic Found. Robot. (WAFR)*, Nice, 2002.
- [HJW90] T. Hagerup, H. Jung, and E. Welzl. Efficient parallel computation of arrangements of hyperplanes in d dimensions. In *Proc. 2nd Annu. ACM Sympos. Parallel Algorithms Architect.*, pages 290–297, 1990.
- [HL03] D. Halperin and E. Leiserowitz. Controlled perturbation for arrangements of circles. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 2003.
- [HLW00] D. Halperin, J.-C. Latombe, and R.H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26:577–601, 2000.
- [HO98] D. Halperin and M.H. Overmars. Spheres, molecules, and hidden surface removal. *Comput. Geom. Theory Appl.*, 11:83–102, 1998.
- [Hob99] J.D. Hobby. Practical segment intersection with finite precision output. *Comput. Geom. Theory Appl.*, 13:199–214, 1999.
- [HP00] S. Har-Peled. Constructing planar cuttings in theory and practice. *SIAM J. Comput.*, 29:2016–2039, 2000.
- [HP02] D. Halperin and E. Packer. Iterated snap rounding. *Comput. Geom. Theory Appl.*, 23:209–225, 2002.
- [HS94] D. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete Comput. Geom.*, 12:313–326, 1994.
- [HS95a] D. Halperin and M. Sharir. Almost tight upper bounds for the single cell and zone problems in three dimensions. *Discrete Comput. Geom.*, 14:385–410, 1995.

- [HS95b] D. Halperin and M. Sharir. Arrangements and their applications in robotics: Recent developments. In K. Goldbergs, D. Halperin, J.-C. Latombe, and R.H. Wilson, editors, *Algorithmic Found. Robot.*, A.K. Peters, Boston, 1995.
- [HS96] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.
- [HS98] D. Halperin and C.R. Shelton. A perturbation scheme for spherical arrangements with application to molecular modeling. *Comput. Geom. Theory Appl.*, 10:273–287, 1998.
- [KCMK99] J. Keyser, T. Culver, D. Manocha, and S. Krishnan. MAPC: A library for efficient and exact manipulation of algebraic points and curves. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1999.
- [Ket99] L. Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *Comput. Geom. Theory Appl.*, 13:65–90, 1999.
- [KLPS86] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.
- [KLPY99] V. Karamcheti, C. Li, I. Pechtchanski, and C.K. Yap. *The CORE Library Project*, 1.2 edition, 1999. <http://www.cs.nyu.edu/exact/core/>.
- [Kol01a] V. Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. In *Proc. 42nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 56–65, 2001.
- [Kol01b] V. Koltun. Sharp bounds for vertical decomposition of linear arrangements in four dimensions. In *Proc. 7th Workshop Algorithms Data Struct.*, pages 99–110, 2001.
- [KS02] V. Koltun and M. Sharir. The partition technique for overlays of envelopes. In *Proc. 43rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 637–646, 2002.
- [Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.
- [Mat02] J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, 2002.
- [McM70] P. McMullen. The maximal number of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.
- [Mil89] V.J. Milenkovic. Calculating approximate curve arrangements using rounded arithmetic. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 197–207, 1989.
- [Mil00] V.J. Milenkovic. Shortest path geometric rounding. *Algorithmica*, 27:57–86, 2000.
- [MMP⁺91] J. Matoušek, N. Miller, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–58, 1991.
- [MN00] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 2000.
- [MPS⁺94] J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. *SIAM J. Comput.*, 23:154–169, 1994.
- [Mul93] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1993.
- [NPP⁺02] E. Nevo, J. Pach, R. Pinchasi, M. Sharir, and S. Smorodinsky. Lenses in arrangements of pseudo-circles and their applications. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 123–132, 2002.
- [O'R98] J. O'Rourke. *Computational Geometry in C*, 2nd edition. Cambridge University Press, 1998.

- [PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. John Wiley & Sons, New York, 1995.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [PS99] J. Pach and M. Sharir. On the boundary of the union of planar convex sets. *Discrete Comput. Geom.*, 21:321–328, 1999.
- [PSS03] J. Pach, I. Safruti, and M. Sharir. The union of congruent cubes in three dimensions. *Discrete Comput. Geom.*, 30:133–160, 2003.
- [Raa99] S. Raab. Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1999.
- [SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [SAL93] A. Schweikard, J.E. Adler, and J.-C. Latombe. Motion planning in stereotaxic radiosurgery. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 764–774, 1993.
- [Sei98] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998.
- [SH02] H. Shaul and D. Halperin. Improved construction of vertical decompositions of 3D arrangements. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 283–292, 2002.
- [Sha91] M. Sharir. On k -sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.
- [Sha94] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.
- [SS97] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a three-dimensional arrangement of surfaces and its applications. *Discrete Comput. Geom.*, 18:269–288, 1997.
- [SST01] M. Sharir, S. Smorodinsky, and G. Tardos. An improved bound for k -sets in three dimensions. *Discrete Comput. Geom.*, 26:195–204, 2001.
- [Szé97] L.A. Székely. Crossing numbers and hard Erdős problems in discrete geometry. *Combinatorics, Prob. Comput.*, 6:353–358, 1997.
- [Tag96] B. Tagansky. A new technique for analyzing substructures in arrangements of piecewise linear surfaces. *Discrete Comput. Geom.*, 16:455–479, 1996.
- [Tot01] G. Tóth. Point sets with many k -sets. *Discrete Comput. Geom.*, 26:187–194, 2001.
- [TT98] H. Tamaki and T. Tokuyama. How to cut pseudo-parabolas into segments. *Discrete Comput. Geom.*, 19:265–290, 1998.
- [vK98] M. van Kreveld. On fat partitioning, fat covering, and the union size of polygons. *Comput. Geom. Theory Appl.*, 9:197–210, 1998.
- [Wei02] R. Wein. High level filtering for arrangements of conic arcs. In *Proc. 10th European Sympos. Algorithms*, volume 2461 of *Lecture Notes Comput. Sci.*, pages 884–895. Springer-Verlag, Rome, 2002.
- [Zas75] T. Zaslavsky. *Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes*, volume 154 of *Memoirs Amer. Math. Soc.* Amer. Math. Soc., Providence, 1975.

25 TRIANGULATIONS AND MESH GENERATION

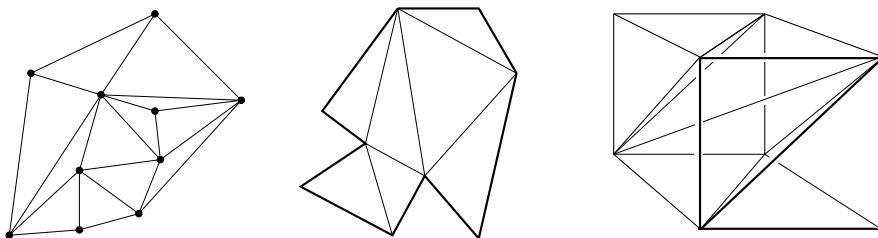
Marshall Bern

INTRODUCTION

A triangulation is a partition of a geometric domain, such as a point set, polygon, or polyhedron, into simplices that meet only at shared faces. (For point sets, the partition stops at the convex hull.) Triangulations are important for representing complicated geometry by piecewise simple geometry. The first four sections of this chapter discuss two-dimensional triangulations: Delaunay triangulation of point sets (Section 25.1); triangulations of polygons, including constrained Delaunay triangulation (Section 25.2); other optimal triangulations (Section 25.3); and mesh generation (Section 25.4). The next section treats the important practical case of polyhedra in \mathbb{R}^3 (Section 25.5). The last section discusses triangulations in arbitrary dimension \mathbb{R}^d (Section 25.6).

FIGURE 25.0.1

Triangulations of a point set, a simple polygon, and a polyhedron.



25.1 DELAUNAY TRIANGULATION

The Delaunay triangulation is the most famous and useful triangulation of a point set. [Chapter 23](#) discusses this construction in conjunction with the Voronoi diagram.

GLOSSARY

Empty circle: No input points in the interior.

Delaunay triangulation (DT): Triangles have empty circumcircles.

Completion: Four or more cocircular points must be further triangulated.

Edge flipping: Local improvement move, used to compute DT.

BASIC FACTS

Let $S = \{s_1, s_2, \dots, s_n\}$ be a set of points in the Euclidean plane \mathbb{R}^2 . The Delaunay triangulation (DT) is defined by the *empty circle condition*: a triangle $s_i s_j s_k$ appears in the DT if and only if its circumcircle neither encloses nor passes through any other points of S .

The DT always includes the convex hull of S . If no four points of S are cocircular, the Delaunay triangulation is indeed a triangulation of S . If four or more points are cocircular, there may be faces with more than three sides, which can be triangulated to *complete* the triangulation of S . The DT is the planar dual of the Voronoi diagram, meaning that an edge $s_i s_j$ appears in the DT if and only if the Voronoi cells of s_i and s_j share a boundary edge.

There is a connection between a Delaunay triangulation in \mathbb{R}^2 and a convex polytope in \mathbb{R}^3 . If we *lift* S onto the paraboloid with equation $z = x^2 + y^2$ by mapping $s_i = (x_i, y_i)$ to $(x_i, y_i, x_i^2 + y_i^2)$, then the DT turns out to be the projection of the lower convex hull of the lifted points. See [Figure 23.1.2](#).

ALGORITHMS

There are a number of practical planar DT algorithms [For95], including edge flipping, incremental construction, sweep-line, and divide-and-conquer. We describe only the edge flipping algorithm, even though its worst-case running time of $O(n^2)$ is not optimal, because it is most relevant to our subsequent discussion.

The edge flipping algorithm starts from any triangulation of S and then locally optimizes each edge. Let e be an internal (nonconvex-hull) edge and Q_e be the triangulated quadrilateral formed by the triangles sharing e . Q_e is *reversed* if the two angles without the diagonal sum to more than 180° , or equivalently, if each triangle circumcircle contains the opposite vertex. If Q_e is reversed, we “flip” it by exchanging e for the other diagonal.

```
Compute an initial triangulation of  $S$ 
Place all internal edges into a queue
while the queue is not empty do
    Remove the first edge  $e$ 
    if quadrilateral  $Q_e$  is reversed then flip it fi
    Add outside edges of  $Q_e$  to the queue od
```

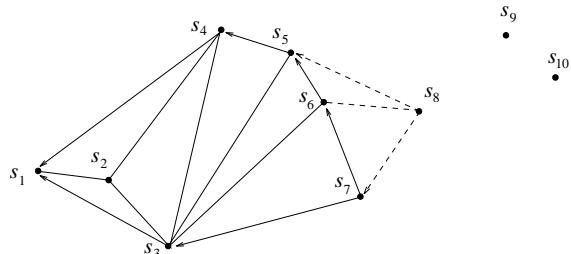


FIGURE 25.1.1

A generic step in computing the initial triangulation.

An initial triangulation can be computed by a sweep-line algorithm, as shown in [Figure 25.1.1](#). This algorithm adds the points of S by x -coordinate order. Upon each addition, the algorithm walks around the convex hull of the already-added points, adding edges until the slope reverses.

The following theorem guarantees the success of edge flipping: a triangulation in which no quadrilateral is reversed must be a completion of the DT. This theorem can be proved using the lifting map; a reversed quadrilateral lifts to a reflex edge, and a surface without reflex edges must be the lower convex hull.

OPTIMALITY PROPERTIES

Certain quality measures [BE95] are improved by flipping a reversed quadrilateral. For example, the minimum angle in a triangle of Q_e must increase. Hence, a triangulation that maximizes the minimum angle cannot have a reversed quadrilateral, implying that it is a completion of the DT. Some completion of the DT:

- minimizes the maximum radius of a circumcircle;
- maximizes the minimum angle (in fact, lexicographically maximizes the angles from smallest to largest);
- minimizes the maximum radius of an enclosing circle;
- maximizes the sum of inscribed circle radii;
- minimizes the “potential energy” of a piecewise-linear surface; and
- minimizes the surface area of a piecewise-linear surface for elevations scaled sufficiently small.

Two additional properties of the DT: Delaunay triangles are acyclically ordered by distance from any fixed reference point, and the distance along edges of the DT between any pair of vertices is at most a constant (at most 2.42) times the Euclidean distance between them.

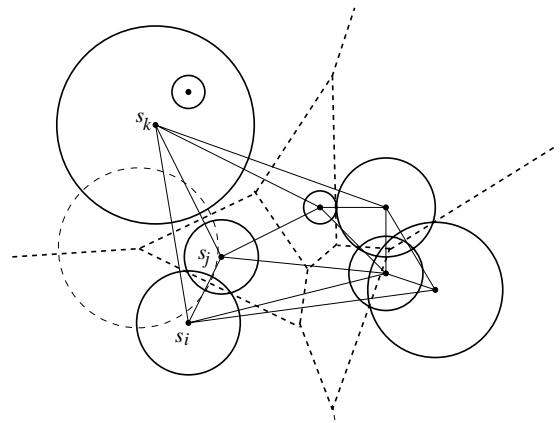


FIGURE 25.1.2

Power diagram and weighted Delaunay triangulation. The dashed circle is the orthogonal circle for triangle $s_i s_j s_k$.

REGULAR TRIANGULATIONS

Delaunay triangulations and Voronoi diagrams may be defined for various distance measures (Section 23.3); here we mention one generalization that retains most of the rich mathematical structure. Suppose each point $s_i = (x_i, y_i)$ in S has a weight w_i . The *regular triangulation* of S (sometimes called *weighted Delaunay triangulation*) is the projection of the lower convex hull of the points $(x_i, y_i, x_i^2 + y_i^2 - w_i)$. With a small (perhaps negative) weight, a site can drop out of the regular triangulation, so in general the regular triangulation is a graph on a subset of the sites S . In the special case that all weights are zero, the regular triangulation is exactly the DT. Because the w_i weights are arbitrary, regular triangulations in \mathbb{R}^2 are exactly the projections of lower convex hulls of polytopes in \mathbb{R}^3 . Not all triangulations are regular; see [Section 17.3](#) for a counterexample.

The planar dual of the regular triangulation is the *power diagram*, a Voronoi diagram in which the distance to s_i is the square of the Euclidean distance minus w_i . We can regard the sites in a power diagram as circles, with the radius of site i being $\sqrt{w_i}$. See [Figure 25.1.2](#). The analogue of the empty circle condition for regular triangulations is the *orthogonal circle condition*: a triangle $s_i s_j s_k$ appears in the triangulation if and only if the circle that crosses circles i, j and k at right angles penetrates no other site circle more deeply.

25.2 TRIANGULATIONS OF POLYGONS

We now discuss triangulations of more complicated inputs: polygons and planar straight-line graphs. We start with the problem of simply computing any triangulation and then progress to constrained Delaunay triangulation.

GLOSSARY

Simple polygon: Connected boundary without self-intersections.

Monotone polygon: Intersection with any vertical line is one segment.

Constrained Delaunay triangulation: Allows input edges as well as vertices.

Triangles have empty circumcircles, meaning no visible input vertices.

SIMPLE POLYGONS

Triangulating a simple polygon is both an interesting problem in its own right and an important preprocessing step in other computations. For example, the following problems are known to be solvable in linear time once the input polygon P is triangulated: computing link distances from a given source, finding a monotone path within P between two given points, and computing the portion of P illuminated by a given line segment,

How much time does it take to triangulate a simple polygon? For practical purposes, one should use either an $O(n \log n)$ deterministic algorithm (such as the one given below for the more general case of planar straight-line graphs) or a slightly

faster randomized algorithm (such as one with running time $O(n \log^* n)$ included in [Mul94]).

However, for theoretical purposes, achieving the ultimate running time was for several years an outstanding open problem. After a sequence of interim results, Chazelle [Cha91] devised a linear-time algorithm. Chazelle's algorithm, like previous algorithms, reduces the problem to that of computing the ***horizontal visibility map*** of P —the partition obtained by shooting horizontal rays left and right from each of the vertices. The “up-phase” of this algorithm recursively merges coarse visibility maps for halves of the polygon (polygonal chains); the “down-phase” refines the coarse map into the complete horizontal visibility map.

PLANAR STRAIGHT-LINE GRAPHS

Let G be a planar straight-line graph (PSLG). We describe an $O(n \log n)$ algorithm [PS85] that triangulates G in two stages, called regularization and triangulation. Regularization adds edges to G so that each vertex, except the first and last, has at least one edge extending to the left and one extending to the right. Conceptually, we sweep a vertical line ℓ from left to right across G while maintaining the list of intervals of ℓ between successive edges of G . For each interval I , we remember a vertex $v(I)$ visible to all points of I ; this vertex will be either an endpoint of one of the two edges bounding I or a vertex between these edges, lacking a right edge. When we hit a vertex u with no left edge, we add the edge $\{u, v(I)\}$, where I is the interval containing u , as shown in Figure 25.2.1(a). After the left-to-right sweep, we sweep from right to left, adding right edges to vertices lacking them.

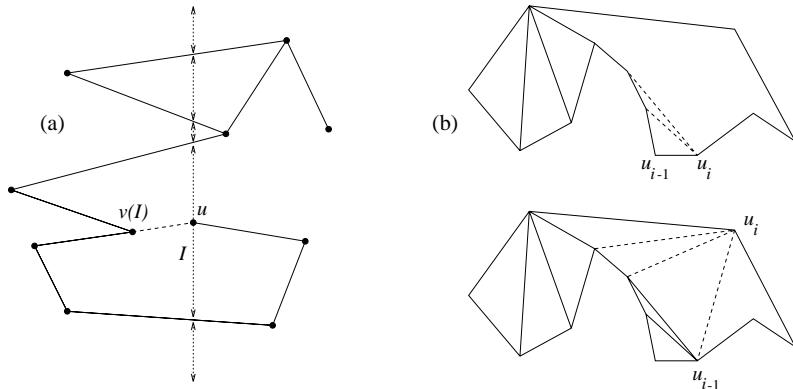
```

Start  $\ell$  at left with  $v(interval(u)) = (-\infty, 0)$ 
for each vertex  $u$  from left to right do
    if  $u$  has no left edges then add edge  $\{u, v(interval(u))\}$  fi
    Delete  $u$ 's left edges from interval list
    Insert  $u$ 's right edges with  $v()$  set to  $u$  od
Repeat the steps above for vertices from right to left

```

FIGURE 25.2.1

(a) Sweep-line algorithm for regularization. (b) Stack-based triangulation algorithm.



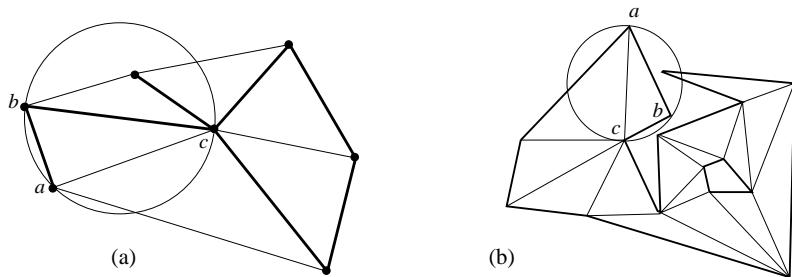
After the regularization stage, each bounded face of G is **monotone**, meaning that a vertical line intersects the face in at most one segment. We consider the vertices u_1, u_2, \dots, u_n of a face in left-to-right order, using a stack to store the not-yet-triangulated vertices (a reflex chain) to the left of the current vertex u_i . If u_i is adjacent to u_{i-1} , the topmost vertex on the stack, as shown in the upper picture of Figure 25.2.1(b), then we pop vertices off the stack and add diagonals from these vertices to u_i , until the vertices on the stack— u_i on top—again form a reflex chain. If u_i is instead adjacent to the leftmost vertex on the stack, as shown in the lower picture, then we can add a diagonal from each vertex on the stack, and clear the stack of all vertices except u_i and u_{i-1} .

CONSTRAINED DELAUNAY TRIANGULATION

Constrained Delaunay triangulation [LL86] provides a way to force the edges of a planar straight-line graph G into the DT. A point p is **visible** to point q if line segment pq does not intersect any edge or vertex in G , except maybe at its endpoints. A triangle abc with vertices from G appears in the **constrained Delaunay triangulation** (CDT) if its circumcircle neither contains nor passes through any other vertex of G visible to some point in abc . If G is a graph with vertices but not edges, then this definition generalizes ordinary, unconstrained Delaunay triangulation. If G is a polygon or polygon with holes, as in Figure 25.2.2(b), then the CDT retains only the triangles interior to G .

FIGURE 25.2.2

Constrained Delaunay triangulations of (a) a PSLG and (b) a polygon with a hole.



The edge flipping algorithm generalizes to the constrained case, with the modification that edges of G are never placed on the queue. There are also $O(n \log n)$ -time algorithms for the CDT, and even a randomized $O(n)$ algorithm for the case that G is just a simple polygon [KL93]. See [Section 64.2](#) for pointers to software for computing the constrained Delaunay triangulation.

25.3 OPTIMAL TRIANGULATIONS

We have already seen two types of optimal triangulations: the DT and the CDT. Some applications, however, demand triangulations with properties other than

those optimized by these two triangulations. Table 25.3.1 gives a summary of results; each result holds for arbitrary PSLGs, except the fourth, which applies only to polygons.

GLOSSARY

- Edge insertion:** Local improvement operation, more general than edge flipping.
- Local optimum:** A solution that cannot be improved by local moves.
- Greedy triangulation:** At each step, add the shortest valid edge.
- Steiner triangulation:** Extra—noninput—points are allowed.

TABLE 25.3.1 Optimal triangulation results.

PROPERTY	ALGORITHMS	TIME
Delaunay	Various algorithms [For95]	$O(n \log n)$
Minmax angle	Fast edge insertion [ETW92]	$O(n^2 \log n)$
Minmax slope terrain	Edge insertion [BEE ⁺ 93]	$O(n^3)$
Min total edge length	Approx'n algorithms [Epp94, LK96]	$O(n \log n)$
Minmax edge length	MST induces polygons [ET91]	$O(n^2)$
Greedy edge length	Dynamic Voronoi diagram [LL92]	$O(n^2)$

EDGE FLIPPING AND EDGE INSERTION

The edge flipping DT algorithm can be modified to compute many other optimal triangulations. For example, if we redefine “reversed” to mean a quadrilateral triangulated with the diagonal that forms the larger maximum angle, then edge flipping can be used to minimize the maximum angle. For minmax angle, however, edge flipping computes only a local optimum, not necessarily the true global optimum.

Although edge flipping seems to work well in practice [ETW92], its theoretical guarantees are very weak: the running time is not known to be polynomially bounded and the local optimum it finds may be greatly inferior to the true optimum.

A more general local improvement method, called *edge insertion* [BEE⁺93, ETW92] exactly solves certain minmax optimization problems, including minmax angle and minmax slope of a piecewise-linear interpolating surface.

Assume that the input is a planar straight-line graph G , and we are trying to minimize the maximum angle. Starting from some initial triangulation of G , edge insertion repeatedly adds a candidate edge e that subdivides the maximum angle. (In general, edge insertion always breaks up a worst triangle by adding an edge incident to its “worst vertex.”) The algorithm then removes the edges that are crossed by e , forming two polygonal holes alongside e . Holes are retriangulated by repeatedly removing *ears* (triangles with two sides on the boundary, as shown in Figure 25.3.1) with maximum angle smaller than the old worst angle $\angle cab$. If retriangulation succeeds, then the overall triangulation improves and edge bc is eliminated as a future candidate. If retriangulation fails, then the overall triangulation is returned to its state before the insertion of e , and e is eliminated as a future candidate. Each candidate insertion takes time $O(n)$, giving a total running time of $O(n^3)$.

```

Compute an initial triangulation with all  $\binom{n}{2}$  edge slots unmarked
while  $\exists$  an unmarked edge  $e$  cutting the worst vertex of worst triangle  $abc$  do
    Add  $e$  and remove all edges crossed by  $e$ 
    Try to retriangulate by removing ears better than  $abc$ 
    if retriangulation succeeds then
        mark  $bc$ 
    else mark  $e$  and undo  $e$ 's insertion fi od

```

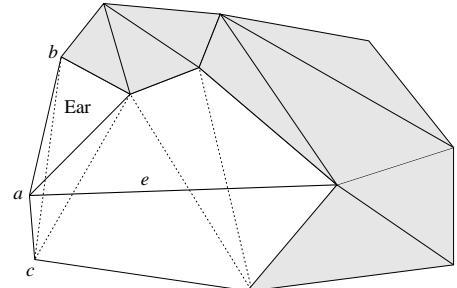


FIGURE 25.3.1

Edge insertion retriangulates holes by removing sufficiently good ears. (From [BE95], with permission.)

Edge insertion can compute the minmax “eccentricity” triangulation or the minmax slope surface [BEE⁺93] in time $O(n^3)$. By inserting candidate edges in a certain order, one can improve the running time to $O(n^2 \log n)$ for minmax angle [ETW92] and maxmin triangle height.

MINIMUM WEIGHT TRIANGULATION

Several natural optimization criteria can be defined using edge lengths [BE95]. The most famous such criterion—called **minimum weight triangulation**—asks for a triangulation of a planar point set minimizing the total edge length. No polynomial-time algorithm is known for this problem, nor is it known to be NP-complete. There is, however, a recently developed algorithm that is quite fast in practice. This algorithm [DKM97] uses a local criterion to find edges sure to be in the minimum weight triangulation. These edges break the convex hull of the point set into regions, such as simple polygons or polygons with one or two disconnected interior points, that can be triangulated optimally using dynamic programming.

The best approximation algorithm for minimum weight triangulation, by Lev-copoulos and Krznaric [LK96], gives a solution within a constant multiplicative factor of the optimal length. Eppstein gave a constant-factor approximation ratio for minimum weight Steiner triangulation, in which extra vertices are allowed.

A commonly used heuristic for minimum weight triangulation is **greedy triangulation**. This algorithm adds edges one at a time, each time choosing the shortest edge that is not already crossed. Greedy triangulation can be viewed as an optimal triangulation in its own right, because it lexicographically minimizes the sorted vector of edge lengths. For arbitrary planar point sets, the greedy triangulation can be computed in time $O(n^2)$ by dynamic maintenance of a bounded Voronoi diagram [LL92].

Another natural criterion asks for a triangulation minimizing the maximum

edge length. Edelsbrunner and Tan [ET91] showed that such a triangulation—like the DT—must contain the edges of the minimum spanning tree (MST). This geometric lemma gives the following polynomial-time algorithm: compute the MST and then triangulate the resulting simple polygons optimally using dynamic programming.

OPEN PROBLEMS

1. Explain the empirical success of edge flipping for non-Delaunay optimization criteria, both solution quality and running time.
 2. Settle the complexity of min weight triangulation—in P or NP-complete?
 3. Show that the min weight Steiner triangulation exists, that is, rule out the possibility that more and more Steiner points decrease the total edge length forever.
-

25.4 PLANAR MESH GENERATION

A *mesh* is a decomposition of a geometric domain into *elements*, usually triangles or quadrilaterals in \mathbb{R}^2 . Meshes are used to discretize continuous functions, especially solutions to partial differential equations. Practical mesh generation problems tend to be application-specific: one desires small elements where the function changes rapidly and larger elements elsewhere. However, certain goals apply fairly generally, and computational geometers have formulated problems incorporating these considerations. Table 25.4.1 summarizes these results, and below we discuss some of them in detail.

GLOSSARY

Steiner point: An extra vertex, not an input point.

Conforming mesh: Elements exactly fill out the input domain.

Quadtree: A recursive subdivision of the plane with squares.

NO SMALL ANGLES

Sharp angles can degrade appearance and accuracy, so most mesh generation methods attempt to avoid small angles. (There is an exception: properly aligned sharp triangles prove quite useful in simulations of viscous flow.)

Baker et al. [BGR88] gave a grid-based algorithm for triangulating a PSLG so that all *new* angles—a sharp angle in the input cannot be erased—measure at least 14° . Bern et al. [BEG94] used quadtrees instead of a uniform grid and proved the following efficiency guarantee: the number of triangles is $O(1)$ times the minimum number in any no-small-angle triangulation of the input. The number of triangles required depends not just on the number of input vertices n , but also on the

TABLE 25.4.1 Mesh generation results.

PROPERTY	INPUTS	ALGORITHMS	SIZE
No small angles	polygons	Quadtrees [BEG94], circles [Rup93]	$O(1) \cdot \text{Optimal}$
No small solid angles	polyhedra	Octrees [MV92]	$O(1) \cdot \text{Optimal}$
No small or obtuse	polygons	Grids [BGR88], quadtrees	$O(1) \cdot \text{Optimal}$
No obtuse angles	polygons	Disk packing [BMR94]	$O(n)$
No obtuse angles	some PSLGs	Grids [BE92]	$O(n^4)$
No large angles	PSLGs	Propagating horns [Mit93, Tan94]	$O(n^2)$
Conforming Delaunay	PSLGs	Blocking & propagation [ET93]	$O(n^3)$

geometry of the input. The simple example of a long skinny rectangle shows why the number of triangles depends upon the geometry. Ruppert [Rup93], building on work of Chew, devised a **Delaunay refinement** algorithm with the same guarantee. The main loop of Ruppert’s algorithm attempts to add the circumcenter of a too-sharp triangle. If the circumcenter “encroaches” upon a boundary edge, meaning that it falls within the edge’s diameter circle and is visible to that edge, then the algorithm subdivides the boundary edge instead of adding the triangle circumcenter. Edelsbrunner and Guoy [EG01] proposed a more selective—and empirically more efficient—form of Delaunay refinement called **sink insertion**; this method does not add the circumcenter of the too-sharp triangle, but rather follows a chain of triangles until reaching one that contains its own circumcenter.

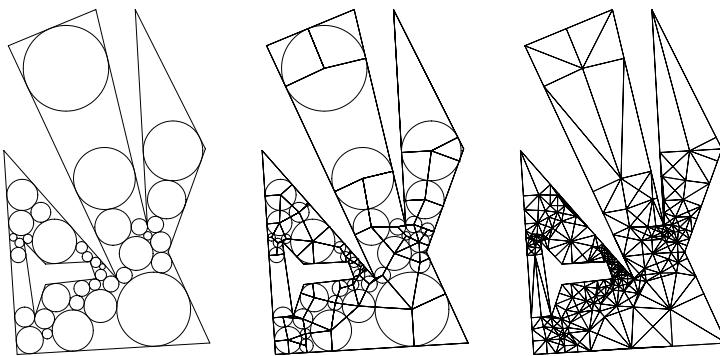
The efficiency guarantees for these Delaunay refinement algorithms follow from a stronger guarantee: at each point p of the domain the mesh triangle will be within a constant factor of the “local feature size,” which for polygons can be simply stated as the distance from p to the second-closest polygon vertex (see also [Chapter 30](#)). Miller et al. [MT⁺95] expanded Ruppert’s algorithm to a sort of paradigm, randomizable and parallelizable: pack the domain with a maximal set of non-overlapping disks with radii within a constant factor of the local feature size, and then compute the Delaunay triangulation of the disk centers. This approach is also related to “bubble meshing,” which simulates physical forces in order to place mesh vertices. Disk packing and placement, and in three dimensions ball placement, has proved to be a powerful and flexible approach to mesh generation, that can handle difficult practical issues such as multilevel meshes and solution adaptation, without sacrificing provable guarantees [Ber02].

NO LARGE ANGLES

A weaker condition than avoiding sharp angles is to avoid large angles (close to 180°). The strictest bound on large angles that does not also imply a bound on small angles is to ask for no obtuse angles, that is, all angles at most 90° . Surprisingly, it is possible to triangulate any polygon (possibly with holes) with only $O(n)$ nonobtuse triangles [BMR94]. [Figure 25.4.1](#) illustrates the algorithm: the domain is packed with nonoverlapping disks until each uncovered region has either 3 or 4 sides; radii to tangencies are added in order to split the domain into small polygons; and finally these polygons are triangulated with right triangles, without adding any new subdivision points (vertices embedded within edges).

FIGURE 25.4.1

Nonobtuse triangulation steps. (From [BMR94], [BE95], with permission.)



By relaxing the bound on the largest angle from 90° to something larger, researchers have obtained results for arbitrary PSLGs. Mitchell [Mit93] gave an algorithm that uses $O(n^2 \log n)$ triangles to guarantee that all angles measure less than $\frac{7}{8}\pi$. The algorithm traces a cone of possible angle-breaking edges, called a *horn*, from each vertex—including subdivision points—with a larger angle. Horns propagate around the PSLG until meeting an exterior edge or another horn. By adding some more horn-stopping “traps,” Tan [Tan94] improved the angle bound to $\frac{11}{15}\pi$ and the complexity bound to $O(n^2)$, matching a lower bound.

CONFORMING DELAUNAY TRIANGULATION

A convenient mesh generation approach adds extra vertices—*Steiner points*—to the input, until the Delaunay triangulation of the vertices “conforms” to the input, meaning that each input edge is a union of Delaunay edges.

There are a number of algorithms for this problem in the plane; all take the basic approach of covering the input edges by disks that do not enclose any input vertices. Edelsbrunner and Tan [ET93] gave an algorithm that uses $O(n^3)$ triangles, currently the only polynomial algorithm.

SURFACE MESHES

A topic that sits between two and three dimensions is surface meshes for 3D solids. Key problems include *surface reconstruction*, that is, fitting a triangulated surface to a set of sample points, *mesh simplification*, reducing the number of triangles while preserving essential topology and geometry, and *geometry compression*, encoding the geometry efficiently.

Recent papers on surface reconstruction [ABK98, ACDL00, ACK01] assume that the input points satisfy a *sampling condition*: at any location on a smooth surface the closest sample point is no farther away than some constant times the distance to the surface’s *medial axis*. Under this condition—which in some sense captures both surface curvature and thickness of the solid—the 3D Delaunay triangulation of the sample points contains a set of triangles conforming to the surface, and algorithms based on the shapes of Voronoi cells can pick out such a set. See

[Chapter 30](#) for further details.

For mesh simplification, surface curvature is more relevant than thickness of the solid, because the topology of the surface is already known. A smart simplification algorithm [GH97] repeatedly contracts an edge and repositions the coalesced vertex to the location in space that minimizes the sum of squared distances (“quadric error”) to all the planes supporting original faces incident to vertices that have gone into the coalesced vertex.

Geometry compression (cf. [Chapter 54](#)) can be either lossless or lossy. The key to lossless compression [TR98] is good prediction of vertex coordinates based on neighboring vertices. Lossy compression can simplify the mesh or otherwise change its connectivity for still more compact encoding. One very effective lossy method [GSS99] remeshes the surface into a *semiregular mesh*, in which all but the largest triangles are obtained by repeated subdivision of a triangle into four congruent copies of itself; the resulting hierarchical encoding is related to wavelet-based image compression.

OPEN PROBLEMS

1. Does every PSLG have a polynomial-size nonobtuse triangulation?
2. Does every PSLG have a conforming Delaunay triangulation of size $O(n^2)$?
3. What sampling condition is necessary and sufficient to reconstruct surfaces with corners and creases?

25.5 THREE-DIMENSIONAL POLYHEDRA

In this section we discuss the triangulation (or *tetrahedralization*) of 3D polyhedra. A polyhedron P is a flat-sided (connected) solid, usually assumed to satisfy the following nondegeneracy condition: around any point on the boundary of P , a sufficiently small ball contains one connected component of each of the interior and exterior of P . With this assumption, the numbers of vertices, edges, and faces (facets) of P are all linearly related.

GLOSSARY

Reflex edge: An edge with interior dihedral angle greater than 180° . (The dihedral angle between faces is measured on a plane normal to the shared edge.)

Convex polyhedron: A polyhedron without reflex edges.

Simple polyhedron: Topologically equivalent to a ball; edge skeleton forms a planar graph.

General polyhedron: May be topologically equivalent to a solid torus or higher-genus object, and may have more than one boundary component (i.e., cavities).

BAD EXAMPLES

Three dimensions is not as nice as two. Triangulations of the same input may contain different numbers of tetrahedra. For example, a triangulation of an n -vertex convex polyhedron may have as few as $n - 3$ or as many as $\binom{n-2}{2}$ tetrahedra. Below et al. [BLR00] recently proved that finding the minimum number of tetrahedra needed to triangulate (without Steiner points) a convex polyhedron is NP-complete. And when we move to nonconvex polyhedra, we get an even worse surprise: some inputs cannot even be triangulated without Steiner points.

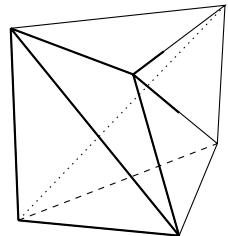


FIGURE 25.5.1

A twisted prism cannot be triangulated without Steiner points.

Schönhardt's polyhedron, shown in Figure 25.5.1, is the simplest example of a polyhedron that cannot be triangulated. Ruppert and Seidel [RS92] proved the NP-completeness of determining whether a polyhedron can be triangulated without Steiner points, and of testing whether k Steiner points suffice.

Chazelle [Cha84] gave an n -vertex polyhedron that requires $\Omega(n^2)$ Steiner points. This polyhedron is a box with thin wedges removed from the top and bottom faces (Figure 25.5.2). The tips of the wedges nearly meet at the hyperbolic surface $z = xy$ and divide this surface into $\Omega(n^2)$ small squares, no pair of which can lie in the same tetrahedron in a triangulation.

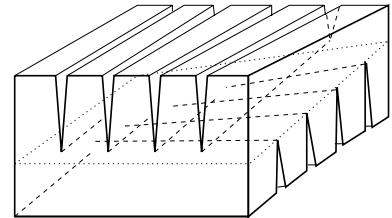


FIGURE 25.5.2

A polyhedron that requires $\Omega(n^2)$ tetrahedra. (From [BE95], with permission.)

GENERAL POLYHEDRA

Any polyhedron can be triangulated with $O(n^2)$ tetrahedra, matching the lower bound. One algorithm shoots vertical walls up and down from each edge of the polyhedron boundary; walls stop when they reach some other part of the boundary. The tops and bottoms of the resulting “cylinders” are then triangulated to produce $O(n^2)$ triangular prisms, which can each be triangulated with a single interior Steiner point. An improvement first plucks off “pointed vertices” with unhindered “caps.” Such a vertex, together with its incident faces, forms an empty convex cone.

The improved algorithm uses $O(n + r^2)$ tetrahedra, where r is the number of reflex edges on the original polyhedron [CP90].

An alternative algorithm [Cha84] divides the polyhedron into convex solids by incrementally bisecting each reflex angle with a plane that extends away from the reflex angle in all directions until it first contacts the polyhedron boundary. This algorithm produces at most $O(nr + r^{7/3})$ tetrahedra [HS92].

SPECIAL POLYHEDRA

Any strictly convex polyhedron can be triangulated with at most $2n - 7$ tetrahedra by “starring” from a vertex. The region between two convex polyhedra (the convex hull of the union, minus the polyhedra), with a total of n vertices, can be triangulated without any Steiner points. If Steiner points are allowed, $O(n)$ tetrahedra suffice. The union of three convex polyhedra can also be tetrahedralized without Steiner points. The region between a convex polyhedron and a terrain can be triangulated with $O(n \log n)$ tetrahedra, and in fact, some such regions require $\Omega(n \log n)$ tetrahedra [CS94].

THREE-DIMENSIONAL MESH GENERATION

Mesh generation for 3D solids is an important, largely open, practical problem. Current approaches include octrees (the generalization of quadtrees), advancing front, bubble meshing, and Delaunay refinement, but no one method gives satisfactory results for all applications. Some of the practical issues include types of elements (tetrahedral or cubical or perhaps a mix), shapes of elements (solid and dihedral angles bounded away from extremes), anisotropy (stretched elements for accurate discretization of laminar flows), and solution adaptation (refinement and derefinement in regions where it is needed).

On the theoretical side, Mitchell and Vavasis [MV92] gave an octree method that guarantees well-shaped tetrahedra (equivalently, no small solid angles) and efficiency within a constant factor of optimal, the generalization of [BEG94] to \mathbb{R}^3 . Miller et al. [MT⁺95] used maximal ball packing and Delaunay triangulation to guarantee well-shaped tetrahedra with the exception of *slivers*, the unique type of bad tetrahedron that can occur in a DT of a well-spaced point set. A sliver is a flat tetrahedron whose projection onto a plane that passes near all its vertices is fairly square; this is the only type of bad tetrahedron that has a small ratio of circumsphere radius to shortest edge. Luckily slivers are relatively fragile and can be removed (with weak but provable guarantees) by perturbing the point set [Ber02, Ede01]. One such perturbation, called *sliver exudation*, has the advantage that it does not actually move the points, but rather changes vertex weights in a weighted Delaunay triangulation; key to the success of sliver exudation is the result that a mild change in vertex weights dramatically changes the size of the orthogonal sphere [Ede01].

Constrained Delaunay triangulation does not extend to \mathbb{R}^3 , because not every polyhedron has a triangulation without Steiner points, and even “easy” polyhedra may not have triangulations that use only tetrahedra with empty circumspheres. Shewchuk [She98] devised the closest thing to a 3D constrained Delaunay triangulation. Call a segment of a polyhedron X *strongly Delaunay* if it has a circumsphere

that neither encloses nor passes through any other vertex of X . Call a simplex (line, triangle, tetrahedron) *constrained Delaunay* if it has a circumsphere that encloses no vertex of X visible to any point in the relative interior of the simplex. If each of X 's segments is strongly Delaunay, then X has a triangulation in which each simplex is constrained Delaunay. In the nondegenerate case of no five cospherical vertices, the constrained Delaunay triangulation of X is unique. Shewchuk has used this notion of constrained Delaunay triangulation in a Delaunay refinement mesh generator that generalizes Ruppert's 2D generator: input edges are subdivided until their diametral spheres are empty, input faces (assumed triangles) are subdivided until their equatorial spheres are empty, and finally badly shaped tetrahedra are fixed by adding their circumcenters. This generator eliminates all types of bad tetrahedra except slivers.

OPEN PROBLEMS

1. Can the region between k convex polytopes, with n vertices in total, be (Steiner) triangulated with $O(n + k^2)$ tetrahedra?
2. Give an *input-sensitive* tetrahedralization algorithm, for example, one that uses only $O(1)$ times the smallest number of tetrahedra.
3. Give a polynomial bound (or even a simple-to-state bound depending upon geometry) on the number of Steiner points needed to make all segments of a polyhedron strongly Delaunay.
4. (Üngör) Can a cube be triangulated such that all tetrahedra have only acute dihedral angles? The corresponding question in two dimensions—triangulate a square with acute triangles—is a well-known, and fairly easy, puzzle.
5. Give an algorithm for computing tetrahedralizations of point sets or polyhedra, such that each tetrahedron contains its own circumcenter. This condition guarantees a desirable matrix property for a finite-volume formulation of an elliptic partial differential equation [Ber02].

25.6 ARBITRARY DIMENSION

We now discuss triangulation algorithms for arbitrary dimension \mathbb{R}^d . In our big-O expressions, we consider the dimension d to be fixed.

GLOSSARY

Polytope: A bounded intersection of halfspaces in \mathbb{R}^d .

Face: A subpolytope such as a vertex, edge, or 2D face.

Simplex: The convex hull of $d + 1$ affinely independent points in \mathbb{R}^d .

Circumsphere: The sphere through the vertices of a simplex.

Flip: A local operation, sometimes called a geometric bistellar operation, that exchanges two different triangulations of $d + 2$ points in \mathbb{R}^d .

POINT SETS

Delaunay triangulation—and more generally regular triangulation—extends to \mathbb{R}^d . The DT contains a simplex if and only if its circumsphere neither encloses nor passes through any other input points. The lifting map generalizes as well, and can be used to show that the DT includes at most $O(n^{\lceil d/2 \rceil})$ simplices. For practical applications such as interpolation, surface reconstruction and mesh generation, however, the DT rarely attains its worst-case complexity. The DT of random points within a volume or on a convex surface in \mathbb{R}^3 has linear expected complexity, but on a nonconvex surface can have near-quadratic complexity [Eri03]. DT complexity can also be bounded by geometric parameters such as the ratio between longest and shortest pairwise distances [Eri03].

Due to the lifting relation, any convex hull algorithm can be used to compute DTs. Many of the two-dimensional algorithms mentioned above also generalize to \mathbb{R}^d ; however, the generalization of the edge flipping algorithm is not entirely straightforward. A flip in \mathbb{R}^d exchanges two triangulations of $d+2$ points in convex position. For example, 5 points in \mathbb{R}^3 can be triangulated by two tetrahedra sharing a face or by three tetrahedra sharing an edge. Flipping from an arbitrary triangulation in \mathbb{R}^3 can get stuck before reaching the DT [Joe89], but incrementally adding a point, splitting a simplex, and then flipping cannot. In fact, randomized incremental insertion is the most popular algorithm for computing DTs in \mathbb{R}^3 .

Most DT optimality properties do not generalize to higher dimensions. One exception: the DT minimizes the maximum radius of a simplex enclosing sphere. The *enclosing sphere* is the smallest sphere containing a simplex, either the circumsphere, or the circumsphere of some face.

Of interest in algebraic geometry as well as computational geometry is the *flip graph* or *triangulation space*, which has a vertex for each distinct triangulation and an edge for each flip. Using the lifting relation, we can view flipping as exchanging the lower and upper convex hulls of $d+2$ lifted points. For the flip graph we do not require the $d+2$ points to be in convex position, and thus we allow a flip that inserts a new vertex, for example, splitting a tetrahedron in \mathbb{R}^3 into four by inserting an interior vertex. The flip graph of regular triangulations has the structure of a high-dimensional polytope [BFS90, GKZ90], but the flip graph including nonregular triangulations is not well understood. Santos [San98] recently showed that for points in \mathbb{R}^5 the flip graph including nonregular triangulations may not be connected, and in \mathbb{R}^6 may even have an isolated vertex.

The following is known about Steiner triangulations of point sets in \mathbb{R}^d . It is always possible to add $O(n)$ Steiner points, so that the DT of the augmented point set has size only $O(n)$, and there is always a nonobtuse Steiner triangulation containing at most $O(n^{\lceil d/2 \rceil})$ path simplices [BCER95]. A path simplex is one containing a path of d pairwise orthogonal edges.

POLYTOPES

Triangulations of polytopes in \mathbb{R}^d arise in combinatorics and algebra [GKZ90, Sta80]. Several algorithms are known for triangulating the hypercube, but there is a gap between the most efficient algorithm (least number of simplices) and the best lower bound [OS02]; see [Section 17.5.2](#). It is known that the region between

two convex polytopes—a nonconvex polytope—can always be triangulated without Steiner points [GP88]; see [Section 17.3.1](#). Below et al. [BBLR00] have shown that there can be significant differences (linear in the number of vertices) in the minimum numbers of simplices in a triangulation and a *dissection* of a 3D polytope, which is a partition of a polytope into simplices whose faces may meet only partially (for example, a triangle bordering two other triangles along one of its sides).

OPEN PROBLEMS

1. Is the flip graph of the triangulations of a point set or polytope in \mathbb{R}^3 or \mathbb{R}^4 necessarily connected?
 2. What is the asymptotic complexity of the maximum number of triangulations of a set of n points in \mathbb{R}^d ? See [SS02] for results in \mathbb{R}^2 .
 3. Narrow the gap between the upper and lower bounds on the minimum number of simplices in a triangulation of the d -cube.
-

25.7 SOURCES AND RELATED MATERIAL

SURVEYS

For more complete descriptions and references, consult the following sources.

[Aur91]: Describes a number of generalizations of the Voronoi diagram and Delaunay triangulation.

[Ede01]: Geometry relevant to triangular and tetrahedral mesh generation.

[Ber02]: A recent survey of mesh generation algorithms.

[DRS]: A book in preparation, focusing on triangulations in arbitrary dimension.

The World Wide Web currently is a rich source on mesh generation and triangulation; see [Chapter 64](#).

RELATED CHAPTERS

[Chapter 17: Subdivisions and triangulations of polytopes](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 26: Polygons](#)

[Chapter 30: Curve and surface reconstruction](#)

[Chapter 54: Surface simplification and 3D geometry compression](#)

REFERENCES

- [ABK98] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. ACM Conf. SIGGRAPH 98*, pages 415–421, 1998.
- [ACDL00] N. Amenta, S. Choi, T.K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 213–222, 2000.
- [ACK01] N. Amenta, S. Choi, and R.K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Comput. Geom. Theory Appl.*, 19:127–153, 2001.
- [Aur91] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, 1991.
- [Bak89] T.J. Baker. Developments and trends in three-dimensional mesh generation. *Appl. Numer. Math.*, 5:275–304, 1989.
- [BLR00] A. Below, J.A. De Loera, J. Richter-Gebert. Finding minimal triangulations of convex 3-polytopes is NP-hard. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 65–66, 2000.
- [BBLR00] A. Below, U. Brehm, J.A. De Loera, J. Richter-Gebert. Minimal simplicial dissections and triangulations of convex 3-polytopes. *Discrete Comput. Geom.*, 24:35–48, 2000.
- [Ber02] M. Bern. Adaptive mesh generation. In T. Barth and H. Deconinck, editors, *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, pages 1–56. Springer-Verlag, Heidelberg, 2002.
- [BCER95] M. Bern, L.P. Chew, D. Eppstein, and J. Ruppert. Dihedral bounds for mesh generation in high dimensions. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 189–196, 1995.
- [BE92] M. Bern and D. Eppstein. Polynomial-size nonobtuse triangulation of polygons. *Internat. J. Comput. Geom. Appl.*, 2:241–255, 1992.
- [BE95] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry, 2nd Edition*, pages 47–123. World Scientific, Singapore, 1995.
- [BEE⁺93] M. Bern, H. Edelsbrunner, D. Eppstein, S.A. Mitchell, and T.-S. Tan. Edge-insertion for optimal triangulations. *Discrete Comput. Geom.*, 10:47–65, 1993.
- [BEG94] M. Bern, D. Eppstein, and J.R. Gilbert. Provably good mesh generation. *J. Comput. Syst. Sci.*, 48:384–409, 1994.
- [BFS90] L. Billera, P. Filliman, and B. Sturmfelz. Constructions and complexity of secondary polytopes. *Adv. Math.*, 83:155–179, 1990.
- [BGR88] B.S. Baker, E. Grosse, and C.S. Rafferty. Nonobtuse triangulation of polygons. *Discrete Comput. Geom.*, 3:147–168, 1988.
- [BMR94] M. Bern, S.A. Mitchell, and J. Ruppert. Linear-size nonobtuse triangulation of polygons. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 221–230, 1994.
- [Cha84] B. Chazelle. Convex partitions of polyhedra: A lower bound and worst-case optimal algorithm. *SIAM J. Comput.*, 13:488–507, 1984.
- [Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
- [CP90] B. Chazelle and L. Palios. Triangulating a nonconvex polytope. *Discrete Comput. Geom.*, 5:505–526, 1990.

- [CS94] B. Chazelle and N. Shouraboura. Bounds on the size of tetrahedralizations. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 231–239, 1994.
- [DRS] J.A. De Loera, J. Rambau, and F. Santos. *Triangulations of Polyhedra and Point Sets*, in preparation.
- [DKM97] M.T. Dickerson, J.M. Keil, and M.H. Montague. A large subgraph of the minimum weight triangulation. *Discrete Comput. Geom.*, 18:289–304, 1997.
- [Epp94] D. Eppstein. Approximating the minimum weight triangulation. *Discrete Comput. Geom.*, 11:163–191, 1994.
- [Ede01] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge University Press, 2001.
- [EG01] H. Edelsbrunner and D. Guoy. Sink-insertion for mesh improvement. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 115–123, 2001.
- [ET91] H. Edelsbrunner and T.-S. Tan. A quadratic time algorithm for the minmax length triangulation. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 414–423, 1991.
- [ET93] H. Edelsbrunner and T.-S. Tan. An upper bound for conforming Delaunay triangulations. *Discrete Comput. Geom.*, 10:197–213, 1993.
- [ETW92] H. Edelsbrunner, T.-S. Tan, and R. Waupotitsch. A polynomial time algorithm for the minmax angle triangulation. *SIAM J. Sci. Statist. Comput.*, 13:994–1008, 1992.
- [Eri03] J. Erickson. Nice point sets can have nasty Delaunay triangulations. *Discrete Comput. Geom.*, 30:109–132, 2003.
- [For95] S.J. Fortune. Voronoi diagrams and Delaunay triangulations. In F.K. Hwang and D.-Z. Du, editors, *Computing in Euclidean Geometry, 2nd Edition*, pages 225–265. World Scientific, Singapore, 1995.
- [GH97] M. Garland and P.S. Heckbert. Surface simplification using quadric error metrics. In *Proc. ACM Conf. SIGGRAPH 97*, pages 209–216, 1997.
- [GKZ90] I.M. Gelfand, M.M. Kapranov, and A.V. Zelevinsky. Newton polytopes of the classical discriminant and resultant. *Adv. Math.*, 84:237–254, 1990.
- [GP88] J.E. Goodman and J. Pach. Cell decomposition of polytopes by bending. *Israel J. Math.*, 64:129–138, 1988.
- [GSS99] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proc. ACM Conf. SIGGRAPH 99*, pages 325–334, 1999.
- [HS92] J. Hershberger and J. Snoeyink. Convex polygons made from few lines and convex decompositions of polyhedra. In *Proc. 3rd Scand. Workshop Algorithm Theory*, volume 621 of *Lecture Notes Comput. Sci.*, pages 376–387. Springer-Verlag, New York, 1992.
- [Joe89] B. Joe. Three-dimensional triangulations from local transformations. *SIAM J. Sci. Stat. Comput.*, 10:718–741, 1989.
- [KL93] R. Klein and A. Lingas. A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 124–132, 1993.
- [LK96] C. Levcopoulos and D. Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 392–401, 1996.
- [LL86] D.T. Lee and A. Lin. Generalized Delaunay triangulation for planar graphs. *Discrete Comput. Geom.*, 1:201–217, 1986.

- [LL92] C. Levcopoulos and A. Lingas. Fast algorithms for greedy triangulation. *BIT*, 32:280–296, 1992. Also in *Proc. 2nd Scand. Workshop Algorithm Theory*, volume 447 of *Lecture Notes Comput. Sci.*, pages 238–250. Springer-Verlag, New York, 1990.
- [MT⁺95] G.L. Miller, D. Talmor, S.-H. Teng, and N. Walkington. A Delaunay based numerical method for three dimensions: Generation, formulation, and partition. In *Proc. 36th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 683–692, 1995.
- [Mit93] S.A. Mitchell. Refining a triangulation of a planar straight-line graph to eliminate large angles. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 583–591, 1993.
- [Mul94] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1994.
- [MV92] S.A. Mitchell and S.A. Vavasis. Quality mesh generation in three dimensions. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 212–221, 1992.
- [OS02] D. Orden and F. Santos. Asymptotically efficient triangulations of the d -cube. Manuscript, 2002.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [RS92] J. Ruppert and R. Seidel. On the difficulty of tetrahedralizing 3-dimensional non-convex polyhedra. *Discrete Comput. Geom.*, 7:227–253, 1992.
- [Rup93] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 83–92, 1993.
- [San98] F. Santos. A point set whose space of triangulations is disconnected. *J. Amer. Math. Soc.*, 13:611–637, 2000.
- [SS02] F. Santos and R. Seidel. A better upper bound on the number of triangulations of a planar point set. Manuscript, 2002. <http://arxiv.org/abs/math.CO/0204045>.
- [She98] J.R. Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 76–85, 1998.
- [Sta80] R.P. Stanley. Decompositions of rational convex polytopes. *Ann. Discrete Math.*, 6:333–342, 1980.
- [Tan94] T.-S. Tan. An optimal bound for conforming quality triangulations. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 240–249, 1994.
- [TR98] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Trans. Graphics*, 17:84–115, 1998.

26 POLYGONS

Joseph O'Rourke and Subhash Suri

INTRODUCTION

Polygons are among the fundamental building blocks in geometric modeling, and they are used to represent a wide variety of shapes and figures in computer graphics, vision, pattern recognition, robotics, and other computational fields. By a polygon we will mean a region of the plane enclosed by a simple cycle of straight line segments; a *simple cycle* means that nonadjacent segments do not intersect and two adjacent segments intersect only at their common endpoint. This chapter describes a collection of results on polygons with both combinatorial and algorithmic flavors. After classifying polygons in the opening section, Section 26.2 covers polygon decomposition, and Section 26.3 polygon intersection. Sections 26.4 and 26.5, respectively, discuss path finding problems and polygon containment problems. Section 26.6 touches upon a few miscellaneous problems and results.

26.1 POLYGON CLASSIFICATION

Polygons can be classified in several different ways depending on their domain of application. In VLSI applications, for instance, the most commonly used polygons have their sides parallel to the coordinate axes.

GLOSSARY

Simple polygon: A closed region of the plane enclosed by a simple cycle of straight line segments.

Convex polygon: The line segment joining any two points of the polygon lies within the polygon.

Monotone polygon: Any line parallel to some fixed direction intersects the polygon in a single connected piece.

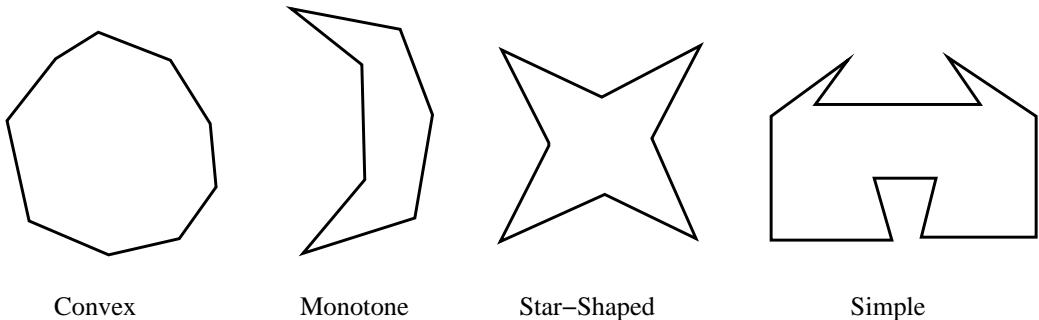
Monotone mountain: A monotone polygon one of whose two monotone chains is a single segment.

Star-shaped polygon: The entire polygon is visible from some point inside the polygon.

Orthogonal polygon: A polygon with sides parallel to the (orthogonal) coordinate axes. Sometimes called a *rectilinear polygon*.

POLYGON TYPES

FIGURE 26.1.1
A classification of polygons.



Before starting our discussion on problems and results concerning polygons, we clarify a few technical issues. The qualifier “simple” in the definition of a simple polygon states a *topological* property, meaning “nonself-intersection.” Not to be confused with “uncomplicated polygons,” in fact, these polygons include the most complex among polygons that are topologically equivalent to a disk (see the classification below). Finally, we will make a standard **general position** assumption throughout this chapter that no three vertices of a polygon are collinear.

The following *hierarchical* classification of polygons is one of the most commonly used (see Figure 26.1.1):

$$\begin{array}{c} \text{STAR-SHAPED} \\ \text{CONVEX} \subset \qquad \qquad \qquad \subset \text{SIMPLE POLYGONS} \\ \text{MONOTONE} \end{array}$$

This hierarchy is best explained using the concept of visibility (see [Chapter 25](#)). We say that two points x and y in a polygon P are mutually **visible** if the line segment \overline{xy} does not intersect the complement of P ; thus the segment \overline{xy} is allowed to graze the polygon boundary but not cross it. We call a set of points $K \subset P$ the **kernel** of P if all points of P are visible from every point in the kernel (see Figure 33.4.4). Then, a polygon P is convex if $K = P$; the polygon is star-shaped if $K \neq \emptyset$; otherwise, the polygon is merely a simple polygon. Speaking somewhat loosely, a monotone polygon can be viewed as a special case of a star-shaped polygon with the exterior kernel at infinity—that is, a monotone polygon can be decomposed into two polygonal chains, each of which is entirely visible from the (same) point at infinity in the extended plane. Notice that the star-shaped polygon in Figure 26.1.1 is also a monotone polygon. The more specialized *monotone mountains* have also proved to be useful intermediate shapes, for, e.g., triangulation [O’R98, Sec. 2.3].

By definition, a simple polygon P is a polygon *without holes*—that is, the interior of the polygon is topologically equivalent to a disk. A **polygon with holes** is a higher-genus variant of a simple polygon, obtained by removing a nonoverlapping

set of strictly interior, simple subpolygons from P . Figure 26.1.2 illustrates the distinction between a simple polygon and a polygon with holes.

An important class of polygons are the *orthogonal polygons*, where all edges are parallel to the coordinate axes. These polygons arise quite naturally in certain applications such as VLSI design, and often algorithms are faster on these more structured polygons.

It would be useful to have a clear notion of a “random polygon” so that algorithms could be tested for typical rather than worst-case behavior. This leads to the issue of generating the *simple polygonalizations* of a fixed point set, a simple polygon whose vertices are the points. This has been solved only in special cases, e.g., for computing the number of monotone simple polygonalizations [ZSSM96], or via heuristic methods [AH96]. One impediment is the following unresolved question.

OPEN PROBLEM

Simple polygonalization: Can the number of simple polygonalizations of a set of n points in the plane be computed in polynomial time?

26.2 POLYGON DECOMPOSITION

Many computational geometry algorithms that operate on polygons first decompose them into more elementary pieces, such as triangles or quadrilaterals. There is a substantial body of literature in computational geometry on this subject. The most celebrated problem in this category is the “polygon triangulation problem.”

GLOSSARY

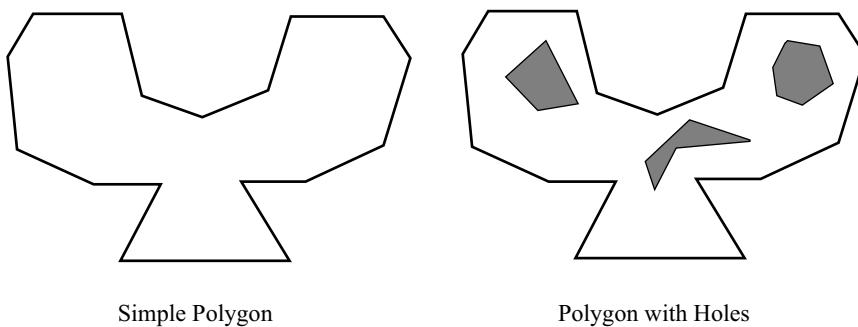
Steiner point: A vertex not part of the input set.

Diagonal: A line segment connecting two polygon nonadjacent vertices and contained in the polygon. An *edge* connects adjacent vertices.

Polygon cover: A collection of subpolygons whose union is exactly the input polygon.

FIGURE 26.1.2

Examples of a simple polygon and a polygon with holes.



Polygon partition: A collection of subpolygons with *pairwise disjoint* interiors whose union is exactly the input polygon.

Dissection: A dissection of one polygon P to another Q is a partition of P into a finite number of pieces that may be reassembled to form Q .

TRIANGULATION

The polygon triangulation problem is to dissect a polygon into triangles by drawing a maximal number of noncrossing diagonals. Only the vertices of the polygon are used as triangle vertices, and no additional (Steiner) vertices are allowed. It is an easy and well-known result that every simple polygon can be triangulated, and that the number of triangles is invariant over all triangulations. More precisely:

THEOREM 26.2.1

Every simple polygon admits a triangulation, and every triangulation of an n -vertex polygon has $n - 3$ diagonals and $n - 2$ triangles.

The number of possible diagonals in a polygon may vary from linear (e.g., a spiral polygon) to quadratic (e.g., a convex polygon). A diagonal that breaks the polygon into two roughly equal halves is called a **balanced** diagonal. In designing his $O(n \log n)$ time algorithm for triangulating a polygon, Chazelle [Cha82] proved the following fact, which has found numerous applications in divide-and-conquer based algorithms for polygons:

THEOREM 26.2.2

Every n -vertex simple polygon admits a diagonal that breaks the polygon into two subpolygons, neither one with more than $\lceil 2n/3 \rceil + 1$ vertices.

By recursively dividing the polygon using balanced diagonals, we get a balanced decomposition of P , which can be modeled by a tree of height $O(\log n)$. The existence of a balanced diagonal follows easily once we consider the graph-theoretic dual of a triangulation. This dual graph of a polygon triangulation is a tree, with maximum node degree three. Diagonals of the triangulation correspond to the edges of the dual tree, and thus a balanced diagonal corresponds to an edge whose removal breaks the tree into two subtrees, each with at most $\lceil 2n/3 \rceil + 1$ nodes.

The problem of computing a triangulation of a polygon has had a long and distinguished history [O'R87], culminating in Chazelle's linear-time algorithm [Cha91]. [Table 26.2.1](#) lists some of the best-known algorithms for this problem. The algorithm in [Sei91] is a randomized Las Vegas algorithm (see [Chapter 34](#)). All others are deterministic algorithms, with worst-case time bounds as shown.

Chazelle's deterministic linear-time algorithm is formidably complex, but has led to a simpler randomized algorithm that runs in linear expected time [AGR01].

Finally, if the polygon contains holes, then it has been shown that $\Theta(n \log n)$ time is both necessary and sufficient for triangulating the region [HM85]. See [Table 26.2.2](#).

TABLE 26.2.1 Results on triangulating a simple polygon.

TIME COMPLEXITY	ALGORITHM	SOURCE
$O(n \log n)$	monotone pieces	[GJPT78]
$O(n \log n)$	divide-and-conquer	[Cha82]
$O(n \log n)$	plane sweep	[HM85]
$O(n \log^* n)$	randomized	[Sei91]
$O(n)$	polygon cutting	[Cha91]

TABLE 26.2.2 Results on triangulating a polygon with holes.

TIME COMPLEXITY	ALGORITHM	SOURCES
$O(n \log n)$	plane sweep	[HM85]
$O(n \log n)$	local sweep	[RR94]

COVERS AND PARTITIONS

The problem of decomposing polygons into different types of simpler polygons has numerous applications within and outside computational geometry (see, e.g., Chapter 43). Unlike the triangulation problem, most variants of the covering and partitioning problems turn out to be provably hard. In a covering problem, the goal is to cover the interior of the polygon with the smallest number of subpolygons of a particular type, for instance, convex or star-shaped polygons. Table 26.2.3 lists results for various polygon covering problems. In this table, “cover type” refers to the family of polygons allowed in the cover, while “domain” refers to the polygonal region that needs to be covered. For the most part, we consider only four types of domains: simple polygons, with and without holes, and orthogonal polygons, with and without (orthogonal) holes. In all of these problems, the cover or partition pieces are allowed to use Steiner points for their vertices. Almost all variations of the covering problem are intractable. The last important open problem in this area, determining the complexity of covering polygons by convex pieces, was settled in [CR88]; this paper also serves as a good source of pointers to related work on polygon covering problems. It remains unclear if their NP-hardness proof could be adapted to settle the same question without Steiner points.

The polygon-partitioning problems are similar to the covering problem, except that the tesselating pieces are not allowed to overlap. Table 26.2.4 collects results on polygon partitioning problems permitting Steiner points. Polynomial-time algorithms can be achieved for simple polygons using the dynamic programming technique. The same problems, however, turn out to be intractable when the polygon has holes. Disallowing Steiner points also leads to polynomial-time algorithms. For example, partitioning a polygon without holes into the fewest convex pieces, not employing Steiner points, is achievable in $O(n^3 \log n)$ time [Kei85, KS98].

Two useful references for polygon partitioning problems are [AAI86] and [Kei85]. The latter presents several polynomial-time algorithms for optimally partitioning

TABLE 26.2.3 Results on polygon covering problems.

COVER TYPE	DOMAIN	HOLES	COMPLEXITY	SOURCE
Rectangles	orthogonal	Y	NP-complete	[Mas78]
Convex-star	polygons	Y	NP-hard	[OS83]
Star	polygons	N	NP-hard	[Agg84]
Rectangles	orthogonal	N	NP-hard	[CR94]
Convex	polygons	N	NP-hard	[CR94]

TABLE 26.2.4 Results on polygon partitioning problems.

PARTITION	DOMAIN	HOLES	COMPLEXITY	SOURCE
Convex	polygons	N	$O(n^3)$	[CD79]
Convex	polygons	Y	NP-hard	[CD79]
Trapezoids	polygons	N	$O(n^2)$	[Kei85]
Trapezoids	polygons	Y	NP-complete	[AAI86]
Rectangles	orthogonal	Y	$O(n^{3/2} \log n)$	[LLL ⁺ 79, OSTT83]

a simple polygon into convex pieces *without* using Steiner points. See [Chapter 43](#) for applications of polygon decomposition problems.

The intractability of most covering and partitioning problems naturally leads to the question of approximability—how well can we approximate the size of an optimal cover or partition in polynomial time. In many cases, there are only a polynomial number of covering candidates—for instance, rectangle covers or convex polygon covers. In these cases, a greedy set-cover heuristic can be used to achieve an approximation factor of $O(\log n)$.

FAT PARTITIONS

Because many algorithms work faster on “fat” shapes, partitioning polygons into fat pieces has become a recent focus. One notion of fatness asks for a partition into convex polygons that minimizes the largest aspect ratio of any piece of the partition. The *aspect ratio* of a polygon P is the ratio of the diameters of the smallest circumscribing circle to the largest inscribed circle. Thus, the fatness corresponds to circularity. If Steiner points are disallowed, i.e., if the pieces of the partition must have their vertices chosen among P ’s vertices, then a polynomial-time algorithm is known [DI02]. Permitting Steiner points leads to considerable complexity. For example, the optimal partition of an equilateral triangle needs an infinite number of pieces, and the optimal partition for a square is not yet known [DO03]. See [Figure 26.2.1](#).

ORTHOGONAL POLYGONS

Partitions and covers of orthogonal polygons into rectangles were mentioned above. With the goal achieving the fewest number of rectangles, finding optimal covers is NP-complete, whereas finding optimal partitions is polynomial, $O(n^{3/2} \log n)$. If the goal is to minimize the total length of the “cuts” between the rectangles (minimum “ink”), then an optimum partition can be found in $O(n^4)$ time for poly-

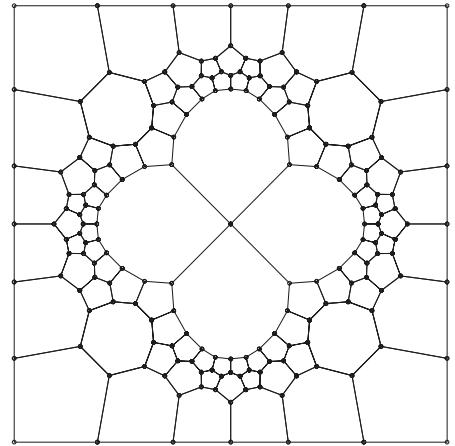


FIGURE 26.2.1

A 92-piece partition achieving an aspect ratio of 1.29950, the smallest so far achieved. ([DO03])

gons without holes, but is NP-complete with holes [LTL89]. Approximations are available; for example, one that guarantees a solution within a factor of 3 of the minimum length [GZ90]. For the goal of maximizing the shortest rectangle side over all rectangles in the partition (a type of “fat” partition, motivated by VLSI chip masking), a polynomial-time algorithm is known for polygons without holes [OT02]. See Figure 26.2.2 for such a partition, here only employing cuts incident to vertices.

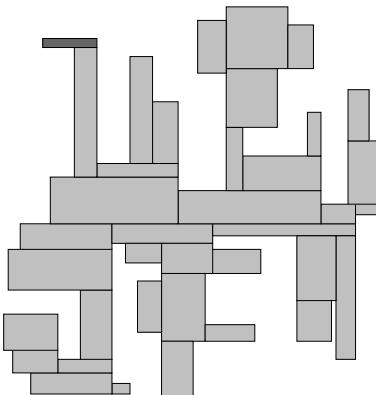


FIGURE 26.2.2

38-rectangle partition of a $n = 82$ vertex orthogonal polygon. The dark rectangle is the thinnest.

Covering orthogonal polygons without holes with the fewest squares is polynomial, $O(n^{3/2})$, but NP-complete for polygons with holes [ACKO88].

AREA BISECTION

A particularly useful partition of a polygon P is an **area bisection**: a line determining a halfplane H such that $H \cap P$ and $\bar{H} \cap P$ have the same area. In [DO90] an $O(n \log n)$ algorithm for area bisection was developed, and then used to “ham-sandwich section” a pair of polygons. Motivated by positioning parts in industrial

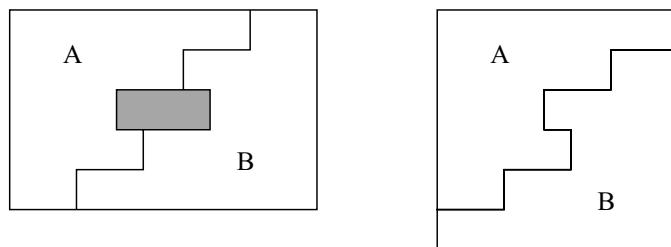
part-feeding systems, Böhringer et al. [BDH99] developed an output-size sensitive algorithm for computing the complete set of combinatorially distinct area bisectors, which they show can have size $\Omega(n^2)$.

DISSECTIONS

A *dissection* of one polygon P to another Q is a partition of P into a finite number of pieces that may be reassembled to form Q . P and Q are then said to be *equidecomposable*. Dissections have been studied as puzzles for centuries. A typical example is shown in Figure 26.2.3 [Fre97, p. 66]. It has been known since the early

FIGURE 26.2.3

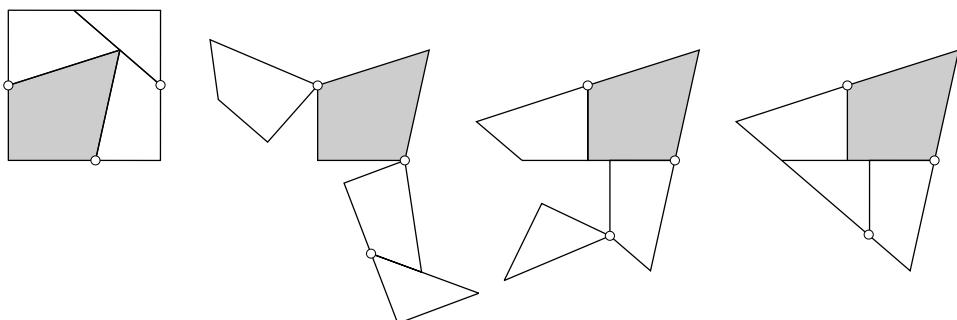
Sam Loyd's "A&P Baking Powder" puzzle reassembles a rectangle with a hole to a rectangle without a hole via a two-piece dissection.



19th century that any two polygons of equal area are equidecomposable [Fre97, p. 221]. The same question for the more constrained *hinged dissections* remains unresolved. See Fig. 26.2.4 for the famous Dudeney-McElroy hinged dissection between a square and an equilateral triangle [Fre02]. Partial results here are that any two polyominoes (Chapter 15) of the same area have a hinged dissection [DDE⁺03], and any asymmetric polygon has a hinged dissection to its mirror image [Epp01].

FIGURE 26.2.4

A four-piece hinged dissection between a square and an equilateral triangle.



OPEN PROBLEMS

1. *Convex cover without Steiner points:* What is the complexity of covering a polygon without holes by convex pieces, without employing Steiner points? The Culberson-Reckhow NP-hardness proof [CR94] uses Steiner points.
 2. *Approximating the number of art gallery guards:* Give a polynomial-time algorithm for computing a constant-factor approximation of the minimum number of point guards needed to cover a simple polygon.
 3. *Fat partition of a square:* What is the optimal partition of a square into “fat” convex polygons?
 4. *Hinged dissections:* Does every pair of equal-area polygons have a hinged dissection?
-

26.3 POLYGON INTERSECTION

Polygon intersection problems deal with issues of detection and computation of the collision between two polygonal shapes. In the detection problem, one is only interested in deciding *whether* the two polygons have a point in common. In the intersection computation problem, the algorithm is asked to report the overlapping parts of the two polygons. Such problems arise naturally in robotics and computer games; see [Chapter 33](#) for additional material.

The maximum *number* of points at which two polygons may cross each other depends on the type of polygons. If p and q , respectively, denote the number of vertices of the two polygons, then the maximum number of intersections is $\min(2p, 2q)$ if both polygons are convex, $\max(2p, 2q)$ if one is convex, and pq otherwise.

Algorithmically, intersection-detection between convex polygons can be done significantly faster than intersection computation, if we allow reasonable preprocessing of polygons. By a reasonable preprocessing, we mean that the preprocessing algorithm takes into account the *structure* of the polygons but *not their positions*. In Table 26.3.1, n denotes the total number of vertices in the two polygons; that is, $n = p + q$.

TABLE 26.3.1 Intersecting polygons.

POLYGON TYPES	PREPROCESSING	QUERY	SOURCE
Convex-convex	$O(1)$	$O(n)$	[CD80]
Convex-convex	$O(n)$	$O(\log n)$	[CD80]
Simple-simple	$O(1)$	$O(n)$	[Cha91]
Simple-simple		$O(m \log^2 n)$	[Mou92]

The parameter m in the query time for intersections of two simple polygons is the complexity of a *minimum link witness* for the intersection or disjointness of the

two polygons, and we always have $m \leq n$. The preprocessing space requirement is linear when the polygons are preprocessed.

26.4 PATHS IN POLYGONS

Path planning in polygons is another well-studied area of research. An abstract robot motion planning problem ([Chapter 40](#)) is to find a shortest path for a point in the midst of a collection of disjoint polygons in the plane. This simplified scenario lets us focus exclusively on the combinatorial aspect of the robotics problem, ignoring such practical issues as kinematics and control ([Chapter 41](#)). The polygons represent obstacles in the path of the robot, which itself is modeled as a point. The free space is the set of all points accessible to the robot via a free path. By convention, for the case of a single polygon, the free space is defined to be the closed interior of the polygon (think of an art gallery).

GLOSSARY

Free space: The complement of the union of the interiors of obstacle polygons.

Free path: A path lying entirely in the free space.

Shortest path: A free path of minimum total length.

Shortest path tree: The union of shortest paths from one fixed vertex to all other vertices. (Strictly speaking, this may not be a tree in special cases.)

Shortest path map: The minimal partition of the plane with respect to a fixed source point s so that all points in a region have the same combinatorial structure for their shortest path to s , i.e., the list of vertices on the path is the same. See [Figure 24.0.1](#).

Geodesic diameter: The maximum shortest path distance between any points.

Geodesic center: A point minimizing the maximum shortest path distance to all other points.

Minimum link path: An obstacle-avoiding path between two given points with the minimum number of edges.

Link distance: The link distance between two points p and q is the minimum number of straight-line segments needed in any free path connecting p and q .

Window partition: The window partition of a polygon P with respect to a source point s (or a line segment) is the minimal partition of P into regions with the property that all points in a region have the same link distance to s .

EUCLIDEAN MEASURE

The problem of computing a shortest Euclidean path between two points in the presence of polygonal obstacles is one of the best-known problems of computational geometry (see [Chapter 24](#)). The geometry of the Euclidean plane ensures that the shortest path is a nonself-intersecting polygonal path with corners at obstacle vertices. [Figure 26.4.1](#) shows an example of a shortest path problem. A *shortest*

path tree [GHL⁺87] extends the notion of a single shortest path to shortest paths to all vertices of the polygonal domain from a specified source point.

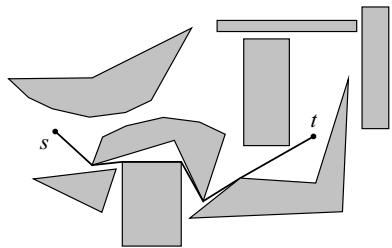


FIGURE 26.4.1
A shortest path among polygons.

The shortest path distance function is a metric, and therefore several natural measures lend themselves to our new setting: in particular, the *shortest path diameter* (also called the *geodesic diameter*) and the *geodesic center*. The following table summarizes the main results known today for these shortest path problems. The long-standing open problem of computing a shortest path map in optimal time was settled only recently [HS93]. A related question is computing the shortest diagonal in a simple polygon. It may be found in linear time [HS97].

TABLE 26.4.1 Results for Euclidean shortest paths in the plane.

PROBLEM	DOMAIN	RESULT	SOURCE
Shortest path	simple polygon	$O(n \log n)$	[GM91]
Shortest path tree	simple polygon	$O(n \log n)$	[GHL ⁺ 86]
Shortest path tree	triang. simple poly.	$O(n)$	[HS91]
Geodesic diameter	simple polygon	$O(n)$	[AT87]
Geodesic center	simple polygon	$O(n \log n)$	[PSR89]
Shortest path tree	polygon with holes	$O(E + n \log n)$	[GM91]
Shortest path map	polygon with holes	$O(n \log n)$	[HS93]

In the Table 26.4.1, the use of a triangulated polygon in [GHL⁺87, HS91] is meant to separate the cost of triangulating the polygon from the cost of computing a shortest path tree. However, since the publication of these results, a linear-time algorithm for polygon triangulation has been achieved [Cha91], making this distinction unnecessary. Interest in the geodesic diameter and center was partly motivated by Lantuejoul and Maisonneuve [LM84], who proposed these measures for quantitative image analysis.

SHORTEST PATH QUERIES

Often it is desirable to preprocess a polygon (with or without holes) to speed up subsequent query answering. For the case where the domain is a simple polygon and all queries are with respect to a fixed source point, an optimal data structure is presented in [GHL⁺87]. Essentially, the shortest path tree implicitly partitions the polygon into regions that have the same shortest path structure. In combination with a point-location data structure, this partition achieves $O(\log n)$ query time using $O(n)$ space. When the source point is not fixed, the problem is more difficult and requires more advanced data structuring methods. Nevertheless, an optimal solution is known with $O(n)$ space and $O(\log n)$ query time [GH89, Her91].

For polygons with holes, only the case of a fixed source point is satisfactorily solved: the algorithm of Hershberger and Suri [HS93] computes an $O(n)$ -space shortest path map, which can be used to answer queries in $O(\log n)$ time apiece. When the source is not fixed, we know of no sublinear time query algorithm! The most promising direction in this case is via fast approximation algorithms; only recently has some progress been made in this direction. An algorithm by Chen [Che95] takes $O(n^{3/2} \log n)$ space and $O(\log n)$ query time to compute a $(6 + \epsilon)$ -approximation of the shortest path distance.

LINK MEASURE

Another measure of distance that has received considerable attention in computational geometry is the link distance [Sur87, Sur90]. The motivation behind the link distance comes from situations where the cost of “turning” outweighs the cost of straight-line travel. Figure 26.4.2 shows an example of a minimum link path which is not a shortest path.

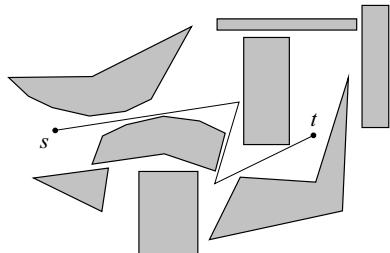


FIGURE 26.4.2
A minimum link path.

For link distance problems in a simple polygon, a construction known as a window partition has proved to be very useful [Sur90]. A window partition is best explained using the idea of visibility. All points of the polygon directly visible from s are at link distance one. Call this set V_1 . The boundary between the visible and invisible region of the polygon consists of a collection of chords, called *windows* of V_1 . The points in $P \setminus V_1$ that are visible from some point of a window of V_1 form the region with link distance two. Repeating this construction yield the window partition of P . For a fixed source point in a simple polygon, the *window tree* data structure of Suri [Sur90] yields the optimal query time of $O(\log n)$ using $O(n)$ space.

When both source and destination are specified as part of the query, the best data structure known is due to Arkin, Mitchell, and Suri [AMS92], achieving $O(\log n)$ time but at the expense of $O(n^3)$ space. Further results on link and geodesic queries can be found in Chiang and Tamassia [CT94], and Section 24.3 of this Handbook.

The problems of computing a shortest path, the diameter, and the center all extend to the link measure, and Table 26.4.2 summarizes the known results for these problems.

TABLE 26.4.2 Results for minimum link path problems.

PROBLEM	DOMAIN	RESULT	SOURCE
Min link path	triang simple poly	$O(n)$	[Sur87, Sur90]
Min link tree	triang simple poly	$O(n)$	[Sur87, Sur90, GM90]
Orthogonal min link path	orthogonal obstacles	$O(n \log n)$	[OSTT83]
Link diameter	simple polygon	$O(n \log n)$	[Sur87]
Link center	simple polygon	$O(n \log n)$	[Ke89, DLS92]
Link dist query	simple poly, arbitrary s, t	$O(\log n)$, $O(n^3)$ space	[AMS92]

The result in [DLS92], achieving $O(n \log n)$ for both link center and radius of a simple polygon, is the culmination of the window trees ideas initiated in [Sur87] and further articulated in [Ke89].

VISIBILITY AND RAY SHOOTING

Algorithms and data structures for computing visibility have come to occupy an important role in computational geometry, in large part due to their successful application in solving other problems. In a polyhedral environment modeling a real-life scene, determining what is visible from a particular location has obvious relevance to the problem of robot motion planning. The ray shooting problem represents a very specific instance of visibility computation: determine the first point of contact between a query ray and the polyhedral scene. In addition to obvious applications in collision-detection, the ray shooting problem also plays a fundamental role in designing other computational geometry algorithms, such as data structures for the equally important “point-location” problem.

The topic of computing the visibility region of a point, line segment, or other objects is treated in [Chapter 25](#). In the present section, we cover the results on ray shooting, which are presented in [Table 26.4.3](#).

The query performance in the case of polygons with holes is sensitive to the number of holes—if the number of holes is $k \leq n$, then the query time for the last two algorithms improves to $O(\sqrt{k} \log n)$, with preprocessing cost $O(n\sqrt{k} + n \log n + k^{3/2} \log k)$.

TABLE 26.4.3 Results for the ray shooting problem in polygons.

DOMAIN	PREPROCESSING	QUERY	SOURCE
Convex polygon	$O(n)$	$O(\log n)$	
Simple polygon	$O(n)$	$O(\log n)$	[GHL ⁺ 87]
Polygon with holes	$O(n)$	$O(\sqrt{n} \log n)$	[HS93]

OPEN PROBLEMS

1. *Shortest path query problem:* Build a data structure to compute shortest-path distance between pairs of query points in the presence of polygonal obstacles. The goal is to achieve $O(n \log n)$ space, $O(\log n)$ query time, and $O(1)$ approximation factor on the distance. (The constant of approximation should be small, say, at most 2.) No sublinear query algorithm for the exact problem is known.
2. *Non-Steiner minimum link path problem:* Given a simple polygon P and a pair of points $p, q \in P$, find a minimum link path in P from p to q subject to the condition that the path turns only at the vertices of P . Can this problem be solved in $O(n \log n)$ time?

26.5 POLYGON CONTAINMENT

Polygon containment refers to a class of problems that deals with the placement of one polygonal figure inside another. Polygon inscription, polygon circumscription, and polygon nesting are other variants of this type of problem.

GLOSSARY

Inscribed polygon: We will say that a polygon Q is inscribed in polygon P if $Q \subset P$. P is then called a *circumscribing polygon*.

Polygon nesting: P, Q is a nested pair if $Q \subset P$ or vice versa.

CONTAINMENT OF POLYGONS

Let P, Q be two simple polygons with p and q vertices, respectively. The polygon containment problem asks for the largest copy of Q that can be contained in P using rotations and translations. (In this section, all scalings are assumed to be *uniform*; thus “shearing” is not permitted.) Several authors have considered the polygonal containment problem under various restrictions on the shape of the polygons and the allowable motions. Table 26.5.1 collects the best results known for the most important cases. See Section 47.4 for a description of the near-linear λ_s function.

 TABLE 26.5.1 Results for the polygon containment problem.

P	Q	TRANSFORMS	RESULTS	SOURCE
Ortho-convex	ortho-convex	translate, scale	$O((p+q)^2 \log pq)$	[For85]
Convex	convex	translate, scale	$O((p+q)^2 \log pq)$	[For85]
Convex	convex	translate, rotate	$O(qp^2)$	[Cha83]
Simple	simple	translate, rotate	$O(p^3q^3 \log pq)$	[AB88]
Convex	polygon w holes	translate, rotate, scale	$O(q^4p\lambda_4(pq) \log p)$	[CK93]
Convex	points	translate, rotate, scale	$O(q^2p\lambda_3(pq) \log p)$	[CK93]

It has been shown recently that the decision problem—whether there exists a transformation of Q that permits it to be contained in P —is 3SUM-hard, under a variety of allowable transformations [BHP01]. Thus it is unlikely the above complexities can be pushed below quadratic.

Considerable work has focused on packing shapes for its practical applications. For example, the apparel industry is interested in packing clothing patterns on a bolt of cloth efficiently. Much of the progress on this inherently intractable problem has proceeded by studying particular containment problems. See, e.g., [Mil96, DMR97, Mil99]. Finally, a number of specialized results are available. For example, there is an $O(n \log n)$ randomized algorithm for placing two equal-radius disks in a convex polygon, a problem with application to facility location [KSY00]. Finding the largest pair of equal-radius disks in an arbitrary simple polygon has a surprising application to folding polygons [BDD⁺98], and can be found again in $O(n \log n)$ randomized time [BMV01].

INSCRIBING/CIRCUMSCRIBING POLYGONS

We now consider problems related to inscribing and circumscribing polygons. In these problems, a polygon P is given, and the task is to find a polygon Q of some specified number of vertices k that is inscribed in (resp. circumscribes) P while maximizing (resp. minimizing) certain measure of Q . The common measures include area and perimeter. See Table 26.5.2 for results concerning this class of problems; n denotes the number of vertices of P . See references [AP88] and [MS90] for these results and other relevant material on this problem. The latest addition is [BM02], an improvement of the $O(n \log n)$ minimum perimeter algorithm of [AP88] to $O(n)$.

 TABLE 26.5.2 Inscribing and circumscribing polygons.

TYPE	k	P	MEASURE	RESULTS	SOURCE
Inscribe	3	convex	max area	$O(n)$	[DS79]
Inscribe	k	convex	max area/perimeter	$O(kn + n \log n)$	[AKM ⁺ 87]
Inscribe	convex	simple	max area	$O(n^7)$	[CY86]
Inscribe	3	simple	max area/perimeter	$O(n^4)$	[MS90]
Circumscribe	3	convex	min area	$O(n)$	[OAMB86]
Circumscribe	3	convex	min perimeter	$O(n)$	[BM02]
Circumscribe	k	convex	min area	$O(kn + n \log n)$	[AP88]

NESTING POLYGONS

The nested polygon problem asks for a polygon with the smallest number of vertices that fits between two nested polygons. More precisely, given two nested polygons P and Q , where $Q \subset P$, find a polygon K of the least number of vertices such that $Q \subset K \subset P$. Generalizing the notion of nested polygons, one can also pose the problem of determining a polygonal subdivision of the least number of *edges* that “separates” a family of polygons. Table 26.5.3 lists the results on these problems. In this table, n is the total number of vertices in the input polygons, while k is the number of vertices in the output polygon (or subdivision). Reference [MS92] is a good source of pointers to other results on polygon nesting problems.

TABLE 26.5.3 Results for polygon nesting.

TYPES OF P, Q	TYPE OF K	RESULTS	SOURCE
Convex-convex	convex	$O(n \log k)$	[ABO ⁺ 89]
Simple-simple	simple	$O(n \log k)$	[Gho91]
Polygonal family	subdivision	NP-complete	[Das90]
Polygonal family	subdivision	$O(1)$ -Opt in $O(n \log n)$	[MS92]

Several other results on polygon nesting have been obtained. In particular, if the minimum-vertex nested polygon is nonconvex, then it can be found in $O(n)$ time [GM90]. There is also a relation here to *offset polygons* (Chapter 56), e.g., [BBDG98].

26.6 MISCELLANEOUS

There is a rather large number of results pertaining to polygons, and it would be impossible to cover them all in a single chapter. Having focused on a selected list of topics so far, we now provide below an unorganized collection of some miscellaneous results.

POLYGON MORPHING

To *morph* one polygon into another is to find a continuous deformation from the source polygon to the target polygon. Guibas and Hershberger [GH94] introduce the problem of morphing a simple polygon P to another simple polygon Q whose edges, taken in counterclockwise order, are parallel to the corresponding edges of P and oriented the same way. An atomic morphing step is a uniform scaling or translation of a part of the polygon. It is shown in [GH94] that $O(n^{4/3+\epsilon})$ morphing steps are always sufficient to convert one polygon to another. This result was improved shortly afterward by Hershberger and Suri [HS95], who reduced the number of morphing steps to $O(n \log n)$.

An alternative approach to morphing is suggested by *polyhedral reconstruction*: Given two polygons lying in parallel planes, construct an interpolating polyhedron whose top and bottom faces are the two given polygons and all intermediate slices are simple polygons. See [Chapter 26](#) for more details on reconstruction problems.

FLIPPING POCKETS

Let a *pocket* of a polygon P be a region bounded by a subchain of the polygon edges and an edge of the convex hull of P , the *pocket lid*. Every nonconvex polygon has at least one pocket. Erdős defined a *flip* as a rotation of a pocket's chain of edges into 3D about the pocket lid by 180° , landing the subchain back in the plane of the polygon, and asked [Erd35] whether every polygon may be convexified by a finite number of simultaneous pocket flips. The answer is YES [dSN39], although no bound may be placed on the number of required flips as a function of the number of polygon vertices n .

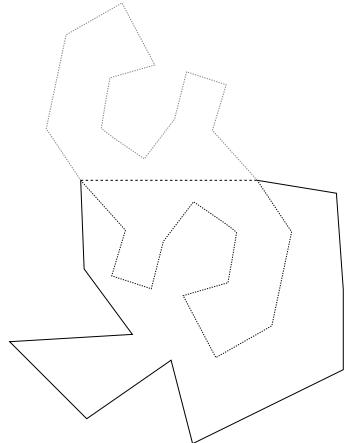


FIGURE 26.6.1
A *flipturn* about pocket lid ab .

This motivates the *flipturn* operation, which rotates by 180° a subchain bounding a pocket of the polygon, not in 3D about the pocket lid, but in 2D around the midpoint of the lid; see Figure 26.6.1. It was established in [ACD⁺02] that the length of the longest convexifying flipturn sequence is at most $n^2/4 - O(1)$. Whether there might be a smaller upper bound remains open.

For related questions of moving between polygons whose vertices are defined by a fixed point set, via flips or other local transformations, see [HHH02].

CSG REPRESENTATION

In [DGHS88] Dobkin et al. consider the problem of deriving a *Peterson-style* formula given the boundary representation of a simple polygon. A Peterson-style formula is a “constructive solid geometry” representation, in which the polygon is presented as a set of Boolean operations; see [Chapter 47](#). Peterson proved that every simple polygon in two dimensions admits a representation by a Boolean formula

on the halfplanes supporting the edges of the polygon. Furthermore, the resulting formula is *monotone*; that is, there is no negation and each halfplane appears exactly once. Dobkin et al. consider the algorithmic problem of constructing such a formula, and give an $O(n \log n)$ time algorithm, where n is the number of vertices of the polygon. Interestingly, it turns out that not all 3D polyhedra admit a Peterson-style formula [DGHS88].

POLYGON SEARCHING

In these problems, the goal is to design *on-line* search strategies for locating an (identifiable) object in a polygon; the word “on-line” means that the searcher does not have a complete knowledge of the polygon, rather it “discovers” the polygon during its navigation. The motivation stems from robotics applications. Table 26.6.1 summarizes some basic results on this class of problems. (The parameter k in the last line denotes the number of distinct initial placements of the robot having the same visibility polygon.) References [IK95] and [DRW98] provide a good starting point for a search on this topic.

TABLE 26.6.1 Results for polygon searching.

ENVIRONMENT	GOAL	COMPETITIVE RATIO	SOURCE
n oriented rectangles	shortest path	$\Theta(\sqrt{n})$	[BRS91]
“Street” polygon	shortest path	$1 + \frac{3}{2}\pi$	[Kle92]
Gen. Streets	shortest path	9.06-Opt	[DI99]
Star-shaped polygon	reach kernel	≈ 5.52	[IK95]
Orthogonal polygon	exploration	randomized $5/4$	[Kle94]
Simple polygon	localization with min travel	$(k-1)$ -Opt	[DRW98]
Simple polygon	shortest watchman tour	26.5-Opt	[HIKK01]

THREE-DIMENSIONAL POLYGONS

A 3D polygon is an unknotted closed chain of segments in \mathbb{R}^3 such that adjacent segments share an endpoint, and nonadjacent segments do not intersect. A triangulation of a 3D polygon has the same combinatorial structure as a triangulation of a planar polygon—all triangle vertices are polygon vertices, each polygon edge is a side of one triangle, each diagonal is shared by exactly two triangles—with the surface they define a nonself-intersecting topological disk. This disk is said to *span* the polygon. Barequet et al. proved that determining whether a 3D polygon has a triangulation in this sense is NP-complete [BDE98]. Another negative result along the same lines is that there exist 3D polygons of n vertices that can only be spanned by nonself-intersecting piecewise-linear disks which, when triangulated, need $2^{\Omega(n)}$ triangles [HST03]. Note that here the triangle vertices are not necessarily polygon vertices, i.e., Steiner points are (necessarily) used. This exponential lower bound shows that *knot triviality* algorithms (which check whether a closed chain is the trivial “unknot”) that search for such spanning disks necessarily

lead to exponential-time algorithms. This unknotting problem is known to be in NP [HLP97].

OPEN PROBLEMS

1. *Natural morphing*: The transformation in [GH94] is not very natural: it morphs the source polygon to a simple intermediate shape, and then expands it to the target polygon. Explore a more natural morphing transformation.
2. *Morphing with holes*: Investigate the morphing problem for polygons with holes.
3. *3D Peterson formulas*: Characterize the 3D polyhedra that can be represented by Peterson-style formulas.
4. *Shortest flipturn sequence*: Is there a subquadratic upper bound on the length of the shortest flipturn sequence to convexify a polygon of n vertices?

26.7 SOURCES AND RELATED MATERIAL

SURVEYS

The survey article by Mitchell and Suri [MS95] addresses optimization problems in computational geometry, many involving polygons. Keil surveys polygon decomposition algorithms in [Kei00]. Link distance problems are surveyed in [MSD00].

RELATED CHAPTERS

- [Chapter 25: Triangulations](#)
[Chapter 27: Shortest paths and networks](#)
[Chapter 28: Visibility](#)
[Chapter 34: Point location](#)
[Chapter 51: Pattern recognition](#)
[Chapter 58: Geographic information systems](#)

REFERENCES

- [AAI86] Ta. Asano, Te. Asano, and H. Imai. Partitioning a polygonal region into trapezoids. *J. Assoc. Comput. Mach.*, 33:290–312, 1986.
- [AB88] F. Aurenhammer and J.-D. Boissonnat. Polygon placement under translation and rotation. *Proc. 5th Sympos. Theoret. Aspects Comput. Sci.*, volume 294 of *Lecture Notes Comput. Sci.*, pages 322–333. Springer-Verlag, Berlin, 1988.

- [ABO⁺89] A. Aggarwal, H. Booth, J. O'Rourke, S. Suri, and C.K. Yap. Finding minimal convex nested polygons. *Inform. Comput.*, 83:98–110, 1989.
- [ACD⁺02] O. Aichholzer, C. Cortés, E.D. Demaine, V. Dujmović, J. Erickson, H. Meijer, M.H. Overmars, B. Palop, S. Ramaswami, and G.T. Toussaint. Flipturning polygons. *Discrete Comput. Geom.*, 28:231–253, 2002.
- [ACKO88] L.J. Aupperle, H.E. Conn, J.M. Keil, and J. O'Rourke. Covering orthogonal polygons with squares. In *Proc. 26th Allerton Conf. Commun. Control Comput.*, pages 97–106, 1988.
- [Agg84] A. Aggarwal. *The art gallery problem: Its variations, applications, and algorithmic aspects*. Ph.D. thesis, Dept. of Comput. Sci., Johns Hopkins Univ., Baltimore, 1984.
- [AGR01] N.M. Amato, M.T. Goodrich, and E.A. Ramos. A randomized algorithm for triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 26:245–265, 2001.
- [AH96] T. Auer and M. Held. Heuristics for the generation of random polygons. In *Proc. 8th Canad. Conf. Comput. Geom.*, pages 38–43, 1996.
- [AKM⁺87] A. Aggarwal, M.M. Klawe, S. Moran, P.W. Shor, and R. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2:195–208, 1987.
- [AMS92] E.M. Arkin, J.S.B. Mitchell, and S. Suri. Optimal link path queries in a simple polygon. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 269–279, 1992.
- [AP88] A. Aggarwal and J.K. Park. Notes on searching in multidimensional monotone arrays. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 497–512, 1988.
- [AT87] Te. Asano and G.T. Toussaint. Computing the geodesic center of a simple polygon. In D.S. Johnson, editor, *Perspectives in Computing: Discrete Algorithms and Complexity*, pages 65–79. Academic Press, Boston, 1987.
- [BBDG98] G. Barequet, A.J. Briggs, M.T. Dickerson, and M.T. Goodrich. Offset-polygon annulus placement problems. *Comput. Geom. Theory Appl.*, 11:125–141, 1998.
- [BDD⁺98] T.C. Biedl, E.D. Demaine, M.L. Demaine, A. Lubiw, and G.T. Toussaint. Hiding disks in folded polygons. In *Proc. 10th Canad. Conf. Comput. Geom.*, pages 10–12, 1998.
- [BDE98] G. Barequet, M.T. Dickerson, and D. Eppstein. On triangulating three-dimensional polygons. *Comput. Geom. Theory Appl.*, 10:155–170, 1998.
- [BDH99] K.-F. Böhringer, B.R. Donald, and D. Halperin. The area bisectors of a polygon. *Discrete Comput. Geom.*, 22:269–285, 1999.
- [BHP01] G. Barequet and S. Har-Peled. Polygon containment and translational min-Hausdorff-distance between segment sets are 3SUM-hard. *Internat. J. Comput. Geom. Appl.*, 11:465–474, 2001.
- [BM02] B.K. Bhattacharya and A. Mukhopadhyay. On the minimum perimeter triangle enclosing a convex polygon. In *Japan Conf. Discrete Comput. Geom.*, pages 19–20, Tokyo, 2002.
- [BMV01] P. Bose, P. Morin, and A. Vigneron. Packing two disks into a polygonal environment. volume 2108 of *Lecture Notes Comput. Sci.*, Springer-Verlag, Berlin, pages 142–149, 2001.
- [BRS91] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 494–503, 1991.
- [CD79] B. Chazelle and D.P. Dobkin. Decomposing a polygon into its convex parts. In *Proc. 11th Annu. ACM Sympos. Theory Comput.*, pages 38–48, 1979.
- [CD80] B. Chazelle and D.P. Dobkin. Detection is easier than computation. In *Proc. 12th Annu. ACM Sympos. Theory Comput.*, pages 146–153, 1980.

- [Cha82] B. Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci.*, pages 339–349, 1982.
- [Cha83] B. Chazelle. The polygon containment problem. In F.P. Preparata, editor, *Computational Geometry*, volume 1 of *Adv. Comput. Res.*, pages 1–33. JAI Press, Greenwich, 1983.
- [Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
- [Che95] D.Z. Chen. On the all-pairs Euclidean shortest path problem. In *Proc. 6th Annu. Symp. Discrete Algorithms*, 1995.
- [CK93] L.P. Chew and K. Kedem. Placing the largest similar copy of a convex polygon among polygonal obstacles. *Comput. Geom. Theory Appl.*, 3:59–89, 1993.
- [CR88] J.C. Culberson and R.A. Reckhow. Covering polygons is hard. In *Proc. 29th Annu. IEEE Symp. Found. Comput. Sci.*, pages 601–611, 1988.
- [CR94] J.C. Culberson and R.A. Reckhow. Covering polygons is hard. *J. Algorithms*, 17:2–24, 1994.
- [CT94] Y.-J. Chiang and R. Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons in the presence of obstacles. Report CS-94-03, Comput. Sci. Dept., Brown Univ., Providence, 1994.
- [CY86] J.S. Chang and C.K. Yap. A polynomial solution for the potato-peeling problem. *Discrete Comput. Geom.*, 1:155–182, 1986.
- [Das90] G. Das. *Approximation schemes in computational geometry*. Ph.D. thesis, Univ. of Wisconsin, 1990.
- [DDE⁺03] E.D. Demaine, M.L. Demaine, D. Eppstein, G.N. Frederickson, and E. Friedman. Hinged dissection of polyominoes and polyforms. *Comput. Geom. Theory Appl.*, 2003.
- [DGHS88] D.P. Dobkin, L.J. Guibas, J. Hershberger, and J. Snoeyink. An efficient algorithm for finding the CSG representation of a simple polygon. *Proc. ACM Conf. SIGGRAPH 88*, pages 31–40, 1988.
- [DI99] A. Datta and C. Icking. Competitive searching in a generalized street. *Comput. Geom. Theory Appl.*, 13:109–120, 1999.
- [DI02] M. Damian. Exact and approximation algorithms for computing optimal α -fat decompositions. In *Proc. 14th Canad. Conf. Comput. Geom.*, pages 93–96, 2002.
- [DLS92] H.N. Djidjev, A. Lingas, and J.-R. Sack. An $O(n \log n)$ algorithm for computing the link center of a simple polygon. *Discrete Comput. Geom.*, 8:131–152, 1992.
- [DMR97] K. Daniels, V.J. Milenkovic, and D. Roth. Finding the largest area axis-parallel rectangle in a polygon. *Comput. Geom. Theory Appl.*, 7:125–148, 1997.
- [DO90] M. Díaz and J. O'Rourke. Ham-sandwich sectioning of polygons. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 282–286, 1990.
- [DO03] M. Damian and J. O'Rourke. Partitioning regular polygons into circular pieces I: Convex partitions. In *Proc. 15th Canad. Conf. Comput. Geom.*, pages 43–46, 2003.
- [DRW98] G. Dudek, K. Romanik, and S.H. Whitesides. Localizing a robot with minimum travel. *SIAM J. Comput.*, 27:583–604, 1998.
- [DS79] D.P. Dobkin and L. Snyder. On a general method for maximizing and minimizing among certain geometric problems. In *Proc. 20th Annu. IEEE Symp. Found. Comput. Sci.*, pages 9–17, 1979.
- [dSN39] B. Szökefalvi Nagy. Solution to problem 3763. *Amer. Math. Monthly*, 46:176–177, 1939.

- [Epp01] D. Eppstein. Hinged kite mirror dissection. ACM Computing Research Repository, 2001. arXiv:cs.CG/0106032.
- [Erd35] P. Erdős. Problem 3763. *Amer. Math. Monthly*, 42:627, 1935.
- [For85] S.J. Fortune. A fast algorithm for polygon containment by translation. In *Proc. 12th Internat. Colloq. Automata Lang. Program.*, volume 194 of *Lecture Notes Comput. Sci.*, pages 189–198. Springer-Verlag, Berlin, 1985.
- [Fre97] G.N. Frederickson. *Dissections: Plane and Fancy*. Cambridge University Press, 1997.
- [Fre02] G.N. Frederickson. *Hinged Dissections: Swinging & Twisting*. Cambridge University Press, 2002.
- [GH89] L.J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39:126–152, 1989.
- [GH94] L.J. Guibas, and J. Hershberger. Morphing simple polygons. *Proc. 10th Annu. Sympos. Comput. Geom.*, pages 267–276, 1994.
- [GHL⁺87] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- [Gho91] S.K. Ghosh. Computing visibility polygon from a convex set and related problems. *J. Algorithms*, 12:75–95, 1991.
- [GHL⁺86] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear time algorithms for visibility and shortest path problems inside simple polygons. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 1–13, 1986.
- [GJPT78] M.R. Garey, D.S. Johnson, F.P. Preparata, and R.E. Tarjan. Triangulating a simple polygon. *Inform. Process. Lett.*, 7:175–179, 1978.
- [GM90] S.K. Ghosh and A. Maheshwari. An optimal algorithm for computing a minimum nested nonconvex polygon. *Inform. Process. Lett.*, 36:277–280, 1990.
- [GM91] S.K. Ghosh and D.M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.
- [GZ90] T. Gonzalez and S.-Q. Zheng. Approximation algorithms for partitioning a rectangle with interior points. *Algorithmica*, 5:11–42, 1990.
- [Her91] J. Hershberger. A new data structure for shortest path queries in a simple polygon. *Inform. Process. Lett.*, 38:231–235, 1991.
- [HHH02] C. Hernando, F. Hurtado, and M.E. Houle. On local transformation of polygons with visibility properties. *Theoret. Comput. Sci.*, 289:919–937, 2002.
- [HIKK01] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem. *SIAM J. Comput.*, 31:577–600, 2001.
- [HLP97] J. Hass, J.C. Lagarias, and N. Pippenger. The computational complexity of knot and link problems. In *IEEE Sympos. Found. Comput. Sci.*, pages 172–181, 1997.
- [HM85] S. Hertel and K. Mehlhorn. Fast triangulation of the plane with respect to simple polygons. *Inform. Control*, 64:52–76, 1985.
- [HS91] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes Comput. Sci.*, pages 331–342. Springer-Verlag, Berlin, 1991.
- [HS93] J. Hershberger and S. Suri. Efficient computation of Euclidean shortest paths in the plane. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 508–517, 1993.

- [HS93] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 54–63, 1993.
- [HS95] J. Hershberger, and S. Suri. Morphing Binary Trees. *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 396–404, 1995.
- [HS97] J. Hershberger and S. Suri. Finding a shortest diagonal of a simple polygon in linear time. *Comput. Geom. Theory Appl.*, 7:149–160, 1997.
- [HST03] J. Hass, J. Snoeyink, and W.P. Thurston. The size of spanning disks for polygonal curves. *Discrete Comput. Geom.*, 29:1–18, 2003.
- [Ke89] Y. Ke. An efficient algorithm for link-distance problems. In *Proc. 5th Annu. ACM Symp. Comput. Geom.*, pages 69–78, 1989.
- [Kei85] J.M. Keil. Decomposing a polygon into simpler components. *SIAM J. Comput.*, 14:799–817, 1985.
- [Kei00] J.M. Keil. Polygon decomposition. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 491–518. Elsevier North-Holland, Amsterdam, 2000.
- [Kle92] R. Klein. Walking an unknown street with bounded detour. *Comput. Geom. Theory Appl.*, 1:325–351, 1992.
- [Kle94] J. Kleinberg. On-line search in a simple polygon. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 8–15, 1994.
- [KS98] J.M. Keil and J. Snoeyink. On the time bound for convex decomposition of simple polygons. In *Proc. 10th Canad. Conf. Comput. Geom.*, pages 54–55, Montréal, 1998.
- [KSY00] S.K. Kim, C.-S. Shin, and T.-C. Yang. Placing two disks in a convex polygon. *Inform. Process. Lett.*, 73:33–39, 2000.
- [IK95] C. Icking, and R. Klein. Searching for the kernel of a polygon—A competitive strategy. *Proc. 11th Annu. ACM Symp. Comput. Geom.*, pages 258–266, 1995.
- [LLL⁺79] W. Lipski, Jr., E. Lodi, F. Luccio, C. Mugnai, and L. Pagli. On two-dimensional data organization, Part II. *Fundam. Inform.*, 2:245–260, 1979.
- [LM84] C. Lantuejoul, and F. Maisonneuve. Geodesic methods in quantitative image analysis. *Pattern Recogn.*, 17:177–187, 1984.
- [LTL89] W.T. Liou, J.J.M. Tan, and R.C.T. Lee. Minimum partitioning simple rectilinear polygons in $O(n \log \log n)$ time. In *Proc. 5th Annu. ACM Symp. Comput. Geom.*, pages 344–353, 1989.
- [Mas78] W.J. Masek. Some NP-complete set covering problems. Manuscript, MIT, 1978.
- [Mil96] V.J. Milenkovic. Translational polygon containment and minimal enclosure using linear programming based restriction. In *Proc. 28th Annu. ACM Symp. Theory Comput.*, pages 109–118, 1996.
- [Mil99] V.J. Milenkovic. Rotational polygon containment and minimum enclosure using only robust 2D constructions. *Comput. Geom.*, 13:3–19, 1999.
- [Mou92] D.M. Mount. Intersection detection and separators for simple polygons. In *Proc. 8th Annu. ACM Symp. Comput. Geom.*, pages 303–311, 1992.
- [MS90] E.A. Melissaratos and D.L. Souvaine. On solving geometric optimization problems using shortest paths. In *Proc. 6th Annu. ACM Symp. Comput. Geom.*, pages 350–359, 1990.
- [MS92] J.S.B. Mitchell and S. Suri. Separation and approximation of polyhedral surfaces. In *Proc. 3rd ACM-SIAM Symp. Discrete Algorithms*, pages 296–306, 1992.

- [MS95] J.S.B. Mitchell, and S. Suri. Geometric algorithms. In M.O. Ball, T.L. Magnati, C.L. Monma, and G.L. Nemhauser, editors, *Handbook of Operations Research/Management Science*, pages 425–479. Elsevier, Amsterdam, 1995.
- [MSD00] A. Maheshwari, J.-R. Sack, H.N. Djidjev. Link distance problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 519–558. Elsevier North-Holland, Amsterdam, 2000.
- [OAMB86] J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin. An optimal algorithm for finding minimal enclosing triangles. *J. Algorithms*, 7:258–269, 1986.
- [O'R87] J. O'Rourke. *Art Gallery Theorems and Algorithms. The Internat. Series of Monographs on Computer Science*. Oxford University Press, New York, 1987.
- [O'R98] J. O'Rourke. *Computational Geometry in C*, second edition. Cambridge University Press, 1998.
- [OS83] J. O'Rourke and K.J. Supowit. Some NP-hard polygon decomposition problems. *IEEE Trans. Inform. Theory*, IT-30:181–190, 1983.
- [OSTT83] T. Ohtsuki, M. Sato, M. Tachibana, and S. Torii. Minimum partitioning of rectilinear regions. *Trans. Inform. Processing Soc. Japan*, 1983.
- [OT02] J. O'Rourke and G. Tewari. Partitioning orthogonal polygons into fat rectangles in polynomial time. In *Proc. 14th Canad. Conf. Comput. Geom.*, pages 97–100, 2002.
- [RR94] R. Ronfard and J. Rossignac. Triangulating multiply-connected polygons: A simple yet efficient algorithm. *Comput. Graph. Forum*, 13:C281–292, 1994.
- [PSR89] R. Pollack, M. Sharir, and G. Rote. Computing of the geodesic center of a simple polygon. *Discrete Comput. Geom.*, 4:611–626, 1989.
- [Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1:51–64, 1991.
- [Sur87] S. Suri. *Minimum link paths in polygons and related problems*. Ph.D. thesis, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, 1987.
- [Sur90] S. Suri. On some link distance problems in a simple polygon. *IEEE Trans. Robot. Autom.*, 6:108–113, 1990.
- [ZSSM96] C. Zhu, G. Sundaram, J. Snoeyink, and J.S.B. Mitchell. Generating random polygons with given vertices. *Comput. Geom. Theory Appl.*, 6:277–290, 1996.

27 SHORTEST PATHS AND NETWORKS

Joseph S.B. Mitchell

INTRODUCTION

Computing an optimal path in a geometric domain is a fundamental problem in computational geometry, with applications in robotics, geographic information systems (GIS), wire routing, etc.

A taxonomy of shortest-path problems arises from several parameters that define the problem:

1. Objective function: the length of the path may be measured according to the Euclidean metric, an L_p metric, the number of links, a combination of criteria, etc.
2. Constraints on the path: the path may have to get from s to t while visiting a specified set of points or regions along the way.
3. Input geometry: the map of the geometric domain also specifies constraints on the path, requiring it to avoid various types of obstacles.
4. Type of moving object: the object to be moved along the path may be a single point or may be a robot of some specified geometry.
5. Dimension of the problem: often the problem is in 2 or 3 dimensions, but higher dimensions arise in some applications.
6. Single shot vs. repetitive mode queries.
7. Static vs. dynamic environments: in some cases, obstacles may be inserted or deleted or may be moving in time.
8. Exact vs. approximate algorithms.
9. Known vs. unknown map: the on-line version of the problem requires that the moving robot sense and discover the shape of the environment along its way.

We survey various forms of the problem, primarily in two and three dimensions, for motion of a single point, since most results have focused on these cases. We discuss shortest paths in a simple polygon (Section 27.1), shortest paths among obstacles (Section 27.2), and other metrics for length (Section 27.3). We also survey other related network optimization problems (Section 27.4). Higher dimensions are discussed in Section 27.5. Finally, in Section 27.6, we survey results on t -spanners and their application to shortest paths and network optimization.

GLOSSARY

Polygonal s - t path: A path from point s to point t consisting of a finite number of line segments (*edges*, or *links*) joining a sequence of points (*vertices*).

Length of a path: A nonnegative number associated with a path, measuring its total cost according to some prescribed metric. Unless otherwise specified, the length will be the Euclidean length of the path.

Shortest/optimal/geodesic path: A path of minimum length among all paths that are feasible (satisfying all imposed constraints). See Figure 27.0.1.

Shortest-path distance: The metric induced by a shortest-path problem. The shortest-path distance between s and t is the length of a shortest s - t path; in many geometric contexts, it is also referred to as *geodesic distance*.

Locally shortest/optimal path: A path that cannot be improved by making a small change to it that preserves its *combinatorial structure* (e.g., the ordered sequence of triangles visited, for some triangulation of a polygonal domain P); also known as a *taut-string* path in the case of a shortest obstacle-avoiding path.

Simple polygon P of n vertices: A closed, simply-connected region whose boundary is a union of n (straight) line segments (edges), whose endpoints are the vertices of P .

Polygonal domain of n vertices and h holes: A closed, multiply-connected region whose boundary is a union of n line segments, forming $h+1$ closed (polygonal) cycles. A simple polygon is a polygonal domain with $h=0$.

Triangulation of a simple polygon P : A decomposition of P into triangles such that any two triangles intersect in either a common vertex, a common edge, or not at all. A triangulation of P can be computed in $O(n)$ time. See Section 25.2.

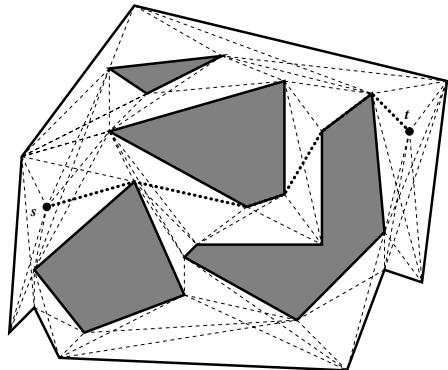


FIGURE 27.0.1

The visibility graph $VG(P)$. Edges of $VG(P)$ are of two types: (1) the heavy dark boundary edges of P , and (2) the edges that intersect the interior of P , shown with thin dashed segments. A shortest s - t path is highlighted.

Obstacle: A region of space whose interior is forbidden to paths. The complement of the set of obstacles is the *free space*. If the free space is a polygonal domain P , the obstacles are the $h+1$ connected components (*h holes*, plus the *face at infinity*) of the complement of P .

Visibility graph $VG(P)$: A graph whose nodes are the vertices of P and whose edges join pairs of nodes for which the corresponding segment lies inside P . See Chapter 27. An example is shown in Figure 27.0.1.

Single-source query: A query that specifies a goal point t , and requests the length of a shortest path from a *fixed* source point s to t . The query may also require that a shortest s - t path be reported; in general, this can be done in additional time $O(k)$, where k is the number of edges in the output path.

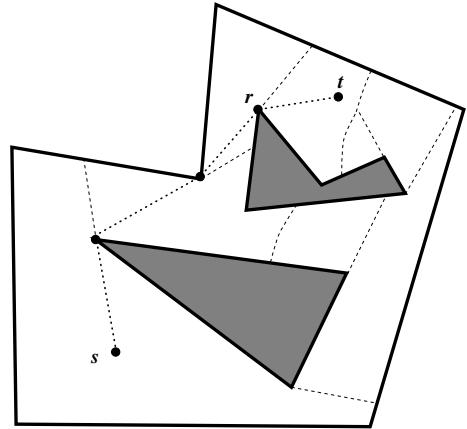


FIGURE 27.0.2

A shortest path map with respect to source point s within a polygonal domain. The dotted path indicates the shortest s - t path, which reaches t via the root r of its cell.

Shortest path map, SPM(s): A decomposition of free space into regions (**cells**) according to the “combinatorial structure” of shortest paths from a fixed source point s to points in the regions. Specifically, for shortest paths in a polygonal domain, SPM(s) is a decomposition of P into cells such that for all points t interior to a cell, the sequence of obstacle vertices along an s - t path is fixed. In particular, the *last* obstacle vertex along a shortest s - t path is the **root** of the cell containing t . Each cell is **star-shaped** with respect to its root, which lies on the boundary of the cell. See Figure 27.0.2, where the root of the cell containing t is labeled r . If SPM(s) is preprocessed for point location (see [Chapter 34](#)), then single-source queries can be answered efficiently by locating the query point t within the decomposition.

Two-point query: A query that specifies two points, s and t , and requests the length of a shortest path between them. It may also request that a path be reported.

Geodesic Voronoi diagram (VD): A Voronoi diagram for a set of **sites**, in which the underlying metric is the geodesic distance. See [Chapters 23](#) and [25](#).

Geodesic center of P : A point within P that minimizes the maximum of the shortest-path lengths to any other point in P .

Geodesic diameter of P : The length of a longest shortest path between a pair of points $s, t \in P$; s and t are vertices for any longest s - t shortest path.

27.1 PATHS IN A SIMPLE POLYGON

The most basic geometric shortest-path problem is to find a shortest path inside a **simple** polygon P (having no holes), connecting two points, s and t . The complement of P serves as an obstacle through which the path is not allowed to travel. In this case, there is a unique taut-string path from s to t , since there is only one way

to “thread” a string through a simply-connected region.

Algorithms for computing a shortest s - t path begin with a triangulation of P ($O(n)$ time; Section 25.2), whose dual graph is a tree. The **sleeve** is comprised of the triangles that correspond to the (unique) path in the dual that joins the triangle containing s to that containing t . By considering the effect of adding the triangles in order along the sleeve, it is not hard to obtain an $O(n)$ time algorithm for collapsing the sleeve into a shortest path. At a generic step of the algorithm, the sleeve has been collapsed to a structure called a **funnel** (with *base* ab and *root* r) consisting of the shortest path from s to a vertex r , and two (concave) shortest paths joining r to the endpoints of the segment ab that bounds the triangle abc processed next (see Figure 27.1.1). In adding triangle abc , we “split” the funnel in two according to the taut-string path from r to c , which will, in general, include a segment uc joining c to some (vertex) point of tangency u , along one of the two concave chains of the funnel. After the split, we keep that funnel (with base ac or bc) that contains the s - t taut-string path. The work needed to search for u can easily be charged off to those vertices that are discarded from further consideration. Thus, a shortest s - t path is found in time $O(n)$, which is worst-case optimal.

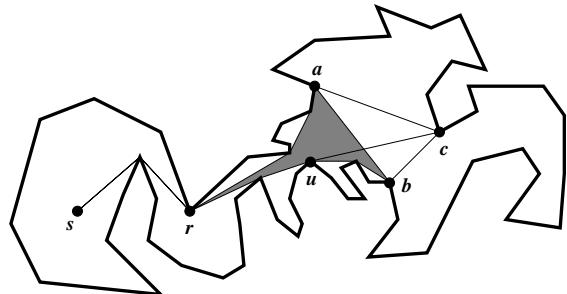


FIGURE 27.1.1
Splitting a funnel.

SHORTEST PATH MAPS

The shortest path map $\text{SPM}(s)$ for a simple polygon has a particularly simple structure, since the boundaries between cells in the map are (line segment) chords of P obtained by extending appropriate edges of the visibility graph $\text{VG}(P)$. It can be computed in time $O(n)$ by using somewhat more sophisticated data structures to do funnel splitting efficiently; in this case, we cannot discard one side of each split funnel. Single-source queries can be answered in $O(\log n)$ time, after storing the $\text{SPM}(s)$ in an appropriate $O(n)$ -size point location data structure (see [Chapter 34](#)). $\text{SPM}(s)$ includes a tree of shortest paths from s to every vertex of P .

TWO-POINT QUERIES

A simple polygon can be preprocessed in time $O(n)$, into a data structure of size $O(n)$, to support shortest-path queries between any two points $s, t \in P$. In time $O(\log n)$ the length of the shortest path can be reported, and in additional time $O(k)$, the shortest path can be reported, where k is the number of vertices in the output path [GH89].

TABLE 27.1.1 Shortest paths and geodesic distance in simple polygons.

PROBLEM VERSION	COMPLEXITY	NOTES	SOURCE
Shortest $s-t$ path Single-source query; SPM(s)	$O(n)$ $O(\log n)$ query $O(n)$ preproc/space	builds SPM(s)	[LP84] [GHL ⁺ 87]
Two-point query	$O(\log n)$ query		[GH89]
Two-polygon query	$O(n)$ preproc/space $O(\log k + \log n)$ query $O(n)$ space	between convex k -gons in simple n -gon	[CT97]
Dynamic two-point query	$O(\log^2 n)$ update/query $O(n)$ space		[GT97]
Dynamic two-polygon query	$O(\log k + \log^2 n)$ query $O(\log^2 n)$ update $O(n)$ space	between convex k -gons in simple n -gon	[CT97]
Parallel algorithm (CREW PRAM)	$O(\log n)$ time		[Her95]
Geodesic VD	$O(n/\log n)$ processors		[PL98]
All nearest neighbors	$O((n+k)\log(n+k))$		[HS97]
Geodesic farthest-site VD	$O(n)$ $O((n+k)\log(n+k))$ time $O(n+k)$ space	k point sites for set of vertices k point sites	[ÅFW93]
All farthest neighbors	$O(n)$		[HS97]
Geodesic diameter	$O(n)$		[HS97]
Geodesic center	$O(n \log n)$	for set of vertices	[PSR89]

DYNAMIC VERSION

In the dynamic version of the problem, one allows the polygon P to change with addition and deletion of edges and vertices. If the changes are always made in such a way that the set of all edges yields a **connected planar subdivision** of the plane into simple polygons (i.e., no “islands” are created), then one can maintain a data structure of size $O(n)$ that supports two-point query time of $O(\log^2 n)$ (plus $O(k)$ if the path is to be reported), and update time of $O(\log^2 n)$ for each addition/deletion of an edge/vertex [GT93].

OTHER RESULTS

Several other problems studied with respect to geodesic distances induced by a simple polygon are summarized in Table 27.1.1. See also [Table 25.4.1](#).

Shortest paths within simple polygons yield a wealth of structural information about the polygon. In particular, they have been used to give an output-sensitive algorithm for constructing the visibility graph of a simple polygon ([Her89]) and can be used for constructing a **geodesic triangulation** of a simple polygon, which allows for efficient ray-shooting (see [CEG⁺94]). They also form a crucial step in solving **link distance** problems, as we will discuss later.

OPEN PROBLEMS

1. Can one devise a simple $O(n)$ time algorithm for computing the shortest path between two points in a simple polygon, *without* resorting to a (complicated) linear-time triangulation algorithm?

-
2. Can the geodesic Voronoi diagram for k sites within P be computed in time $O(n + k \log k)$?
 3. Can the geodesic center of a simple polygon be computed in $O(n)$ time?
-

27.2 PATHS IN A POLYGONAL DOMAIN

While in a simple polygon there is a unique taut-string path between two points, in a general polygonal domain P , there can be an exponential number of taut-string simple paths between two points.

The homotopy type of a path can be expressed as a sequence (with repetitions) of triangles visited, for some triangulation of P . For any given homotopy type, expressed with N triangles, a shortest path of that type can be computed in $O(N)$ time [HS94]. Efficient algorithms for computing a set of homotopic shortest paths among obstacles, for many pairs of start and goal points, have been recently given [Bes03, EKL02]. One can also efficiently test, in time $O(n \log n)$, if two simple paths are of the same homotopy type in a polygonal domain; here, n is the total number of vertices of the input paths and the polygonal domain [CLMS02].

SEARCHING THE VISIBILITY GRAPH

Without loss of generality, we can assume that s and t are vertices of P (since we can make “point” holes in P at s and t). It is easy to show that any locally optimal s - t path must lie on the visibility graph $\text{VG}(P)$ ([Figure 27.0.1](#)). We can construct $\text{VG}(P)$ in output-sensitive time $O(E_{\text{VG}} + n \log n)$, where E_{VG} denotes the number of edges of $\text{VG}(P)$ [GM91], even if we allow only $O(n)$ working space [PV95]. Given the graph $\text{VG}(P)$, whose edges are weighted by their Euclidean lengths, we can use Dijkstra’s algorithm to construct a tree of shortest paths from s to all vertices of P , in time $O(E_{\text{VG}} + n \log n)$ [FT87]. Thus, Euclidean shortest paths among obstacles in the plane can be computed in time $O(E_{\text{VG}} + n \log n)$. This bound is worst-case quadratic in n , since $E_{\text{VG}} \leq \binom{n}{2}$; note too that domains exist with $E_{\text{VG}} = \Omega(n^2)$.

Given the tree of shortest paths from s , we can compute $\text{SPM}(s)$ in time $O(n \log n)$, by computing an additive weight Voronoi diagram (see [Chapter 23](#)) of the vertices, with each vertex weighted by its distance from s .

CONTINUOUS DIJKSTRA METHOD

Instead of searching the visibility graph (which may have quadratic size), an alternative paradigm for shortest-path problems is to construct the (linear-size) shortest path map directly. The *continuous Dijkstra* method was developed for this purpose.

Building on the success of the method in solving (in nearly linear time) the shortest-path problem for the L_1 metric, Mitchell [Mit96] developed a version of the continuous Dijkstra method applicable to the Euclidean shortest-path problem, obtaining the first subquadratic ($O(n^{1.5+\epsilon})$) time bound. Subsequently, this result was improved by Hershberger and Suri [HS99], who achieve a nearly optimal algo-

rithm based also on the continuous Dijkstra method. They give an $O(n \log n)$ time and $O(n \log n)$ space algorithm, coming close to the lower bounds of $\Omega(n + h \log h)$ time and $O(n)$ space.

The continuous Dijkstra paradigm involves simulating the effect of a wavefront propagating out from the source point, s . The **wavefront** at distance δ from s is the set of all points of P that are at geodesic distance δ from s . It consists of a set of curve pieces, called **wavelets**, which are arcs of circles centered at obstacle vertices that have already been reached. At certain critical “events,” the structure of the wavefront changes due to one of the following possibilities:

- (1) a wavelet disappears (due to the closure of a cell of the SPM);
- (2) a wavelet collides with an obstacle vertex;
- (3) a wavelet collides with another wavelet; or
- (4) a wavelet collides with an obstacle edge at a point interior to that edge.

It is not difficult to see from the fact that $\text{SPM}(s)$ has linear size, that the total number of such events is $O(n)$. The challenge in applying this propagation scheme is devising an efficient method to know *what* events are going to occur and in being able to *process* each event as it occurs (updating the combinatorial structure of the wavefront).

One approach, used in [Mit96], is to track a “pseudo-wavefront,” which is allowed to run over itself, and to “clip” only when a wavelet collides with a vertex that has already been labeled due to an earlier event. Detection of when a wavelet collides with a vertex is accomplished with range-searching techniques. An alternative approach, used in [HS99], simplifies the problem by first decomposing the domain P using a *conforming subdivision*, which allows one to propagate an approximate wavefront on a cell-by-cell basis. A key property of a conforming subdivision is that any edge of length L of the subdivision has only a constant number of (constant-sized) cells within geodesic distance L .

APPROXIMATION ALGORITHMS

One can compute approximate Euclidean shortest paths using standard methods of discretizing the set of directions. Clarkson [Cla87] gives an algorithm that uses $O((n \log n)/\epsilon)$ time to build a data structure of size $O(n/\epsilon)$, after which a $(1 + \epsilon)$ -approximate shortest path query can be answered in time $O(n \log n + n/\epsilon)$. (These bounds rely also on an observation in [Che95].) Using a related approach, based on approximating Euclidean distance with fixed orientation distances, Mitchell [Mit92] computes a $(1 + \epsilon)$ -approximate shortest path in time $O((n \log n)/\sqrt{\epsilon})$ using $O(n/\sqrt{\epsilon})$ space. Chen, Das, and Smid [CDS01] have shown an $\Omega(n \log n)$ lower bound, in the algebraic computation tree model, on the time required to compute a $(1 + \epsilon)$ -approximate shortest path.

TWO-POINT QUERIES

Two-point queries in a polygonal domain are much more challenging than the case of simple polygons, where optimal algorithms are known. One natural approach (observed by Chen et al. [CDK01]) is to store the shortest path map, $\text{SPM}(v)$,

rooted at each vertex v ; this requires $O(n^2)$ space. Then, for a query pair (s, t) , we compute the set of k_s vertices visible to s and k_t vertices visible to t , in time $O(\min\{k_s, k_t\} \log n)$, using the visibility complex of Pocchiola and Vegter [PV93]. Then, assuming that $k_s \leq k_t$, we simply locate t in each of the k_s SPM's rooted at the vertices visible from s . This permits two-point queries to be answered in time $O(\min\{k_s, k_t\} \log n)$, which is worst-case $\Omega(n \log n)$, making it no better than computing a shortest path from scratch, in the worst case.

Methods for exact two-point queries that are efficient in the worst case utilize an **equivalence decomposition** of the domain P , for which all points z within a cell of the decomposition have topologically equivalent shortest path maps. Given query points s and t , one locates s within the decomposition, and then uses the resulting SPM, along with a parametric point location data structure, to locate t within the SPM with respect to s . The complexity of the decomposition can be quite high; there can be $\Omega(n^4)$ topologically distinct shortest path maps with respect to points within P . Chiang and Mitchell [CM99] have utilized this approach to obtain various tradeoffs between space and query time; see [Table 27.2.1](#). Unfortunately, the space bounds are all impractically high.

More efficient methods allow one to approximately answer two-point queries. As observed in [Che95], the method of Clarkson [Cla87] can be used to construct a data structure of size $O(n^2 + n/\epsilon)$ in $O(n^2 \log n + (n/\epsilon) \log n)$ time, so that two-point $(1 + \epsilon)$ -optimal queries can be answered in time $O((\log n)/\epsilon)$, for any fixed $\epsilon > 0$. Chen [Che95] was the first to obtain nearly *linear*-space data structures for approximate shortest path queries; these were obtained, though, at the cost of a higher approximation factor. He obtains a $(6 + \epsilon)$ -approximation, using $O(n^{3/2}/\log^{1/2} n)$ time to build a data structure of size $O(n \log n)$, after which queries can be answered in time $O(\log n)$. The best current bounds are given by Arikati et al. [ACC⁺96], who give a spectrum of results based on planar t -spanners (see [Section 27.6](#)), with tradeoffs among the approximation factor and the preprocessing time, storage space, and query time. One such result gives a $(3\sqrt{2} + \epsilon)$ -approximation in query time $O(\log n)$, after using $O(n^{3/2}/\log^{1/2} n)$ time to build a data structure of size $O(n \log n)$.

In the special case that the polygonal domain is “ t -rounded,” meaning that the shortest path distance between any two vertices is at most some constant t times the Euclidean distance between them, Gudmundsson et al. [GLNS02a, GLNS02b] show that in query time $O(\log n)$, one can give a $(1 + \epsilon)$ -approximate answer to a two-point shortest path query while using only $O(n \log n)$ space and preprocessing time. Their result utilizes approximate distance oracles in t -spanner graphs, giving $O(1)$ -time approximate distance queries between pairs of vertices; see [Section 27.6](#).

OTHER RESULTS

The geodesic Voronoi diagram of k sites inside P can be constructed in time $O((n + k) \log(n + k))$, using the continuous Dijkstra method, simply starting with multiple source points. While the geodesic center/diameter problem has been carefully examined for the case of simple polygons, we are unaware of results, beyond brute force, for polygonal domains.

[Table 27.2.1](#) summarizes various results.

 TABLE 27.2.1 Shortest paths among planar obstacles, in a polygonal domain.

PROBLEM	COMPLEXITY	NOTES	SOURCE
Shortest s - t path	$O(n \log n)$ $O(n + h^2 \log n)$ $O(n^{1.5+\epsilon})$	$O(n \log n)$ space $O(n)$ space $O(n)$ space	[HS99] [KMM97] [Mit96]
Approx shortest s - t path SPM(s)/geodesic VD	$O((n \log n)/\sqrt{\epsilon})$ $O(n \log n)$ $O(n^{1.5+\epsilon})$	$O(n/\sqrt{\epsilon})$ space $O(n \log n)$ space $O(n)$ space	[Mit92] [HS99] [Mit96]
Two-point query	$O(\log n)$ query	exact	[CM99]
Two-point query	$O(n^{11})$ preproc/space	exact	[CM99]
Two-point query	$O(\log^2 n)$ query		
Two-point query	$O(n^{10} \log n)$ preproc/space		
Two-point query	$O(n^{1-\delta} \log n)$ query	exact	[CM99]
Two-point query	$O(n^{5+10\delta+\epsilon})$ preproc/space	$0 < \delta \leq 1$	
Two-point query	$O(\log n + h)$ query	exact	[CM99]
Two-point query	$O(n^5)$ preproc/space		
Two-point query	$O(h \log n)$ query		
Approx two-point query	$O(n + h^5)$ preproc/space	exact	[CM99]
Approx two-point query	$O(\log n)$ query	($1+\epsilon$)-approx	[Cla87, Che95]
Approx two-point query	$O(n^2)$ space		
Approx two-point query	$O(n^2 \log n)$ preproc		
Approx two-point query	$O(\log n)$ query		
Approx two-point query	$O(n \log n)$ space		
Approx two-point query	$O(n^{3/2}/\log^{1/2} n)$ preproc		
Approx two-point query	$O(\log n)$ query		
Approx two-point query	$O(n \log n)$ space		
Approx two-point query	$O(n \log n)$ preproc	$(1 + \epsilon)$ -approx t -rounded domain	[GLNS02a] [GLNS02b]

OPEN PROBLEMS

1. Can the Euclidean shortest-path problem be solved in $O(n + h \log h)$ time and $O(n)$ space?
2. How efficiently, and using what size data structure, can one preprocess a polygonal domain for exact two-point queries? Can one obtain sublinear queries using a reasonable amount of space (say, subquadratic)?
3. How efficiently can one compute a geodesic center/diameter for a polygonal domain?

27.3 OTHER METRICS FOR LENGTH

In the problems considered so far, the Euclidean metric has been used to measure the length of a path. We consider now several other possible objective functions for measuring path length. Tables 27.3.1 and 27.3.2 summarize results.

GLOSSARY

L_p metric: The L_p distance between $q = (q_x, q_y)$ and $r = (r_x, r_y)$ is given by $d_p(q, r) = [|q_x - r_x|^p + |q_y - r_y|^p]^{1/p}$. The L_p length of a polygonal path is the

sum of the L_p lengths of each edge of the path. Special cases of the L_p metric include the L_1 metric (***Manhattan metric***) and the L_∞ metric ($d_\infty(q, r) = \max\{|q_x - r_x|, |q_y - r_y|\}$).

Rectilinear path: A polygonal path with each edge parallel to a coordinate axis; also known as an ***isothetic*** path.

C-oriented path: A polygonal path with each edge parallel to one of a set C of $c = |C|$ ***fixed orientations***.

Link distance: The minimum number of edges in a polygonal path from s to t within a polygonal domain P . If the paths are restricted to be rectilinear or C -oriented, then we obtain the ***rectilinear link distance*** or ***C-oriented link distance***.

Min-link s - t path: A polygonal path from s to t that achieves the link distance.

Weighted region problem: Given a piecewise-constant function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ that is defined by assigning a nonnegative ***weight*** to each face of a given triangulation in the plane. The ***weighted length*** of an s - t path π is the path integral, $\int_{\pi} f(x, y) d\sigma$, of the weight function along π . The ***weighted region metric*** associated with f defines the distance $d_f(s, t)$ to be the infimum over all s - t paths π of the weighted length of π . The ***weighted region problem*** (WRP) asks for an s - t path of minimum weighted length.

Sailor's problem: Compute a minimum-cost path, where the cost of motion is *direction-dependent*, and there is a cost L per turn (in a polygonal path).

Bounded curvature shortest-path problem: Compute a shortest obstacle-avoiding smooth (C^1) path joining point s , with prescribed velocity orientation, to point t , with prescribed velocity orientation, such that at each point of the path the radius of curvature is at least 1.

Maximum concealment path: A path within polygonal domain P that minimizes the length during which the robot is exposed to a given set of “enemy” observers. This problem is a special case of the weighted region problem, in which weights are 0 (for travel in concealed free space), 1 (for travel in exposed free space), or ∞ (for travel through obstacles).

Total turn for an s - t path: The sum of the absolute values of all turn angles for a polygonal s - t path.

Minimum-time path problem: Find a path to minimize the total time required to move from an initial position, at an initial velocity, to a goal position and velocity, subject to bounds on the allowed acceleration and velocity along the path. This problem is also known as the ***kinodynamic motion planning problem***.

LINK DISTANCE

In the min-link path problem, our goal is to minimize the number of links (and hence the number of turns) in a path connecting s and t . In many problems, the link distance provides a more natural measure of path complexity than the Euclidean length, as well as having applications to curve simplification.

In a simple polygon P , a min-link path can be computed in time $O(n)$, as described in Section 28.4; see also [MSD00] for a survey on link distance. In fact, in

 TABLE 27.3.1 Link distance shortest-path problems.

PROBLEM	COMPLEXITY	NOTES	SOURCE
Min-link path	$O(EVG\alpha^2(n) \log n)$	polygonal domain	[MRW92]
Min-link path	$O(n)$	simple polygon	[Sur86, Sur90]
Rectilinear link path	$O(n \log n)$ time, $O(n)$ space	rectilinear obstacles	[DN91]
Rectilinear link path	$O(n)$	rectilinear simple polygon	[dB91, HS94]
C -oriented link path	$O(c^2 n \log n)$ time	C -oriented obstacles	[AOS94]
Two-point link query	$O(c^2 n \log n)$ space, preproc $O(\log n)$ query	builds SPM(s)	[AMS95]
Two-point rectilinear link query	$O(n^3)$ space, preproc $O(\log n)$ query	simple polygon	[dB91, HS94]
Shortest k -link path	$O(n \log n)$ space, preproc $O(n^3 k^3 \log(Nk/\epsilon^{1/k}))$	rectilinear simple polygon also is L_1 -opt simple polygon	[MPA92]

time $O(n)$ a **window partition** of P with respect to a point s can be computed, after which a min-link path from s to t can be reported in time proportional to the link distance. The algorithm described in Section 28.4 computes the partition via “staged illumination,” essentially a form of the continuous Dijkstra method under the link distance metric.

In a polygonal domain with holes, min-link paths can also be computed using a staged illumination method, but the algorithm is not simple: it relies on efficient methods for computing a single face in an arrangement of line segments (see Chapter 23). A min-link s - t path can be computed in time $O(EVG\alpha^2(n) \log n)$, where $\alpha(n)$ is the inverse Ackermann function (Section 47.4). If we consider C -oriented and rectilinear link distance, then some better time/space bounds are possible, and some of these apply also to combined metrics, in which there is a cost for length as well as links.

Refer to Table 27.3.1 for many related results on link distance, including rectilinear link distance, and on two-point queries. See also Table 27.4.2 for more results specifically on simple polygons.

L_1 METRIC

Instead of measuring path length according to the L_2 (Euclidean) metric, consider the problem of computing shortest paths in a polygonal domain P that are short according to the L_1 metric.

A method based on visibility graph principles allows one to construct a sparse graph (with $O(n \log n)$ nodes and edges) that is **path-preserving** in that it is guaranteed to contain a shortest path between any two vertices. Applying Dijkstra’s algorithm then gives an $O(n \log^{1.5} n)$ time ($O(n \log n)$ space) algorithm for L_1 -shortest paths.

A method based on the continuous Dijkstra paradigm allows the SPM(s) to be constructed in time $O(n \log n)$, using $O(n)$ space [Mit92]. The special property of the L_1 metric that is exploited in this algorithm is the fact that the wavefront in this case is piecewise-linear, with wavelets that are line segments of slope ± 1 , so that the first vertex hit by a wavelet can be determined by rectangular range searching techniques (see Chapter 36).

Methods for finding L_1 -shortest paths generalize to the case of C -oriented paths, in which $c = |C|$ fixed directions are given. Shortest C -oriented paths

 TABLE 27.3.2 Shortest paths in other metrics.

PROBLEM	COMPLEXITY	NOTES	SOURCE
L_1 -shortest path, SPM(s)	$O(n \log n)$	polygonal domain	[Mit92, Mit89]
L_1 two-point query	$O(\log^2 n)$ query $O(n^2 \log n)$ space	polygonal domain	[CKT00]
L_1 two-point query	$O(n^2 \log^2 n)$ preproc $O(\log n)$ query	rectangle obstacles	[AC91, AC93]
L_1 two-point query	$O(n^2)$ space, preproc $O(\sqrt{n})$ query	rectangle obstacles	[EM94]
L_1 approx two-point query	$O(n^{1.5})$ space, preproc $O(\log n)$ query $O(n \log n)$ space	3-approx rectangle obstacles	[CK95b]
C -oriented shortest path two-point query	$O(n \log^2 n)$ preproc $O(cn \log n)$		[Mit92]
Weighted region problem	$O(c^2 \log^2 n)$ query $O(n^8 L)$		[CDK01]
Weighted region problem	$L = O(\log \frac{n NW}{\epsilon})$		[MP91]
Weighted region problem	$O(\frac{n}{\epsilon} \log \frac{1}{\epsilon} \log \frac{n}{\epsilon})$	($1+\epsilon$)-approx geometric parameters	[AMSO0, SR01]
L_1 weighted region problem	$O(n^2)$ $O(n \log^{3/2} n)$ preproc $O(\log n)$ query $O(n \log n)$ space	weights 0, 1, ∞ rectilinear regions single-source queries	[GMMN90] [CKT00]
L_1 WRP, two-point query	$O(\log^2 n)$ query $O(n^2 \log^2 n)$ space, preproc	rectilinear regions	[CKT00]
Bounded curvature path	$O(n^4 \log n)$	moderate obstacles	[BL96]
Sailor's problem ($L = 0$)	$O(n^2)$	polygonal domain	[Sel95]
Sailor's problem ($L > 0$)	$\text{poly}(n, \epsilon)$	ϵ -approx	[Sel95]
Max concealment	$O(v^2(v+n)^2)$	simple polygon	[GMMN90]
v viewpoints	$O(v^4 n^4)$	polygonal domain	[GMMN90]
Min total turn	$O(EVG \log n)$	polygonal domain	[AMP91]

can be computed in time $O(cn \log n)$. Since the Euclidean metric is approximated to within accuracy $O(1/c^2)$ if we use c equally spaced orientations, this results in an algorithm that computes, in time $O((n/\sqrt{\epsilon}) \log n)$, a path guaranteed to have length within a factor $(1+\epsilon)$ of the Euclidean shortest path length.

WEIGHTED REGION METRIC

The weighted region problem (WRP) seeks an optimal $s-t$ path according to the weighted region metric d_f induced by a given piecewise-constant weight function f . This problem is a natural generalization of the shortest-path problem in a polygonal domain: consider a weight function that assigns weight 1 to P and weight ∞ (or a sufficiently large constant) to the obstacles (the complement of P).

The weighted region problem models the minimum-time path problem for a point robot moving in a terrain of varied types (e.g., grassland, brushland, blacktop, bodies of water, etc.), where each type of terrain has an assigned weight equal to the reciprocal of the maximum speed of traversal for the robot.

Assume that f is specified by a triangulation having n vertices, with each face assigned an integer weight $\alpha \in \{0, 1, \dots, W, +\infty\}$. (We can allow each edge of the triangulation to have a weight that is possibly distinct from that of the triangular facets on either side of it; in this way, linear features such as roads can be modeled.) Using an algorithm based on the continuous Dijkstra method, one can find a path

whose weighted length is guaranteed to be within a factor $(1+\epsilon)$ of optimal, where $\epsilon > 0$ is any user-specified degree of precision [MP91]. The time complexity of the algorithm is $O(E \cdot S)$, where E is the number of “events” in the continuous Dijkstra algorithm, and S is the complexity of performing a numerical search to solve the following subproblem: Find a $(1+\epsilon)$ -shortest path from s to t that goes through a given sequence of k edges of the triangulation. It is known that $E = \Theta(n^4)$ in the worst case. The numerical search can be accomplished using a form of binary search that exploits the local optimality condition: An optimal path bends according to *Snell’s Law of Refraction* when crossing a region boundary. This leads to a bound of $S = O(k^2 \log(nNW/\epsilon))$ on the time needed to perform a search on a k -edge sequence, where N is the largest coordinate of any vertex of the triangulation (and all coordinates are integers). Since one can show that $k = O(n^2)$, this yields an overall time bound of $O(n^8L)$, where $L = \log(nNW/\epsilon)$ can be thought of as the bit complexity of the problem instance.

A simple and practical approach for computing an approximate solution is based on searching a discrete graph, such as an “edge subdivision graph” or a “pathnet” [LMS01, MM97], placing Steiner points judiciously on the edges (or, possibly interior to faces) of the input subdivision. In fact, using a logarithmic discretization (as in [Pap85]), with care in how Steiner points are placed near vertices [AMS00, SR01], provable approximation guarantees are obtained whose dependence on n is $O(n \log n)$, which compares favorably with the worst-case upper bounds for the algorithm of [MP91]. See [Table 27.3.2](#). It should be noted, though, that the dependence on $1/\epsilon$ is polynomial (vs. logarithmic) and that the “constants” in the big- O bounds reported conceal dependence on certain geometric parameters that may be unbounded in terms of ϵ and the combinatorial input size n .

Various special cases of the weighted region problem admit faster and simpler algorithms. For example, if the weighted subdivision is rectilinear, and path length is measured according to weighted L_1 length, then efficient algorithms for single-source and two-point queries can be based on searching a path-preserving graph [CKT00]. Similarly, if the region weights are restricted to $\{0, 1, \infty\}$ (while edges may have arbitrary (nonnegative) weights), then an $O(n^2)$ algorithm can be based on constructing a path-preserving graph similar to a visibility graph. This also leads to an efficient method for performing *lexicographic* optimization, in which one prioritizes various types of regions according to which is most important for path length minimization.

MINIMUM-TIME PATHS

The *kinodynamic motion planning problem* (also known as the *minimum-time path problem*) is a nonholonomic motion planning problem in which the objective is to compute a *trajectory* (a time-parameterized path, $(x(t), y(t))$) within a domain P that minimizes the total time necessary to move from an initial configuration (position and initial velocity) to a goal configuration (position and velocity), subject to bounds on the allowed acceleration and velocity along the path. The minimum-time path problem is a difficult optimal control problem; optimal paths will be complicated curves given by solutions to differential equations.

The bounds on acceleration and velocity are most often given by upper bounds on the L_∞ norm (the “decoupled case”) or the L_2 norm (the “coupled case”).

If there is an upper bound on the L_∞ norm of the velocity and acceleration vectors, one can obtain an *exact*, exponential-time, polynomial-space algorithm, based on characterizing a set of “canonical solutions” (related to “bang-bang” controls) that are guaranteed to include an optimal solution path. This leads to an expression in the first-order theory of the reals, which can be solved exactly; see [Chapter 33](#). However, it remains an open question whether or not a polynomial-time algorithm exists.

Donald et al. [DXCR93, DX95, RW00] developed approximation methods, including a polynomial-time algorithm that produces a trajectory requiring time at most $(1 + \epsilon)$ times optimal, for the decoupled case. Their approach is to discretize (uniformly) the four-dimensional phase space that represents position and velocity, with special care to ensure that the size of the grid is bounded by a polynomial in $1/\epsilon$ and n . Approximation algorithms for the coupled case are also known [DX95, RT94].

A closely related shortest-path problem is the ***bounded curvature shortest-path problem***, in which we require that no point of the path have a radius of curvature less than 1. For this problem, $(1+\epsilon)$ -approximation algorithms are known, with polynomial $(O(\frac{n^2}{\epsilon^2} \log n))$ running time [WA96]. The problem is known to be NP-hard in a polygonal domain [RW98]. For the special case in which the obstacles are “moderate” (have differentiable boundary curves, with radius of curvature at least 1), both an approximation algorithm and an exact $O(n^4 \log n)$ algorithm have been found [BL96].

OPTIMAL ROBOT MOTION

So far, we have considered only the problem of optimally moving a *point* robot. If the robot is modeled as a circle, or as a nonrotating polygon, then many of the results carry over by simply applying the standard ***configuration space*** approach in motion planning (see [Chapters 47 and 48](#)): “shrink” the robot to a (reference) point, and “grow” the obstacles (using a Minkowski sum) so that the complement of the grown obstacles models the region of the plane for which there is no collision with an obstacle if the robot has its reference point placed there.

Optimal motion of *rotating* noncircular robots is a much harder problem. Even the simplest case, of moving a (unit) line segment (a ***ladder***) in the plane, is highly nontrivial. One notion of optimal motion requires that we minimize the average distance traveled by a set of k fixed points, evenly distributed along the ladder. This “ d_k -distance” in fact defines a metric (for $k \geq 2$). The special case of $k = 2$ is the well-known ***Ulam’s problem***, for which optimal motions are fully characterized in the absence of obstacles [IRWY93]. The case of $k = \infty$ is an especially interesting case, requiring that we compute a minimum *work* motion of a ladder; however, no results are known for this problem. (The work measures the integral (over $\lambda \in [0, 1]$) of the path length, $L(\lambda)$, for each infinitesimal subsegment of length $d\lambda$.) While d_1 does not define a metric, several cases of d_1 -motion, and its generalization of measuring the distance traveled by any fixed “focus” F on the ladder, have been studied. In particular, if F is restricted to move on the visibility graph of a polygonal environment, polynomial-time algorithms are known. Without restrictions, minimizing the d_1 -distance (for any F not at an endpoint of the ladder) is NP-hard, but there exists an approximation algorithm [AKY96].

MULTIPLE CRITERION OPTIMAL PATHS

The standard shortest-path problem asks for paths that minimize some *one* objective (length) function. Frequently, however, an application requires us to find paths to minimize *two or more* objectives; the resulting problem is a **bicriterion** (or **multi-criterion**) shortest-path problem. A path is called **efficient** or **Pareto optimal** if no other path has a better value for one criterion without having a worse value for the other criterion.

Multi-criterion optimization problems tend to be hard. Even the bicriterion path problem in a graph is NP-hard: Does there exist a path from s to t whose length is less than L and whose weight is less than W ? Pseudo-polynomial-time algorithms are known, and many heuristics have been devised.

In geometric problems, various optimality criteria are of interest, including any pair from the following list: Euclidean (L_2) length, rectilinear (L_1) length, other L_p metrics, link distance, total turn, and so on. NP-hardness lower bounds are known for several versions [AMP91]. One problem of particular interest is to compute a Euclidean shortest path within a polygonal domain, constrained to have at most k links. No exact solution is currently known for this problem. Part of the difficulty is that a minimum-link path will not, in general, lie on the visibility graph (or on any simple discrete graph). Furthermore, the computation of the turn points of such an optimal path appears to require the solution to high-degree polynomials. A $(1 + \epsilon)$ -approximation to the shortest k -link path in a simple polygon P can be found in time $O(n^3 k^3 \log(Nk/\epsilon^{1/k}))$, where N is the largest integer coordinate of any vertex of P [MPA92]. In a *simple* polygon, one can always find an s - t path that simultaneously is within a factor 2 of optimal in link distance and within a factor $\sqrt{2}$ of optimal in Euclidean length; a corresponding result is not possible for polygons with holes. However, in $O(kE_{VG}^2)$ time, one can compute a path in a polygonal domain having at most $2k$ links and length at most that of a shortest k -link path.

In a rectilinear polygonal domain, efficient algorithms are known for the bicriterion path problem that combines *rectilinear* link distance and L_1 length [LYW96]. For example, efficient algorithms are known in two or more dimensions for computing optimal paths according to a *combined metric*, defined to be a linear combination of rectilinear link distance and L_1 path length [dBvKNO92]. (Note that this is not the same as computing the Pareto-optimal solutions.) Chen et al. [CDK97] give efficient algorithms for computing a shortest k -link rectilinear path, a minimum-link shortest rectilinear path, or any combined objective that uses a monotonic function of rectilinear link length and L_1 length in a rectilinear polygonal domain. Single-source queries can be answered in time $O(\log n)$, after $O(n \log^{3/2} n)$ preprocessing time to construct a data structure of size $O(n \log n)$; two-point queries can be answered in time $O(\log^2 n)$, using $O(n^2 \log^2 n)$ preprocessing time and space [CDK97].

OPEN PROBLEMS

1. Can a minimum-link path in a polygonal domain be computed in subquadratic time? The only lower bound known is $\Omega(n \log n)$.
2. What is the smallest size data structure for a simple polygon P that allows logarithmic-time two-point link distance queries?

3. For a polygonal domain (with holes), what is the complexity of computing a shortest k -link path between two given points?
 4. What is the complexity of the ladder problem for a polygonal domain, in which the cost of motion is the total work involved in translation/rotation?
 5. Is it NP-hard to minimize the d_1 -distance of a ladder endpoint?
 6. What is the complexity of the bounded curvature shortest-path problem in a simple polygon?
-
-

27.4 OTHER NETWORK OPTIMIZATION PROBLEMS

All of the problems considered so far involved computing a shortest path from one point to another (or from one point to all other points). We consider now some other network optimization problems, in which the objective is to compute a shortest path, cycle, tree, or other graph, subject to various constraints. A summary of results is given in [Table 27.4.1](#).

GLOSSARY

Minimum spanning tree (MST) of S : A tree of minimum total length whose nodes are a given set S of n points, and whose edges are line segments joining pairs of points.

Minimum Steiner spanning tree (Steiner tree) of S : A tree of minimum total length whose nodes are a superset of a given set S of n points, and whose edges are line segments joining pairs of points. Those nodes that are not points of S are called **Steiner points**.

k -minimum spanning tree (k -MST): A minimum-length tree that spans some subset of $k \leq n$ points of S .

Traveling salesman problem (TSP): Find a shortest cycle that visits every point of a set S of n points.

MAX TSP: Find a *longest* cycle that visits every point of a set S of n points.

Minimum latency tour problem: Find a tour on S that minimizes the sum of the “latencies,” where the latency of $p \in S$ is the length of the tour from the given depot to p . Also known as the **deliveryman problem** or the **traveling repairman problem**.

k -Traveling repairman problem: Find k tours covering S for k repairmen at a common depot, minimizing the total latency.

Min/max-area TSP: Find a cycle on a given set S of points such that the cycle defines a simple polygon of minimum/maximum area.

TSP with neighborhoods: Find a shortest cycle that visits at least one point in each of a set of neighborhoods (e.g., polygons), $\{P_1, P_2, \dots, P_k\}$.

Touring polygons problem: Find a shortest path/cycle that visits *in order* at least one point of each polygon in a sequence (P_1, P_2, \dots, P_k) .

Watchman route (path) problem: Find a shortest cycle (path) within a polygonal domain P such that every point of P is visible from some point of the cycle.

Lawnmowing problem: Find a shortest cycle (path) for the center of a disk (a “lawnmower” or “cutter”) such that every point of a given (possibly disconnected) region is covered by the disk at some position along the cycle (path).

Milling problem: Similar to the lawnmowing problem, but with the constraint that the cutter must at all times remain inside the given region (the “pocket” to be milled).

Zookeeper’s problem: Find a shortest cycle in a simple polygon P (the **zoo**) through a given vertex v such that the cycle visits every one of a set of k disjoint convex polygons (**cages**), each sharing an edge with P .

Aquarium-keeper’s problem: Find a shortest cycle in a simple polygon P (the **aquarium**) such that the cycle touches every edge of P .

Safari route problem: Find a shortest tour visiting a set of convex polygonal cages attached to the inside wall of a simple polygon P .

Relative convex hull of point set S within simple polygon P : The shortest cycle within P that surrounds S . The relative convex hull is necessarily a simple polygon, with vertices among the points of S and the vertices of P .

Monotone path problem: Find a shortest monotone path (if any) from s to t in a polygonal domain P . A polygonal path is **monotone** if there exists a direction vector d such that every directed edge of the path has a nonnegative inner product with d .

MINIMUM SPANNING TREES

The (Euclidean) minimum spanning tree problem can be solved to optimality in the plane in time $O(n \log n)$ by appealing to the fact that the MST is a subgraph of the Delaunay triangulation; see [Chapters 22](#) and [24](#). Efficient approximations in \mathbb{R}^d are based on spanners (Section 27.6).

The Steiner tree and k -MST problems, however, are NP-hard. Polynomial-time approximation schemes have been obtained, allowing one, for any fixed $\epsilon > 0$, to get within a factor $(1+\epsilon)$ of optimality [Aro98, Mit99], in fact in time $O(n \log n)$ in any fixed dimension [RS98].

TRAVELING SALESMAN PROBLEM

The traveling salesman problem is a classical problem in combinatorial optimization, and has been studied extensively in its geometric forms. The problem is NP-hard, but has a simple 2-approximation algorithm based on “doubling” the minimum spanning tree. The somewhat more involved Christofides heuristic yields a 1.5-approximation factor, which, until recently, was the best factor known. There is now a polynomial-time approximation scheme for geometric versions of the planar TSP, allowing one, for any fixed $\epsilon > 0$, to get within a factor $(1+\epsilon)$ of optimality [Aro98, Mit99], in fact in time $O(n \log n)$ in any fixed dimension [RS98]. This result is based on a generalization of the notion of t -spanners (Section 27.6)—the “ t -banyan”—which approximates to within factor t the interconnection cost (allowing Steiner points) for subsets of sites of *any* cardinality (not just 2 sites, as in the

 TABLE 27.4.1 Other optimal path/cycle/network problems.

PROBLEM	COMPLEXITY	NOTES	SOURCE
Minimum spanning tree (MST) in \mathbb{R}^d	$O(n \log n)$	exact, in \mathbb{R}^2	[PS85]
Steiner tree in \mathbb{R}^d	$O(n \log n)$	$(1+\epsilon)$ -approx, fixed d	[CK95a]
k -MST in \mathbb{R}^d	$O(n \log n)$	$(1+\epsilon)$ -approx, fixed d	[RS98]
Min-cost biconnected subgraph	$O(n \log n)$	$(1+\epsilon)$ -approx, fixed d	[RS98]
$(1+\epsilon)$ -approx			[CL00]
Traveling salesman (TSP) in \mathbb{R}^d	$O(n \log n)$	$(1+\epsilon)$ -approx, fixed d	[RS98]
MAX TSP	NP-hard in \mathbb{R}^3		[BFJ ⁺ 02]
	$O(n)$	$(1+\epsilon)$ -approx	[BFJ ⁺ 02]
	$O(n^{f-2} \log n)$	L_1, L_∞ in \mathbb{R}^2	[BFJ ⁺ 02]
Min-area TSP	NP-complete	f -facet polyhedral norm	[Fek00]
Max-area TSP	NP-complete		[Fek00]
TSP with neighborhoods	NP-hard		[MM95]
	$O(1)$ -approx	special regions	[AH94, DM01]
	$O(1)$ -approx	disjoint fat regions	[dBGK ⁺ 02]
	$(1+\epsilon)$ -approx	disjoint unit disks	[DM01]
Touring polygons problem	NP-hard	$(1+\epsilon)$ -approx	[DELM03]
	$O(nk^2 \log n)$	convex polygons	[DELM03]
	$O(nk \log(n/k))$	disjoint convex polygons	[DELM03]
Minimum latency problem	3.59-approx	poly time	[GK99]
	$(1+\epsilon)$ -approx	$n^{O(\log n/\epsilon^2)}$ time	[AK03]
k -Traveling repairman problem	8.497-approx		[FHR03]
Watchman route (fixed source)	$O(n^4 \log n)$	simple polygon	[DELM03]
	$O(n^3 \log n)$	simple polygon	[DELM03]
	$O(n)$	rectilinear simple polygon	[CN91]
Min-link watchman	NP-hard	polygonal domain	[CN88]
	NP-hard	$O(\log n)$ -approx	[AMP03]
	NP-hard	simple polygon	[AL93]
	$O(1)$ -approx	simple polygon	[AL95]
Lawnmowing problem	NP-hard	$O(1)$ -approx	[AFM00]
Milling problem	$O(1)$ -approx	simple polygon	[AFM00]
Simple $s-t$ Hamiltonian path	NP-hard, $O(1)$ -approx	polygonal domain	[AFM00]
	$O(n^2 m^2)$	m points in simple n -gon	[CCS00]
Aquarium-keeper's problem	NP-Complete	polygonal domain	[CCS00]
Zookeeper's problem	$O(n)$	simple polygon	[CEE ⁺ 91]
Relative convex hull	$O(n \log n)$	simple polygon	[Bes02]
Monotone path problem	$\Theta(n + k \log kn)$	k points in simple n -gon	[GH89]
	$O(n^3 \log n)$		[ACM89]

case of t -spanners). It is shown that for any fixed $\epsilon > 0$ and $d \geq 1$, there exists a $(1 + \epsilon)$ -banyan having $O(n)$ vertices and $O(n)$ edges, computable in $O(n \log n)$ time.

The **TSP-with-neighborhoods** problem arises when we require the tour/path to visit a set of regions, rather than a set of points. Constant-factor approximation algorithms are known for some special cases [AH94, dBGK⁺02, DM01], and an $O(\log n)$ -approximation algorithm is known for the general case in the plane.

A closely related problem is that of computing an optimal path for a lawnmower, modeled as, say, a circular cutter that must sweep out a region that covers a given domain of “grass.” This problem is NP-hard in general, but constant-factor approximation algorithms are known.

WATCHMAN ROUTE PROBLEM

Another problem closely related to the TSP is the watchman route problem, which can be thought of as a shortest-path/tour problem in which we have the constraint

that the path/tour must visit the visibility region associated with each point of the domain.

In the case of a simple polygonal domain, the watchman route problem has an $O(n^4 \log n)$ time algorithm to compute an exact solution and $O(n^3 \log n)$ is possible if we are given a point through which the tour must pass [DELM03]. In the case of a polygonal domain with holes, the problem is easily seen to be NP-hard (from Euclidean TSP), and the best approximation algorithm is one with factor $O(\log n)$, assuming rectilinear visibility.

OPEN PROBLEMS

1. Is the MAX TSP NP-hard in the Euclidean plane? What if the tour is required to be noncrossing?
2. Is there a PTAS for the minimum latency problem for points in fixed dimension?
3. Can one obtain a PTAS for the TSP with neighborhoods problem if the regions are disjoint? (Hardness of approximation is known for general regions [DBGK⁺02].) Is there an $O(1)$ -approximation if the neighborhoods are not connected sets (e.g., if the neighborhoods are pairs of points)?
4. Is the milling problem in simple polygons NP-hard?
5. Does the watchman route problem in a polygonal domain have an $O(1)$ -approximation algorithm? Is there a PTAS?

27.5 HIGHER DIMENSIONS

GLOSSARY

Polyhedral domain: A set $P \subset \mathbb{R}^3$ whose interior is connected and whose boundary consists of a union of a finite number of triangles. (The definition is readily extended to d dimensions, where the boundary must consist of a union of $(d-1)$ -simplices.) The complement of P consists of connected (polyhedral) components, which are the *obstacles*.

Orthohedral domain: A polyhedral domain having each boundary facet orthogonal to one of the coordinate axes.

Polyhedral surface: A connected union of triangles, with any two triangles intersecting in a common edge, a common vertex, or not at all, and such that every point in the relative interior of the surface has a neighborhood homeomorphic to a disk.

Edge sequence: The ordered list of obstacle edges that are intersected by a path.

COMPLEXITY

In three or more dimensions, most shortest-path problems become very difficult. In particular, there are two sources of complexity, even in the most basic Euclidean shortest-path problem in a polyhedral domain P .

One difficulty arises from algebraic considerations. In general, the structure of a shortest path in a polyhedral domain need not lie on any kind of discrete graph. Shortest paths in a polyhedral domain will be polygonal, with bend points that generally lie *interior* to obstacle edges, obeying a simple “unfolding” property: The path must enter and leave at the same angle to the edge. It follows that any locally optimal subpath joining two consecutive obstacle vertices can be “unfolded” at each edge along its edge sequence, thereby obtaining a straight segment. Given an edge sequence, this local optimality property uniquely identifies a shortest path through that edge sequence. However, to compare the lengths of two paths, each one shortest with respect to two (different) edge sequences, requires exponentially many bits, since the algebraic numbers that describe the optimal path lengths may have exponential degree.

A second difficulty arises from combinatorial considerations. The number of combinatorially distinct (i.e., having distinct edge sequences) shortest paths between two points may be exponential. This fact leads to a proof of the NP-hardness of the shortest-path problem [CR87], even if the obstacles are simply a set of parallel triangles.

Thus, it is natural to consider approximation algorithms for the general case, or to consider special cases for which polynomial bounds are achievable.

SPECIAL CASES

If the polyhedral domain P has only a small number k of convex obstacles, a shortest path can be found in $n^{O(k)}$ time. If the obstacles are known to be vertical “buildings” (prisms) having only k different heights, then shortest paths can be found in time $O(n^{6k-1})$, but it is not known if this version of the problem is NP-hard if k is allowed to be large.

If we require paths to stay on a polyhedral surface (i.e., the domain P is essentially 2D), then the unfolding property of optimal paths can be exploited to yield polynomial-time algorithms. The continuous Dijkstra paradigm leads to an algorithm requiring $O(n^2)$ time (and $O(n)$ space) algorithm to construct a shortest path map (or a geodesic Voronoi diagram), where n is the number of vertices of the surface [CH96, MMP87]. Kapoor [Kap99, O99] has recently announced an $O(n \log^2 n)$ time algorithm based on the continuous Dijkstra paradigm.

Several facts are known about the set of edge sequences corresponding to shortest paths on the surface of a *convex* polytope P in \mathbb{R}^3 . In particular, the worst-case number of distinct edge sequences that correspond to a shortest path between some pair of points is $\Theta(n^4)$, and the exact set of such sequences can be computed in time $O(n^6 \beta(n) \log n)$, where $\beta(n) = o(\log^* n)$ [AAOS97]. (A simpler $O(n^6)$ algorithm can compute a small superset of the sequences.) The number of **maximal** edge sequences for shortest paths is $\Theta(n^3)$. Some of these results depend on a careful study of the **star unfolding** with respect to a point p on the boundary, ∂P , of P . The star unfolding is the (nonoverlapping) cell complex obtained by subtracting from ∂P the shortest paths from p to the vertices of P , and then flattening the

resulting boundary.

Results on exact algorithms for special cases are summarized in [Table 27.5.1](#).

APPROXIMATION ALGORITHMS

Papadimitriou [Pap85] was the first to study the general problem from the point of view of approximations. He gave a fully polynomial approximation scheme that produces a path guaranteed to be no longer than $(1+\epsilon)$ times the length of a shortest path. His algorithm requires time $O(n^3(L + \log(n/\epsilon))^2/\epsilon)$, where L is the number of bits necessary to represent the value of an integer coordinate of a vertex of P . Clarkson [Cla87] also gives a fully polynomial approximation scheme, which improves upon that of Papadimitriou in the case that $n\epsilon^3$ is large.

Choi, Sellen, and Yap [CSY95, CSY97] have re-examined closely the analysis of Papadimitriou and have addressed some inconsistencies found in the original algorithm, drawing attention to the distinction between bit complexity and algebraic complexity. They have also introduced the notion of “precision-sensitivity” in algorithms, writing the complexity in terms of an implicit parameter, δ , that measures the implicit precision of the input instance [CSY95].

Har-Peled [HP98] shows how to compute an *approximate shortest path map* in polyhedral domains, computing, for fixed source s and $0 < \epsilon < 1$, a subdivision of size $O(n^2/\epsilon^{4+\delta})$ in time roughly $O(n^4/\epsilon^6)$, so that for any point $t \in \mathbb{R}^3$ a $(1+\epsilon)$ -approximation of the length of a shortest $s-t$ path can be reported in time $O(\log(n/\epsilon))$.

Considerable effort has been devoted to approximation algorithms for shortest paths on polyhedral surfaces. Given a convex polytope obstacle, Agarwal et al. [AHPSV97] show how to surround the polytope with a constant-size ($O(\epsilon^{-3/2})$, now improved to $O(\epsilon^{-5/4})$ [CLM03]) convex polytope having the property that shortest paths are approximately preserved (within factor $(1+\epsilon)$) on the outer polytope. This results in an approximation algorithm of time complexity $O(n \log(1/\epsilon) + f(\epsilon^{-5/4}))$, where $f(m)$ denotes the time complexity of solving exactly a shortest-path problem on an m -vertex convex surface (e.g., $f(m) = O(m^2)$ using [CH96], $f(m) = O(m \log^2 m)$ using [Kap99]). Har-Peled [HP99] gives an $O(n)$ -time algorithm to preprocess a convex polytope so that a two-point query can be answered in time $O((\log n)/\epsilon^{3/2} + 1/\epsilon^3)$, yielding the $(1+\epsilon)$ -approximate shortest path distance, as well as a path having $O(1/\epsilon^{3/2})$ segments that avoids the interior of the input polytope.

Varadarajan and Agarwal [VA99] obtained the first subquadratic-time algorithms for approximating shortest paths on general (nonconvex) polyhedral surfaces, computing a $7(1+\epsilon)$ -approximation in $O(n^{5/3} \log^{5/3} n)$ time, or a $15(1+\epsilon)$ -approximation in $O(n^{8/5} \log^{8/5} n)$ time. Their method is based on a partitioning of the surface into $O(n/r)$ patches, each having at most r faces, using a planar separator theorem. (The parameter r is chosen to be $n^{1/3} \log^{1/3} n$ or $n^{2/5} \log^{2/5} n$.) Then, on the boundary of each patch, a carefully selected set of points (“portals”) is selected, and these are interconnected with a graph that approximates shortest paths within each patch.

Practical approximation algorithms are based on searching a discrete graph (an “edge subdivision graph,” or a “pathnet”)[LMS01, MM97] by placing Steiner points judiciously on the edges (or, possibly interior to faces) of the input surface. This approach applies also to the case of weighted surfaces and weighted convex

 TABLE 27.5.1 Shortest paths in 3-space: exact algorithms.

OBSTACLES/DOMAIN	COMPLEXITY	NOTES	SOURCE
Polyhedral domain k convex polytopes	NP-hard $n^{O(k)}$	also for convex obstacles fixed k	[CR87] [Sha87]
Vertical buildings	$O(n^{6k-1})$	k different heights	[GNT89]
Axis-parallel boxes	$O(n^2 \log^3 n)$	L_1 metric	[CKV87]
Axis-parallel disjoint boxes	$O(n^2 \log n)$	L_1 metric	[CY95]
Axis-parallel boxes in \mathbb{R}^d	$O(n^d \log n)$ preproc $O(\log^{d-1} n)$ query $O((n \log n)^{d-1})$ space	monotonicity of paths in \mathbb{R}^d combined L_1 , link metric single-source queries	[dBvKNO92]
Polyhedral surface	$O(n \log^2 n)$ time	builds SPM(s), geodesic Voronoi	[Kap99]
Two-point query	$O((\sqrt{n}/m^{1/4}) \log n)$ query	convex polytope	[AAOS97]
Geodesic diameter	$O(n^6 m^{1+\delta})$ space, preproc $O(n^8 \log n)$	$1 \leq m \leq n^2$, $\delta > 0$ convex polytope	[AAOS97]

decompositions of \mathbb{R}^3 ; see the earlier discussion of the weighted region problem. One can obtain provable results on the approximation factor; see Table 27.5.2. It is worth noting, however, that these complexity bounds are under the assumption that certain geometric parameters are “constants”; these parameters may be unbounded in terms of ϵ and the combinatorial input size n .

OTHER METRICS

Link distance in a polyhedral domain in \mathbb{R}^d can be approximated (within factor 2) in polynomial time by searching a weak visibility graph whose nodes correspond to simplices in a simplicial decomposition of the domain. The complexity of computing the exact link distance is open.

For the case of orthohedral domains and rectilinear (L_1) shortest paths, the shortest-path problem in \mathbb{R}^d becomes relatively easy to solve in polynomial time, since the grid graph induced by the facets of the domain serves as a path-preserving graph that we can search for an optimal path. In \mathbb{R}^3 , we can do better than to use the $O(n^3)$ grid graph induced by $O(n)$ facets; an $O(n^2 \log^2 n)$ size subgraph suffices, which allows a shortest path to be found using Dijkstra’s algorithm in time $O(n^2 \log^3 n)$. More generally, in \mathbb{R}^d one can compute a data structure of size $O((n \log n)^{d-1})$, in $O(n^d \log n)$ preprocessing time, that supports fixed-source link distance queries in $O(\log^{d-1} n)$ time. In fact, this last result can be extended, within the same complexities, to the case of a combined metric, in which path cost is measured as a linear combination of L_1 length and rectilinear link distance.

For the special case of disjoint rectilinear box obstacles and rectilinear (L_1) shortest paths, a recent structural result may help in devising very efficient algorithms: There always exists a coordinate direction such that *every* shortest path from s to t is monotone in this direction [CY96]. In fact, this result has led to an $O(n^2 \log n)$ algorithm for the case $d = 3$.

TABLE 27.5.2 Shortest paths in 3-space: approximation algorithms.

OBSTACLES/DOMAIN	COMPLEXITY	NOTES	SOURCE
Polyhedral domain	$O(n^4(L + \log(\frac{n}{\epsilon}))^2/\epsilon^2)$	(1+ ϵ)-approx	[Pap85]
Polyhedral domain	$O(n^2 \text{polylog } n/\epsilon^4)$	(1+ ϵ)-approx	[Cla87]
	$O(\frac{n^2}{\epsilon^3} \log \frac{1}{\epsilon} \log n)$	(1+ ϵ)-approx	[AMS00]
Weighted polyhedral domain	$O(\frac{n}{\epsilon^3} \log \frac{1}{\epsilon} (\frac{1}{\sqrt{\epsilon}} + \log n))$ n convex faces	geometric parameters (1+ ϵ)-approx	[AMS00]
One convex obstacle	$O(\epsilon^{-5/4}\sqrt{n})$ expected	(1+ ϵ)-approx	[CLM03]
k convex polytopes	$O(n)$	$2k$ -approx	[HS98]
Convex polyhedral surface	$O(n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^3})$	(1+ ϵ)-approx	[AHPSV97]
Convex polyhedral surface	$O(\log \frac{n}{\epsilon})$ query	single-source queries	[HP98]
Convex polyhedral surface	$O(\frac{n}{\epsilon^3} \log \frac{1}{\epsilon} + \frac{n}{\epsilon^{1.5}} \log \frac{1}{\epsilon} \log n)$ preproc $O(\frac{1}{\epsilon^{1.5}} \log n + \frac{1}{\epsilon^3})$ query	$O(\frac{n}{\epsilon} \log \frac{1}{\epsilon})$ size SPM (1+ ϵ)-approx	[HP99]
Nonconvex polyhedral surface	$O(n)$ preproc $O(\log \frac{n}{\epsilon})$ query	two-point query single-source queries	[HP98]
Convex polyhedral surface	$O(n^2 \log n + \frac{n}{\epsilon} \log \frac{1}{\epsilon} \log \frac{n}{\epsilon})$ preproc $O(n + \frac{1}{\epsilon^6})$	$O(\frac{n}{\epsilon} \log \frac{1}{\epsilon})$ size SPM (1- ϵ)-approx diameter	[HP99]
Nonconvex polyhedral surface	$O(n^{5/3} \log^{5/3} n)$	7(1+ ϵ)-approx	[VA99]
Nonconvex polyhedral surface	$O(n^{8/5} \log^{8/5} n)$	15(1+ ϵ)-approx	[VA99]
Vertical buildings	$O(\frac{n}{\epsilon} \log \frac{1}{\epsilon} \log n)$	(1+ ϵ)-approx	[AMS00]
Min-link, polyhedral domain	$O(n^2)$ $\text{poly}(n)$	geometric parameters 1.1-approx 2-approx	[GNT89]

OPEN PROBLEMS

1. Can one compute shortest paths on a polyhedral surface in \mathbb{R}^3 in $O(n \log n)$ time using $O(n)$ space?
2. Can one compute a shortest path map for a polyhedral domain in output-sensitive time?
3. What is the complexity of the minimum-link path problem in 3-space?
4. What is the complexity of the shortest-path problem in 3-space for special cases of obstacles—e.g., disjoint axis-parallel boxes, unit spheres, etc.?

27.6 GEOMETRIC SPANNERS

GLOSSARY

Geometric graph: A graph $G = (V, E)$ together with an embedding in \mathbb{R}^d that maps vertices V to points and edges E to straight line segments. (See Chapter 10.)

Euclidean graph: A geometric graph with Euclidean lengths associated with the edges.

Complete geometric graph: A geometric graph $G = (V, E)$ whose edge set E joins each pair of points of V .

θ -Graph: A geometric graph in which each $v \in V$ is joined by an edge to a “closest” point $u \in V \cap C_i$, where each C_i is a **wedge** with apex v and angle at most θ .

Planar straight-line graph (PSLG): A geometric graph $G = (V, E)$ embedded in \mathbb{R}^2 with noncrossing edges.

t -Spanner: A subgraph $G' = (V, E')$ of a graph $G = (V, E)$ such that for any $u, v \in V$ the distance $\delta_{G'}(u, v)$ within G' is at most t times the distance $\delta_G(u, v)$ within G . We focus on **Euclidean t -spanners** for which the underlying graph G is the complete Euclidean graph in \mathbb{R}^d .

Planar t -spanner: A Euclidean t -spanner that is a PSLG in \mathbb{R}^2 .

Dilation, t^* , of a Euclidean graph $G = (V, E)$:

$$t^* = \max_{u, v \in V, u \neq v} \left\{ \frac{\delta_G(u, v)}{\delta_2(u, v)} \right\}$$

where $\delta_2(u, v)$ is the Euclidean distance between u and v . Thus, t^* is the smallest value of t for which G is a Euclidean t -spanner. The dilation is also known as the **stretch factor** or the **spanning ratio** of G .

Size of a Euclidean graph $G = (V, E)$: The number of edges, $|E|$.

Weight of a Euclidean graph $G = (V, E)$: The sum of the Euclidean lengths of all edges $e \in E$.

Degree of a graph $G = (V, E)$: The maximum number of edges incident on a common vertex $v \in V$.

k -Vertex Fault-Tolerant t -Spanner: A t -spanner with the property that the removal of any subset of at most k nodes, along with the incident edges, results in a subgraph that remains a t -spanner on the remaining set of points.

Well-separated pairs decomposition (WSPD) of a set $S \subset \mathbb{R}^d$ of points for a fixed separation constant $s > 0$: A set, $\{\{A_1, B_1\}, \{A_2, B_2\}, \dots, \{A_m, B_m\}\}$, of pairs of nonempty subsets of S such that (i) $A_i \cap B_i = \emptyset$, for each $i = 1, 2, \dots, m$; (ii) each pair of distinct elements $\{a, b\} \subset S$ has a unique pair $\{A_i, B_i\}$ with $a \in A_i$, $b \in B_i$; and (iii) A_i and B_i are **well-separated**. Sets X and Y are **well-separated** if there are two radius- r enclosing balls, $B_X \supset X$ and $B_Y \supset Y$, such that the distance between B_X and B_Y is at least sr . The **size** of the WSPD is m .

Fair-split tree T associated with a set $S \subset \mathbb{R}^d$: A binary tree, with each node ν having an associated subset $S(\nu) \subseteq S$ and the axis-parallel bounding box $R(\nu)$ of $S(\nu)$, such that (i) $|S(\nu)| = 1$ if ν is a leaf; and (ii) for each internal node ν , there exists a hyperplane orthogonal to the longest edge, ξ , of $R(\nu)$ separating the sets, $S(\nu_1)$ and $S(\nu_2)$, associated with the two children of ν , such that the hyperplane is at distance at least $|\xi|/3$ from each of the sides of $R(\nu)$ parallel to it.

t -SPANNERS

A natural greedy algorithm, similar to Kruskal’s minimum spanning tree algorithm, can be used to construct t -spanners:

Given an input geometric graph $G = (V, E)$ and a real number $t > 1$. Initialize edge set $E' \leftarrow \emptyset$. For each edge $(u, v) \in E$, considered in nondecreasing order of length $\delta_2(u, v)$, if $\delta_{G'}(u, v) > t \cdot \delta_2(u, v)$, then $E' \leftarrow E' \cup \{(u, v)\}$. Output the graph $G' = (V, E')$.

The greedy algorithm results in a t -spanner of size $O(n)$, weight $O(\log n) \cdot |MST|$, and degree $O(1)$, for any fixed dimension d and dilation $t > 1$ [ADD⁺93, CDNS95]. It can be applied also to general (nongeometric) graphs with weighted edges.

The θ -graph construction explicitly takes advantage of geometry and yields a t -spanner with dilation arbitrarily close to 1; specifically, $t = 1 + O(1/\theta)$, for sufficiently small θ [ADD⁺93, Kei88, Yao82].

Callahan and Kosaraju [CK95a] defined the notion of a well-separated pair decomposition (WSPD) and showed the remarkable theorem that a WSPD of size $O(n)$ can be constructed in time $O(n)$, given a fair split tree of an input set S of n points in \mathbb{R}^d , for any fixed dimension d and separation constant s . (More precisely, the size of the WSPD is $O(s^d n)$.) A fair split tree can be constructed using quadtree methods in time $O(n \log n)$ for any fixed dimension.

By selecting a representative edge from each pair in a WSPD, one obtains a t -spanner of size $O(n)$ with dilation that can be made arbitrarily close to 1, depending on the separation constant s . The WSPD has numerous other applications in approximation algorithms for geometric network optimization. One important application is to give a $(1 + \epsilon)$ -approximation algorithm, running in time $O(n \log n)$, for Euclidean minimum spanning trees in any fixed dimension d for any fixed $\epsilon > 0$.

One can in fact obtain t -spanners for n points in \mathbb{R}^d that are simultaneously good with respect to size, weight, and degree—size $O(n)$, weight $O(|MST|)$, and bounded degree (independent of the dimension d). Gudmundsson et al. [GLN02] show that such spanners can be computed in time $O(n \log n)$, improving the previous bound of $O(n \log^2 n)$ [DN97] and re-establishing the time bound claimed in Arya et al [ADM⁺95] (which was found to be flawed). $\Omega(n \log n)$ time is required for constructing *any* t -spanner for n points in \mathbb{R}^d in the algebraic decision tree model [CDS01].

Levcopoulos et al. [LNS02] showed that k -vertex fault-tolerant spanners of size $O(k^2 n)$ can be constructed in time $O(n \log n + k^2 n)$; alternatively, spanners of size $O(kn \log n)$ can be constructed in time $O(kn \log n)$. Lukovszki [Luk99] and recently Czumaj and Shao [CZ03] have shown how to obtain even smaller, degree-bounded low-weight k -vertex fault-tolerant spanners; degree $O(k)$ and weight $O(k^2 |MST|)$ can be obtained, and these bounds are asymptotically optimal.

The dilation (stretch factor) of a graph $G = (V, E)$ can be computed exactly in worst-case time $O(n^2 \log n + n|E|)$ using an all-pairs shortest path computation. Given a Euclidean graph with n vertices and m edges, its dilation (stretch factor) can be $(1 + \epsilon)$ -approximated in time $O(m + n \log n)$ [GLNS02a]. Narasimhan and Smid [NS02] have studied the *bottleneck stretch factor problem*, in which the goal is to be able to compute quickly, for any given $b > 0$, an approximate stretch factor of the *bottleneck graph* $G_b = (V, E_b)$ whose edge set E_b consists of those edges of the complete graph whose length is at most b . We say that t is a (c_1, c_2) -approximate

stretch factor of a graph if the true stretch factor, t^* , satisfies $t/c_1 \leq t^* \leq c_2 t$. A data structure of size $O(\log n)$ can be constructed that supports $O(\log \log n)$ -time queries, for any $b > 0$, yielding a (c_1, c_2) -approximate stretch factor of G_b . The construction of the data structure, which is based on a WSPD, is done using a randomized algorithm with expected running time that is slightly subquadratic.

Spanners can be computed for geodesic distances in a polygonal domain P : a $(1+\epsilon)$ -spanner of the visibility graph $\text{VG}(P)$ can be computed in time $O(n \log n)$, for any $\epsilon > 0$ [ACC⁺96]. Geometric spanners can be used to obtain very efficient approximate two-point shortest path distance queries. For any constant $t > 1$, a t -spanner G for n points in \mathbb{R}^d with m edges can be processed in time $O(m \log n)$, building a structure of size $O(n \log n)$, to support $(1+\epsilon)$ -approximate shortest path (in G) distance queries in $O(1)$ time between any two vertices of G . (A path can be reported in additional time proportional to the number of its edges.) Then, if the visibility graph $\text{VG}(P)$ is a t -spanner of the vertices of P , for some constant t , one obtains $O(1)$ -time (resp., $O(\log n)$ -time) $(1+\epsilon)$ -approximate shortest path distance queries between any two vertices (resp., points) of P . The assumption on $\text{VG}(P)$ holds if P has the “ t -rounded” property for some t : the shortest path distance between any pair of vertices is at most t times the Euclidean distance between them; such is the case if the obstacles are *fat*, as shown by Chew et al. [CDKK02].

PLANAR t -SPANNERS

For point sets in the plane it is natural to consider constructing *planar t -spanner* networks. One cannot hope, in general, to obtain planar t -spanners with t arbitrarily close to 1: four points at the corner of a square have no t -spanner with $t < \sqrt{2}$.

The first result on planar t -spanners is due to Chew [Che86], who showed that the Delaunay triangulation in the L_1 metric is a $\sqrt{10}$ -spanner for the complete Euclidean graph. (It is a $\sqrt{5}$ -spanner for the complete graph whose length are measured in the L_1 metric.) Chew [Che89] improved this result, showing that the Delaunay triangulation in the convex distance function based on an equilateral triangle is a planar graph with dilation at most 2. This is the current best dilation known for a planar t -spanner; the lower bound is $\sqrt{2}$, given by the example just mentioned.

The Euclidean Delaunay triangulation cannot, in general, yield a t -spanner with $t < \pi/2$, as shown by the example of placing points around a circle. The best known upper bound on the dilation, τ_{Del} , of the Euclidean Delaunay triangulation, is $\frac{4\sqrt{3}}{9}\pi \approx 2.42$ [KG92]. It is known that β -skeletons, for any $\beta > 0$, can have unbounded dilation [Epp00]; in particular, the Gabriel graph ($\beta = 1$) and the relative neighborhood graph ($\beta = 2$) are not t -spanners for any constant t . The minimum weight triangulation and the greedy triangulation (see Chapter 24) are t -spanners for constant t . This follows from a more general result of Das and Joseph [DJ89], who show that a PSLG is a t -spanner if it has the “**diamond property**” and the “**good polygon property**.” A **fat triangulation** of S , for which the aspect ratio (ratio of the length of the longest side to the corresponding height) of every triangle is at most α , is known to be a 2α -spanner [KG01].

One can compute planar t -spanners of low weight. In linear time, for any $r > 0$, a planar t -spanner, with $t = (1+1/r)\tau_{\text{Del}}$, of weight at most $(2r+1)|MST|$ can be computed from a Delaunay triangulation, where τ_{Del} is the dilation of the Delaunay

triangulation [LL92]. One can compute in time $O(n \log n)$ a planar t -spanner that is simultaneously low weight ($O(|MST|)$) and low degree (degree at most $14 + \lceil \frac{2\pi}{\alpha} \rceil$), where $t = \max\{\frac{\pi}{2}, \pi \sin \frac{\alpha}{2} + 1\} \cdot \tau_{Del}(1+\epsilon)$ and $0 < \alpha < \frac{\pi}{2}$. One can compute in time $O(n \log n)$ a planar t -spanner that is simultaneously low weight ($O(|MST|)$) and low degree (degree at most 27), with $t = (\pi+1)(1+\epsilon)\tau_{Del} \approx 10.02$ and any $\epsilon > 0$ [BGS02].

Planar t -spanners are also known for geodesic distances. A *conforming triangulation* for a polygonal domain P having triangles of aspect ratio at most α is a 2α -spanner for geodesic distances between vertices of P [KG01]. (A triangulation is *conforming* for P if all vertices of P is a vertex of the triangulation and each edge of P is the union of some edges of the triangulation.) The *constrained Delaunay triangulation* of P is a $\phi\pi$ -spanner [KG01].

OPEN PROBLEMS

1. What is the dilation of the Euclidean Delaunay triangulation? It is known to be between $\pi/2 \approx 1.57$ and $\frac{2\pi}{3\cos(\pi/6)} \approx 2.42$.
2. What is the minimum possible worst-case dilation for triangulations of point sets? It is known to be between $\sqrt{2}$ and 2. (For the L_1 or L_∞ metric, the tight bound on dilation is 2 [ACC⁺96].)

27.7 SOURCES AND RELATED MATERIAL

SURVEYS

Several other surveys offer a wealth of additional material and references:

- [AW88]: A survey of shortest paths and visibility graphs.
- [BE97]: A survey of approximation algorithms for geometric optimization problems.
- [Epp00]: A survey of results on spanning trees and t -spanners.
- [Lat91]: A book on motion planning algorithms.
- [LYW96]: A survey of rectilinear path problems.
- [Mit00]: Another survey on geometric shortest paths and network optimization.
- [NS]: A book on geometric spanners.
- [SW92]: A survey of topological network design problems.
- [Vaz01]: A book on approximation algorithms.

RELATED CHAPTERS

- Chapter 10: Geometric graph theory
 - Chapter 25: Triangulations
 - Chapter 26: Polygons
 - Chapter 28: Visibility
 - Chapter 47: Algorithmic motion planning
-

REFERENCES

- [AAOS97] P.K. Agarwal, B. Aronov, J. O'Rourke, and C. Schevon. Star unfolding of a polytope with applications. *SIAM J. Comput.*, 26:1689–1713, 1997.
- [AC91] M.J. Atallah and D.Z. Chen. Parallel rectilinear shortest paths with rectangular obstacles. *Comput. Geom. Theory Appl.*, 1:79–113, 1991.
- [AC93] M.J. Atallah and D.Z. Chen. On parallel rectilinear obstacle-avoiding paths. *Comput. Geom. Theory Appl.*, 3:307–313, 1993.
- [ACC⁺96] S. Arikati, D.Z. Chen, L.P. Chew, G. Das, M. Smid, and C. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In *Algorithms—ESA '96, 4th Annu. European Sympos.*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 514–528. Springer-Verlag, Berlin, 1996.
- [ACM89] E.M. Arkin, R. Connelly, and J.S.B. Mitchell. On monotone paths among obstacles, with applications to planning assemblies. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 334–343, 1989.
- [ADD⁺93] I. Althöfer, G. Das, D.P. Dobkin, D.A. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9:81–100, 1993.
- [ADM⁺95] S. Arya, G. Das, D.M. Mount, J. Salowe, and M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 489–498, 1995.
- [AFM00] E.M. Arkin, S. Fekete, and J.S.B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, 17:25–50, 2000.
- [AFW93] B. Aronov, S.J. Fortune, and G. Wilfong. Furthest-site geodesic Voronoi diagram. *Discrete Comput. Geom.*, 9:217–255, 1993.
- [AH94] E.M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Appl. Math.*, 55:197–218, 1994.
- [AHPSV97] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. Assoc. Comput. Mach.*, 44:567–584, 1997.
- [AK03] S. Arora and G. Karakostas. Approximation schemes for minimum latency problems. *SIAM J. Comput.*, 32: 1317–1337, 2003.
- [AKY96] Te. Asano, D.G. Kirkpatrick, and C.K. Yap. d_1 -optimal motion for a rod. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 252–263, 1996.
- [AL93] M.H. Alsuwaiyel and D.T. Lee. Minimal link visibility paths inside a simple polygon. *Comput. Geom. Theory Appl.*, 3:1–25, 1993.
- [AL95] M.H. Alsuwaiyel and D.T. Lee. Finding an approximate minimum-link visibility path inside a simple polygon. *Inform. Process. Lett.*, 55:75–79, 1995.

- [AMP91] E.M. Arkin, J.S.B. Mitchell, and C.D. Piatko. Bicriteria shortest path problems in the plane. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 153–156, 1991.
- [AMP03] E.M. Arkin, J.S.B. Mitchell, and C.D. Piatko. Minimum-link watchman tours. *Inform. Process. Lett.*, 86:203–207, 2003.
- [AMS95] E.M. Arkin, J.S.B. Mitchell, and S. Suri. Logarithmic-time link path queries in a simple polygon. *Internat. J. Comput. Geom. Appl.*, 5:369–395, 1995.
- [AMS00] L. Aleksandrov, A. Maheshwari, and J-R. Sack. Approximation algorithms for geometric shortest path problems. In *Proc. 32nd Annu. ACM Sympos. Theory Comput.*, pages 286–295, 2000.
- [AOS94] J. Adegeest, M.H. Overmars, and J. Snoeyink. Minimum-link c -oriented paths: Single-source queries. *Internat. J. Comput. Geom. Appl.*, 4:39–51, 1994.
- [Aro98] S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. Assoc. Comput. Mach.*, 45:753–782, 1998.
- [AW88] H. Alt and E. Welzl. Visibility graphs and obstacle-avoiding shortest paths. *Zeitschrift für Operations Research*, 32:145–164, 1988.
- [BE97] M. Bern and D. Eppstein. Approximation algorithms for geometric problems. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 296–345. PWS Publishing Company, Boston, 1997.
- [Bes02] S.N. Bespamyatnikh. An $O(n \log n)$ algorithm for the zoo-keeper’s problem. *Comput. Geom. Theory Appl.*, 24:63–74, 2002.
- [Bes03] S. Bespamyatnikh. Computing homotopic shortest paths in the plane. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 609–617, 2003.
- [BFJ⁺02] A. Barvinok, S. Fekete, D.S. Johnson, A. Tamir, G. Woeginger, and R. Woodrooffe. The geometric maximum traveling salesman problem. Manuscript, Technische Universität Braunschweig, 2002.
- [BGS02] P. Bose, J. Gudmundsson, and M. Smid. Constructing plane spanners of bounded degree and low weight. In *Proc. 10th Annu. European Sympos. Algorithms*, volume 2461 of *Lecture Notes Comput. Sci.*, pages 234–246. Springer-Verlag, Berlin, 2002.
- [BL96] J.-D. Boissonnat and S. Lazard. A polynomial-time algorithm for computing a shortest path of bounded curvature amidst moderate obstacles. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 242–251, 1996.
- [CCS00] Q. Cheng, M. Chrobak, and G. Sundaram. Computing simple paths among obstacles. *Comput. Geom. Theory Appl.*, 16:223–233, 2000.
- [CDK97] D.Z. Chen, O. Daescu, and K. Klenk. On geometric path query problems. In *Proc. 5th Workshop Algorithms Data Struct.*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 248–257. Springer-Verlag, Berlin, 1997.
- [CDK01] D.Z. Chen, O. Daescu, and K. Klenk. On geometric path query problems. *Internat. J. Comput. Geom. Appl.*, 11:617–645, 2001.
- [CDKK02] L.P. Chew, H. David, M.J. Katz, and K. Kedem. Walking around fat obstacles. *Inform. Process. Lett.*, 83:135–140, 2002.
- [CDNS95] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New sparseness results on graph spanners. *Internat. J. Comput. Geom. Appl.*, 5:125–144, 1995.
- [CDS01] D.Z. Chen, G. Das, and M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. *Discrete Appl. Math.*, 110:151–167, 2001.

- [CEE⁺91] J. Czyzowicz, P. Egyed, H. Everett, D. Rappaport, T.C. Shermer, D.L. Souvaine, G.T. Toussaint, and J. Urrutia. The aquarium keeper’s problem. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 459–464, January 1991.
- [CEG⁺94] B. Chazelle, H. Edelsbrunner, M. Grigni, L.J. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.
- [CH96] J. Chen and Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.
- [Che86] L.P. Chew. There is a planar graph almost as good as the complete graph. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 169–177, 1986.
- [Che89] L.P. Chew. There are planar graphs almost as good as the complete graph. *J. Comput. Syst. Sci.*, 39:205–219, 1989.
- [Che95] D.Z. Chen. On the all-pairs Euclidean short path problem. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 292–301, 1995.
- [CK95a] P.B. Callahan and S. Kosaraju. A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields. *J. Assoc. Comput. Mach.*, 42:67–90, 1995.
- [CK95b] D.Z. Chen and K. Klenk. Rectilinear short path queries among rectangular obstacles. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 169–174, 1995.
- [CKT00] D.Z. Chen, K. Klenk, and H-Y. Tu. Shortest path queries among weighted obstacles in the rectilinear plane. *SIAM J. Comput.*, 29:1223–1246, 2000.
- [CKV87] K.L. Clarkson, S. Kapoor, and P.M. Vaidya. Rectilinear shortest paths through polygonal obstacles in $O(n(\log n)^2)$ time. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 251–257, 1987.
- [CL00] A. Czumaj and A. Lingas. Fast approximation schemes for euclidean multi-connectivity problems. In *Internat. Colloq. Automata Lang. Program.*, volume 1853 of *Lecture Notes Comput. Sci.*, pages 856–868. Springer-Verlag, Berlin, 2000.
- [Cla87] K.L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. 19th Annu. ACM Sympos. Theory Comput.*, pages 56–65, 1987.
- [CLM03] B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 531–540, 2003.
- [CLMS02] S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 160–169, 2002.
- [CM99] Y.-J. Chiang and J.S.B. Mitchell. Two-point Euclidean shortest path queries in the plane. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 215–224, 1999.
- [CN88] W.-P. Chin and S. Ntafos. Optimum watchman routes. *Inform. Process. Lett.*, 28:39–44, 1988.
- [CN91] W.-P. Chin and S. Ntafos. Shortest watchman routes in simple polygons. *Discrete Comput. Geom.*, 6:9–31, 1991.
- [CR87] J.F. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.
- [CSY95] J. Choi, J. Sellen, and C.K. Yap. Precision-sensitive Euclidean shortest path in 3-space. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 350–359, 1995.

- [CSY97] J. Choi, J. Sellen, and C.K. Yap. Approximate Euclidean shortest paths in 3-space. *Internat. J. Comput. Geom. Appl.*, 7:271–295, 1997.
- [CT97] Y-J. Chiang and R. Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons inside a simple polygonal obstacle. *Internat. J. Comput. Geom. Appl.*, 7:85–121, 1997.
- [CY95] J. Choi and C.K. Yap. Rectilinear geodesics in 3-space. In *Proc. 11th Annu. ACM Symp. Comput. Geom.*, pages 380–389, 1995.
- [CY96] J. Choi and C.K. Yap. Monotonicity of rectilinear geodesics in d -space. In *Proc. 12th Annu. ACM Symp. Comput. Geom.*, pages 339–348, 1996.
- [CZ03] A. Czumaj and H. Zhao. Fault-tolerant geometric spanners. In *Proc. 19th Annu. ACM Symp. Comput. Geom.*, pages 1–10, 2003.
- [dB91] M. de Berg. On rectilinear link distance. *Comput. Geom. Theory Appl.*, 1:13–34, 1991.
- [dBGK⁺02] M. de Berg, J. Gudmundsson, M. Katz, C. Levcopoulos, M.H. Overmars, and A.F. van der Stappen. TSP with neighborhoods of varying size. In *Proc. 10th Annu. European Symp. Algorithms*, volume 2461 of *Lecture Notes Comput. Sci.*, pages 187–199. Springer-Verlag, Berlin, 2002.
- [dBvKNO92] M. de Berg, M. van Kreveld, B.J. Nilsson, and M.H. Overmars. Shortest path queries in rectilinear worlds. *Internat. J. Comput. Geom. Appl.*, 2:287–309, 1992.
- [DELM03] M. Dror, A. Efrat, A. Lubiwi, and J.S.B. Mitchell. Touring a sequence of polygons. *Proc. 35th Annu. ACM Symp. Theory Comput.*, pages 473–482, 2003.
- [DJ89] G. Das and D.A. Joseph. Which triangulations approximate the complete graph? In *Proc. Internat. Symp. Optimal Algorithms*, volume 401 of *Lecture Notes Comput. Sci.*, pages 168–192. Springer-Verlag, Berlin, 1989.
- [DM01] A. Dumitrescu and J.S.B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. In *Proc. 12th Symp. Discrete Algorithms*, pages 38–46, 2001.
- [DN91] G. Das and G. Narasimhan. Geometric searching and link distances. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes Comput. Sci.*, pages 261–272. Springer-Verlag, Berlin, 1991.
- [DN97] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. *Internat. J. Comput. Geom. Appl.*, 7:297–315, 1997.
- [DX95] B.R. Donald and P. Xavier. Provably good approximation algorithms for optimal kinodynamic planning for cartesian robots and open chain manipulators. *Algorithmica*, 14:480–530, 1995.
- [DXCR93] B.R. Donald, P. Xavier, J.F. Canny, and J.H. Reif. Kinodynamic motion planning. *J. Assoc. Comput. Mach.*, 40:1048–1066, 1993.
- [EKL02] A. Efrat, S.G. Kobourov, and A. Lubiwi. Computing homotopic shortest paths efficiently. In *Proc. 10th Annu. European Symp. Algorithms*, *Lecture Notes Comput. Sci.*, pages 411–423. Springer-Verlag, Berlin, 2002.
- [EM94] H. ElGindy and P. Mitra. Orthogonal shortest route queries among axis parallel rectangular obstacles. *Internat. J. Comput. Geom. Appl.*, 4:3–24, 1994.
- [Epp00] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier North-Holland, Amsterdam, 2000.

- [Fek00] S. Fekete. On simple polygonalizations with optimal area. *Discrete Comput. Geom.*, 23:73–110, 2000.
- [FHR03] J. Fakcharoenphol, C. Harrelson, and S. Rao. The k -traveling repairman problem. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 655–664, 2003.
- [FT87] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. Assoc. Comput. Mach.*, 34:596–615, 1987.
- [GH89] L.J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39:126–152, 1989.
- [GHL⁺87] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- [GK99] M.X. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. *Math. Prog.*, 82:111–124, 1999.
- [GLN02] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM J. Comput.*, 31:1479–1500, 2002.
- [GLNS02a] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid. Approximate distance oracles for geometric graphs. In *Proc. 13th ACM-SIAM Sympos. Discrete Algorithms*, pages 828–837, 2002.
- [GLNS02b] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid. Approximate distance oracles revisited. In *Proc. 13th Annu. Internat. Sympos. Alg. Comput.*, volume 2518 of *Lecture Notes Comput. Sci.*, pages 357–368. Springer-Verlag, Berlin, 2002.
- [GM91] S.K. Ghosh and D.M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.
- [GMMN90] L. Gewali, A. Meng, J.S.B. Mitchell, and S. Ntafos. Path planning in $0/1/\infty$ weighted regions with applications. *ORSA J. Comput.*, 2:253–272, 1990.
- [GNT89] L. Gewali, S. Ntafos, and I.G. Tollis. Path planning in the presence of vertical obstacles. Tech. Rep., Univ. Texas at Dallas, 1989.
- [GT93] M.T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths via balanced geodesic triangulations. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 318–327, 1993.
- [GT97] M.T. Goodrich and R. Tamassia. Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations. *J. Algorithms*, 23:51–73, 1997.
- [Her89] J. Hershberger. An optimal visibility graph algorithm for triangulated simple polygons. *Algorithmica*, 4:141–155, 1989.
- [Her95] J. Hershberger. Optimal parallel algorithms for triangulated simple polygons. *Internat. J. Comput. Geom. Appl.*, 5:145–170, 1995.
- [HP98] S. Har-Peled. Constructing approximate shortest path maps in three dimensions. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 383–391, 1998.
- [HP99] S. Har-Peled. Approximate shortest paths and geodesic diameters on convex polytopes in three dimensions. *Discrete Comput. Geom.*, 21:216–231, 1999.
- [HS94] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl.*, 4:63–98, 1994.
- [HS97] J. Hershberger and S. Suri. Matrix searching with the shortest path metric. *SIAM J. Comput.*, 26:1612–1634, 1997.

- [HS98] J. Hershberger and S. Suri. Practical methods for approximating shortest paths on a convex polytope in \mathbb{R}^3 . *Comput. Geom. Theory Appl.*, 10:31–46, 1998.
- [HS99] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28:2215–2256, 1999.
- [IRWY93] C. Icking, G. Rote, E. Welzl, and C.K. Yap. Shortest paths for line segments. *Algorithmica*, 10:182–200, 1993.
- [Kap99] S. Kapoor. Efficient computation of geodesic shortest paths. In *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 770–779, 1999.
- [Kei88] J.M. Keil. Approximating the complete Euclidean graph. In *Proc. 1st Scand. Workshop Algorithm Theory*, volume 318 of *Lecture Notes Comput. Sci.*, pages 208–213. Springer-Verlag, Berlin, 1988.
- [KG92] J.M. Keil and C. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete Comput. Geom.*, 7:13–28, 1992.
- [KG01] M. Karavelas and L.J. Guibas. Static and kinetic geometric spanners with applications. In *Proc. 12th Sympos. Discrete Algorithms*, pages 168–176, 2001.
- [KMM97] S. Kapoor, S.N. Maheshwari, and J.S.B. Mitchell. An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane. *Discrete Comput. Geom.*, 18:377–383, 1997.
- [Lat91] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.
- [LL92] C. Levcopoulos and A. Lingas. There are planar graphs almost as good as the complete graphs and almost as cheap as minimum spanning trees. *Algorithmica*, 8:251–256, 1992.
- [LMS01] M. Lanthier, A. Maheshwari, and J-R. Sack. Approximating shortest paths on weighted polyhedral surfaces. *Algorithmica*, 30:527–562, 2001.
- [LNS02] C. Levcopoulos, G. Narasimhan, and M. Smid. Improved algorithms for constructing fault-tolerant spanners. *Algorithmica*, 32:144–156, 2002.
- [LP84] D.T. Lee and F.P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.
- [Luk99] T. Lukovszki. New results on fault tolerant geometric spanners. In *Proc. 6th Workshop Algorithms Data Struct.*, volume 1663 of *Lecture Notes Comput. Sci.*, pages 193–204. Springer-Verlag, Berlin, 1999.
- [LYW96] D.T. Lee, C. Yang, and C. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Appl. Math.*, 70:185–215, 1996.
- [Mit89] J.S.B. Mitchell. An optimal algorithm for shortest rectilinear paths among obstacles. In *Abstracts 1st Canad. Conf. Comput. Geom.*, page 22, 1989.
- [Mit92] J.S.B. Mitchell. L_1 shortest paths among polygonal obstacles in the plane. *Algorithmica*, 8:55–88, 1992.
- [Mit96] J.S.B. Mitchell. Shortest paths among obstacles in the plane. *Internat. J. Comput. Geom. Appl.*, 6:309–332, 1996.
- [Mit99] J.S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k -MST, and related problems. *SIAM J. Comput.*, 28:1298–1309, 1999.
- [Mit00] J.S.B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier North-Holland, Amsterdam, 2000.

- [MM95] C. Mata and J.S.B. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1995.
- [MM97] C. Mata and J.S.B. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 1997.
- [MMP87] J.S.B. Mitchell, D.M. Mount, and C.H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.
- [MP91] J.S.B. Mitchell and C.H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *J. Assoc. Comput. Mach.*, 38:18–73, 1991.
- [MPA92] J.S.B. Mitchell, C.D. Piatko, and E.M. Arkin. Computing a shortest k -link path in a polygon. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 573–582, 1992.
- [MRW92] J.S.B. Mitchell, G. Rote, and G. Woeginger. Minimum-link paths among obstacles in the plane. *Algorithmica*, 8:431–459, 1992.
- [MSD00] A. Maheshwari, J-R. Sack, and H.N. Djidjev. Link distance problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 519–558. Elsevier North-Holland, Amsterdam, 2000.
- [NS02] G. Narasimhan and M. Smid. Approximation algorithms for the bottleneck stretch factor problem. *Nordic J. Computing*, 9:13–31, 2002.
- [NS] G. Narasimhan and M. Smid. Geometric spanner networks. Book Manuscript, Cambridge Univ. Press, to appear.
- [O99] J. O'Rourke. Computational geometry column 35. *Internat. J. Comput. Geom. Appl.*, 10:103–107, 2000. Also in *SIGACT News*, 30:31–32 (1999), Issue 111.
- [Pap85] C.H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Inform. Process. Lett.*, 20:259–263, 1985.
- [PL98] E. Papadopoulou and D.T. Lee. A new approach for the geodesic Voronoi diagram of points in a simple polygon and other restricted polygonal domains. *Algorithmica*, 20:319–352, 1998.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [PSR89] R. Pollack, M. Sharir, and G. Rote. Computing of the geodesic center of a simple polygon. *Discrete Comput. Geom.*, 4:611–626, 1989.
- [PV93] M. Pocchiola and G. Vegter. The visibility complex. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 328–337, 1993.
- [PV95] M. Pocchiola and G. Vegter. Computing the visibility graph via pseudo-triangulations. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 248–257, 1995.
- [RS98] S. Rao and W.D. Smith. Approximating geometrical graphs via “spanners” and “banyans.” In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 540–550, 1998.
- [RT94] J.H. Reif and S. Tate. Approximate kinodynamic planning using L_2 -norm dynamic bounds. *Comput. Math. Appl.*, 27:29–44, 1994.
- [RW98] J.H. Reif and H. Wang. The complexity of the two dimensional curvature-constrained shortest-path problem. In *Proc. 3rd Workshop Algorithmic Found. Robot.*, pages 49–57, 1998.

- [RW00] J.H. Reif and H. Wang. Non-uniform discretization for kinodynamic motion planning and its applications. *SIAM J. Comput.*, 30:161–190, 2000.
- [Sel95] J. Sellen. Direction weighted shortest path planning. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1970–1975, 1995.
- [Sha87] M. Sharir. On shortest paths amidst convex polyhedra. *SIAM J. Comput.*, 16:561–572, 1987.
- [SR01] Z. Sun and J.H. Reif. BUSHWHACK: An approximation algorithm for minimal paths through pseudo-Euclidean spaces. In *Proc. 12th Annu. Internat. Sympos. Algorithms Comput.*, volume 2223 of *Lecture Notes Comput. Sci.*, pages 160–171. Springer-Verlag, Berlin, 2001.
- [Sur86] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process.*, 35:99–110, 1986.
- [Sur90] S. Suri. On some link distance problems in a simple polygon. *IEEE Trans. Robot. Autom.*, 6:108–113, 1990.
- [SW92] J. Smith and P. Winter. Computational geometry and topological network design. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 287–385. World Scientific, Singapore, 1992.
- [VA99] K.R. Varadarajan and P.K. Agarwal. Approximating shortest paths on a nonconvex polyhedron. *SIAM J. Comput.*, 30:1321–1340, 1999.
- [Vaz01] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, 2001.
- [WA96] H. Wang and P.K. Agarwal. Approximation algorithms for curvature constrained shortest paths. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 409–418, 1996.
- [Yao82] A.C. Yao. On constructing minimum spanning trees in k -dimensional spaces and related problems. *SIAM J. Comput.*, 11:721–736, 1982.

28 VISIBILITY

Joseph O'Rourke

INTRODUCTION

In a geometric context, two objects are “visible” to each other if there is a line segment connecting them that does not cross any obstacles. Over 500 papers have been published on aspects of visibility in computational geometry in the last 25 years. The research can be broadly classified as primarily focused on combinatorial issues, or primarily focused on algorithms. We partition the combinatorial work into “art gallery theorems” (Section 28.1) and illumination of convex sets (28.2), and research on visibility graphs (28.3) and the algorithmic work into that concerned with polygons (28.4), more general planar environments (28.5), paths (28.6), and mirror reflections (28.7). All of this work concerns visibility in two dimensions. Investigations in three dimensions, both combinatorial and algorithmic, are discussed in Section 28.8, and the final section (28.9) touches on visibility in \mathbb{R}^d .

28.1 ART GALLERY THEOREMS

A typical “art gallery theorem” provides combinatorial bounds on the number of guards needed to visually cover a polygonal region P (the art gallery) defined by n vertices. Equivalently, one can imagine light bulbs instead of guards and require full direct-light illumination.

GLOSSARY

Guard: A point, a source of visibility or illumination.

Vertex guard: A guard at a polygon vertex.

Point guard: A guard at an arbitrary point.

Interior visibility: A guard $x \in P$ can see a point $y \in P$ if the segment xy is nowhere exterior to P : $xy \subset P$.

Exterior visibility: A guard x can see a point y outside of P if the segment xy is nowhere interior to P ; xy may intersect ∂P , the boundary of P .

Star polygon: A polygon visible from a single interior point.

Diagonal: A segment inside a polygon whose endpoints are vertices, and which otherwise does not touch ∂P .

Floodlight: A light that illuminates from the apex of a cone with aperture α .

Vertex floodlight: One whose apex is at a vertex (at most one per vertex).

MAIN RESULTS

The most general results obtained to date are summarized in Table 28.1.1. In all cases, the number of guards listed is the number that is necessary for some polygons, and sufficient for all polygons. Thus all bounds listed are tight.

TABLE 28.1.1 Number of guards needed.

PROBLEM NAME	POLYGONS	INT/EXT	GUARD	NUMBER
Art gallery theorem	simple	interior	vertex	$\lfloor n/3 \rfloor$
Fortress problem	simple	exterior	point	$\lceil n/3 \rceil$
Prison yard problem	simple	int & ext	vertex	$\lceil n/2 \rceil$
Prison yard problem	orthogonal	int & ext	vertex	$\lceil [5n/16], [5n/12] + 2 \rceil$
Orthogonal polygons	simple orthogonal	interior	vertex	$\lfloor n/4 \rfloor$
Orthogonal with holes	orthogonal with h holes	interior	vertex	$\lfloor n/4 \rfloor$
Polygons with holes	polygons with h holes	interior	point	$\lfloor (n + h)/3 \rfloor$

Of special note is the difficult *orthogonal prison yard problem*: How many vertex guards are needed to cover both the interior and the exterior of an orthogonal polygon? See Figure 28.1.1. The lower and upper bounds listed in the table were obtained by [HK96] via this new graph-coloring theorem: Every plane, bipartite, 2-connected graph has an *even triangulation* (all nodes have even degree) and therefore the resulting graph is 3-colorable.

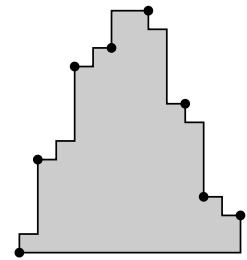


FIGURE 28.1.1

A pyramid polygon with $n = 24$ vertices whose interior and exterior are covered by 8 guards. Repeating the pattern establishes a lower bound of $5n/16 + c$ on the orthogonal prison yard problem [HK93].

COVERS AND PARTITIONS

Each art gallery theorem above implies a cover result, a cover by star polygons. Many of the theorem proofs rely on certain partitions. For example, the orthogonal polygon result depends on the theorem that every orthogonal polygon may be partitioned via diagonals into convex quadrilaterals. See [Section 26.2](#) for more on covers and partitions.

EDGE GUARDS

A variation permits guards (*mobile guards*) to patrol segments, diagonals, or edges; equivalent is illumination by line segment/diagonal/edge light sources (fluorescent light bulbs). Here there are fewer results; see Table 28.1.2. Toussaint conjectures that the last line of this table should be $\lfloor n/4 \rfloor$ for n sufficiently large.

TABLE 28.1.2 Edge guards.

POLYGONS	GUARD	BOUNDS	SOURCE
Polygon	diagonal	$\lfloor n/4 \rfloor$	[O'R83]
Orthogonal polygons	segment	$\lfloor (3n+4)/16 \rfloor$	[Agg84, O'R87]
Orthogonal polygons with h holes	segment	$\lfloor (3n+4h+4)/16 \rfloor$	[GHKS96]
Polygon ($n > 11$)	edge	$[\lfloor n/4 \rfloor, \lfloor 3n/10 \rfloor]$	[She94].

28.1.1 FLOODLIGHT ILLUMINATION

Urrutia introduced a class of questions involving guards with restricted vision, or, equivalently, illumination by floodlights: How many floodlights, each with aperture α , and with their apices at distinct nonexterior points, are sufficient to cover any polygon of n vertices? One surprise is that $\lfloor n/3 \rfloor$ half-guards/ π -floodlights suffice, although not when restricted to vertices. A second surprise is that, for any $\alpha < \pi$, there is a polygon that cannot be illuminated by an α floodlight at every vertex. See Table 28.1.3. A third surprise is that the best result on vertex π -floodlights employs pointed pseudotriangulations (cf. [Chapter 5](#)) in an essential way.

TABLE 28.1.3 Floodlights.

APEX	ALPHA	BOUNDS	SOURCE
Any point	$[180^\circ, 360^\circ]$	$\lfloor n/3 \rfloor$	[Tót00]
Any point	$[90^\circ, 180^\circ]$	$2\lfloor n/3 \rfloor$	[Tót00]
Any point	$[45^\circ, 60^\circ)$	$[n-2, n-1]$	[Tót03a]
Vertex	$< 180^\circ$	not always possible	[ECOUX95]
Vertex	180°	$[9n/14 - c, \lfloor 2n/3 \rfloor - 1]$	[ST03]

28.2 ILLUMINATION OF PLANAR CONVEX SETS

A natural extension of exterior visibility is illumination of the plane in the presence of obstacles. Here it is natural to use “illumination” in the same sense as “visibility.” Under this model, results depend on whether light sources are permitted to lie on obstacle boundaries: $\lfloor 2n/3 \rfloor$ lights are necessary and sufficient (for $n > 5$) if they may [O’R87], and $\lfloor 2(n+1)/3 \rfloor$ if they may not [Tót02]. More work has been done on illuminating the boundary of the obstacles, under a stronger notion of illumination, corresponding to “clear visibility.”

GLOSSARY

Illuminate: x illuminates y if xy does not include a point strictly interior to an obstacle, and does not cross a segment obstacle.

Cross: xy crosses segment s if they have exactly one point p in common, and p is in the relative interior of both xy and s .

Clearly illuminate: x clearly illuminates y if the open segment (x, y) does not include any point of an obstacle.

Compact: Bounded.

Homothetic: Similar and in parallel position.

Isothetic: Sides parallel to the coordinate axes.

MAIN RESULTS

A third, even stronger notion of illumination is considered in Section 28.9 below. The main question that has been investigated is: How many point lights strictly exterior to a collection of n pairwise disjoint compact, convex objects in the plane are needed to clearly illuminate every object boundary point? Answers for a variety of restricted sets are shown in Table 28.2.1.

TABLE 28.2.1 Illuminating convex sets in plane.

FAMILY	BOUNDS	SOURCE
Convex sets	$4n - 7$	[Fej77]
Circular disks	$2n - 2$	[Fej77]
Isothetic rectangles	$[n - 1, n + 1]$	[Urr00]
Homothetic triangles	$[n, n + 1]$	[CRCU93]
Triangles	$[n, \lfloor (5n + 1)/4 \rfloor]$	[Tót01b]
Segments (one side)	$[4n/9 - 2, \lfloor (n + 1)/2 \rfloor]$	[Tót03b]
Segments (both sides)	$[4(n + 1)/5]$	[Tót01a]

The most interesting open problem here is to close the gap for triangles. Urrutia conjectures [Urr00] that $n + c$ lights suffice for some constant c .

28.3 VISIBILITY GRAPHS

Whereas art gallery theorems seek to encapsulate an environment’s visibility into one function of n , the study of visibility graphs endeavors to uncover the more fine-grained structure of visibility. The original impetus for their investigation came from pattern recognition, and its connection to shape continues to be one of its primary sources of motivation; see [Chapter 51](#). Another application is graphics ([Chapter 49](#)): illumination and radiosity depend on 3D visibility relations (Section 28.8.)

GLOSSARY

Visibility graph: A graph with nodes for each object, and arcs between objects that can see one another.

Vertex visibility graph: The objects are the vertices of a simple polygon.

Endpoint visibility graph: The objects are the endpoints of line segments in the plane. See [Figure 28.3.1b](#).

Segment visibility graph: The objects are whole line segments in the plane, either open or closed.

Object visibility: Two objects A and B are visible to one another if there are points $x \in A$ and $y \in B$ such that x sees y .

Point visibility: Two points x and y can see one another if the segment xy is not “obstructed,” where the meaning of “obstruction” depends on the problem.

ϵ -visibility: Lines of sight are finite-width beams of visibility.

Hamiltonian: A graph is Hamiltonian if there is a simple cycle that includes every node.

OBSTRUCTIONS TO VISIBILITY

For polygon vertices, x sees y if xy is nowhere exterior to the polygon, just as in art gallery visibility; this implies that polygon edges are part of the visibility graph. For segment endpoints, x sees y if the closed segment xy intersects the union of all the segments either in just the two endpoints, or in the entire closed segment. This disallows grazing contact with a segment, but includes the segments themselves in the graph.

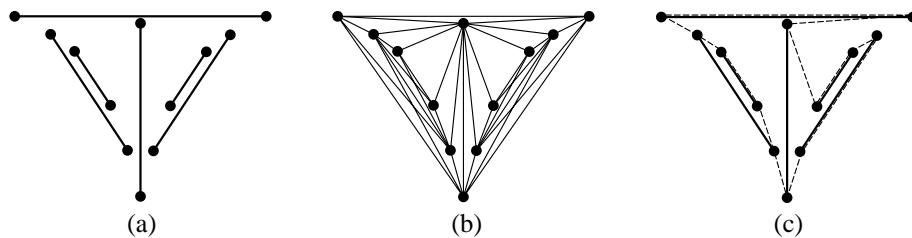
GOALS

Four goals can be discerned in research on visibility graphs:

1. Characterization: asks for a precise delimiting of the class of graphs realizable by a certain class of geometric objects.

FIGURE 28.3.1

(a) A set of segments. (b) Their endpoint visibility graph G . (c) A Hamiltonian cycle in G .



2. Recognition: asks for an algorithm to recognize when a graph is a visibility graph.
3. Reconstruction: asks for an algorithm that will take a visibility graph as input, and output a geometric realization.
4. Counting: concerned with the number of visibility graphs under various restrictions [HN01].

VERTEX VISIBILITY GRAPHS

A complete characterization of polygon vertex visibility graphs has remained elusive, but progress has been made by:

1. Restricting the class of polygons: polynomial-time recognition and reconstruction algorithms for orthogonal staircase polygons have been obtained. See Figure 28.3.2.
2. Restricting the class of graphs: every 3-connected vertex visibility graph has a 3-clique ordering, i.e., an ordering of the vertices so that each vertex is part of a triangle composed of preceding vertices.
3. Adding information: assuming knowledge of the boundary Hamiltonian circuit, four necessary conditions have been established by Ghosh and others [Gho97], and conjectured to be sufficient.

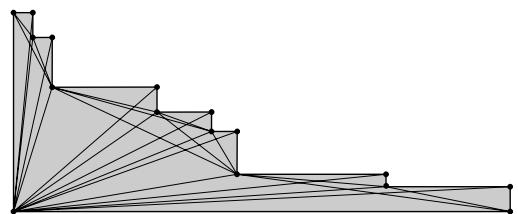


FIGURE 28.3.2

A staircase polygon and its vertex visibility graph.

ENDPOINT VISIBILITY GRAPHS

For segment endpoint visibility graphs, there have been two foci:

1. Are the graphs Hamiltonian? See [Figure 28.3.1c](#). Posed by Mirzaian, this has recently been settled via a complex proof [HT01]: YES, there is always a Hamiltonian polygon (i.e., a noncrossing circuit).
2. Size questions: there must be at least $5n - 4$ edges [SE87], and at least $6n - 6$ when no segment is a “chord” splitting the convex hull [GOH⁺02]; the smallest clique cover has size $\Omega(n^2 / \log^2 n)$ [AAAS94].

SEGMENT VISIBILITY GRAPHS

Whole segment visibility graphs have been investigated most thoroughly under the restriction that the segments are all (say) vertical and visibility is horizontal. Such segments are often called *bars*. The visibility is usually required to be ϵ -visibility. Endpoints on the same horizontal often play an important role here, as does the distinction between closed segments and intervals (which may or may not include their endpoints). There are several characterizations:

1. G is representable by segments, with no two endpoints on the same horizontal, iff there is a planar embedding of G such that, for every interior k -face F , the induced subgraph of F has exactly $2k - 3$ edges.
2. G is representable by segments, with endpoints on the same horizontal permitted, iff there is a planar embedding of G with all cutpoints on the exterior face.
3. Every 3-connected planar graph is representable by intervals.

OTHER VISIBILITY GRAPHS

The notion of a visibility graph can be extended to objects such as disjoint disks: each disk is a node, with an arc if there is a segment connecting them that avoids touching any other disk. Rappaport proved that the visibility graph of disjoint congruent disks is Hamiltonian [R03]. *Rectangle visibility graphs*, which restrict visibility to vertical or horizontal lines of sight between disjoint rectangles, have been studied for their role in graph drawing ([Chapter 52](#)). A typical result is that any graph with a maximum vertex degree of 4 can be realized as a rectangle visibility graph [BDHS97].

OPEN PROBLEMS

1. Given a visibility graph G and a Hamiltonian circuit C , construct in polynomial time a simple polygon such that its vertex visibility graph is G , with C corresponding to the polygon’s boundary.

-
2. Develop an algorithm to recognize whether a polygon vertex visibility graph is planar. Necessary and sufficient conditions are known [LC94].
-

28.4 ALGORITHMS FOR VISIBILITY IN A POLYGON

Designing algorithms to compute aspects of visibility in a polygon P was a major focus of the computational geometry community in the 1980s. For most of the basic problems, optimal algorithms were found, several depending on Chazelle's linear-time triangulation algorithm [Cha91].

GLOSSARY

Throughout, P is a polygon.

Kernel: The set of points in P that can see all of P . See Figure 33.4.4.

Point visibility polygon: The region visible from a point in P .

Segment visibility polygon: The region visible from a segment in P .

MAIN RESULTS

The main algorithms are listed in Table 28.4.1. We discuss two of these algorithms below to illustrate their flavor.

TABLE 28.4.1 Polygon visibility algorithms.

ALGORITHM TO COMPUTE	TIME COMPLEXITY	SOURCE
Kernel	$O(n)$	[LP79]
Point visibility polygon	$O(n)$	[JS87]
Segment visibility polygon	$O(n)$	[GHL ⁺ 87]
Shortest illuminating segment	$O(n)$	[DN94]
Vertex visibility graph	$O(E)$	[Her89]

VISIBILITY POLYGON ALGORITHM

Let $x \in P$ be the visibility source. Lee's linear-time algorithm [JS87] processes the vertices of P in a single counterclockwise boundary traversal. At each step, a vertex is either pushed on or popped off a stack, or a *wait* event is processed. The latter occurs when the boundary at that point is invisible from x . At any stage, the stack represents the visible portion of the boundary processed so far.

Although this algorithm is elementary in its tools, it has proved delicate to implement correctly.

VISIBILITY GRAPH ALGORITHM

In contrast, Hershberger's vertex visibility algorithm [Her89] uses sophisticated tools to achieve output-size sensitive time complexity $O(E)$, where E is the number of edges of the graph. His algorithm exploits the intimate connection between shortest paths and visibility in polygons. It first computes the *shortest path map* (Chapter 27) in $O(n)$ time for a vertex, and then systematically transforms this into the map of an adjacent vertex in time proportional to the number of changes. Repeating this achieves $O(E)$ time overall.

Most of the above algorithms have been parallelized; see, for example, [GSG92].

28.5 ALGORITHMS FOR VISIBILITY AMONG OBSTACLES

The shortest path between two points in an environment of polygonal obstacles follows lines of sight between obstacle vertices. This has provided an impetus for developing efficient algorithms for constructing visibility regions and graphs in such settings. The obstacles most studied are noncrossing line segments, which can be joined end-to-end to form polygonal obstacles. Many of the questions mentioned in the previous section can be revisited for this environment.

The major results are shown in Table 28.5.1; the first three are described in [O'R87]; the fourth is discussed below.

TABLE 28.5.1 Algorithms for visibility among obstacles.

ALGORITHM TO COMPUTE	TIME COMPLEXITY
Point visibility region	$O(n \log n)$
Segment visibility region	$\Theta(n^4)$
Endpoint visibility graph	$O(n^2)$
Endpoint visibility graph	$O(n \log n + E)$

ENDPOINT VISIBILITY GRAPH

The largest effort has concentrated on constructing the endpoint visibility graph. Worst-case optimal algorithms were first discovered by constructing the line arrangement dual to the endpoints in $O(n^2)$ time. Since many visibility graphs have less than a quadratic number of edges, an output-size sensitive algorithm was a significant improvement: $O(n \log n + E)$ where E is the number of edges of the graph [GM91].

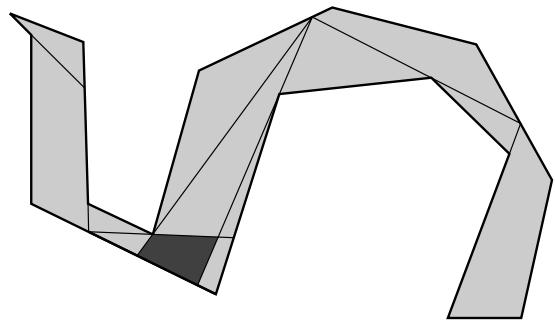


FIGURE 28.6.1

The link center is shown darkly shaded; every point in the polygon can be reached with no more than three links from a point in the center. Several key visibility chords are drawn.

28.6 VISIBILITY PATHS

A fruitful idea was introduced to visibility research in the mid-1980s: the notion of “link distance” between two points, which represents the smallest number of mutually visible relay stations needed to communicate from one point to another (Sections 26.4 and 27.3). A related notion called “watchman tours” was introduced a bit later, mixing shortest paths and visibility problems, and employing many of the concepts developed for link-path problems (Section 26.4).

GLOSSARY

Link: A segment.

Link distance: The smallest number of links in a polygonal path connecting the points.

Link diameter of P : The largest link distance between any two points in P .

Link center of P : The collection of points whose maximal link distance to any point of P is as small as possible. See Figure 28.6.1.

Shortest watchman tour in P : A shortest closed path π in a polygon P such that every point of P is visible from some point of π .

MAIN RESULTS

The main results for link centers are shown in Table 28.6.1. See [Tables 27.4.2](#) and [27.3.1](#) and the related sections for further results.

TABLE 28.6.1 Algorithms for link centers.

LINK CENTER WITHIN	TIME COMPLEXITY	SOURCE
Polygon	$O(n \log n)$	[DLS92]
Orthogonal polygon	$O(n)$	[NS91]
Polygon with holes	NP-hard	[AL93]

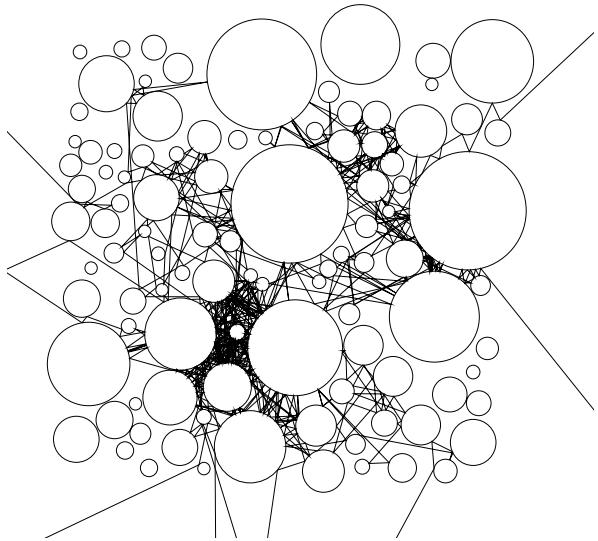


FIGURE 28.7.1

100 mirror disks fail to trap 10 rays from a point source (near the center) [OP01].

28.7 MIRROR REFLECTIONS

GLOSSARY

Light ray reflection: A light ray reflects from an interior point of a mirror with reflected angle equal to incident angle; a ray that hits a mirror endpoint is absorbed.

Mirror polygon: A polygon all of whose edges are mirrors reflecting light rays.

Periodic light ray: A ray that reflects from a collection of mirrors and, after a finite number of reflections, rejoins its path (and thenceforth repeats that path).

Trapped light ray: One that reflects forever, and so never “reaches” infinity.

Klee asked whether every polygonal room whose walls are mirrors (a mirror polygon) is illuminable from every interior point [Kle69, KW91]. Tokarsky answered NO by constructing rooms that leave one point dark when the light source is located at a particular spot [Tok95]. However, a second question of Klee remains open: Is every mirror polygon illuminable from some interior point?

The behavior of light reflecting in a polygon is complex. Aronov et al. [ADD⁺98] proved that after k reflections, the boundary of the illuminated region has combinatorial complexity $O(n^{2k})$, with a matching lower bound for any fixed k . Even determining whether every triangle supports a periodic ray is unresolved; see [HH00].

Pach asked whether a finite set of disjoint circular mirrors can trap all the rays from a point light source [Ste96]. See Fig. 28.7.1. This and many other related questions [OP01] remain open.

28.8 VISIBILITY IN THREE DIMENSIONS

Research on visibility in three dimensions (3D) has concentrated on three topics: hidden surface removal, polyhedral terrains, and various 3D visibility graphs.

28.8.1 HIDDEN SURFACE REMOVAL

“Hidden surface removal” is one of the key problems in computer graphics ([Chapter 49](#)), and has been the focus of intense research for two decades. The typical problem instance is a collection of (planar) polygons in space, from which the view from $z = \infty$ must be constructed. Traditionally, hidden-surface algorithms have been classified as either *image-space* algorithms, exploiting the ultimate need to compute visible colors for image pixels, and *object-space* algorithms, which perform exact computations on object polygons. We only discuss the latter.

The complexity of the output scene can be quadratic in the number of input vertices n . A worst-case optimal $\Theta(n^2)$ algorithm can be achieved by projecting the lines containing each polygon edge to a plane and constructing the resulting arrangement of lines [Dév86, McK87]. Most recent work has focused on obtaining output-size sensitive algorithms, whose time complexity depends on the number of vertices k in the output scene (the complexity of the *visibility map*), which is often less than quadratic in n . See Table 28.8.1 for selected results. In the table, k is the complexity of the *visibility map*, the “wire-frame” projection of the scene. A notable example is based on careful construction of “visibility maps,” which leads, e.g., to a complexity of $O((n + k) \log^2 n)$ for performing hidden surface removal on nonintersecting spheres, where k is the complexity of the output map.

TABLE 28.8.1 Hidden-surface algorithm complexities.

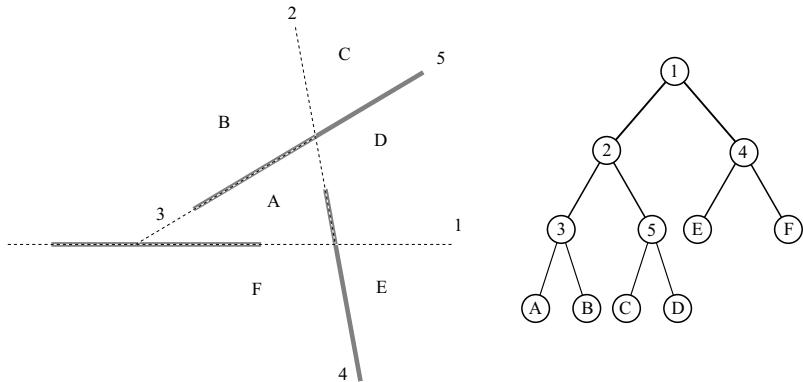
ENVIRONMENT	COMPLEXITY	SOURCE
Isothetic rectangles	$O((n + k) \log n)$	[dBO92]
Polyhedral terrain	$O((n + k) \log n \log \log n)$	[RS88]
Nonintersecting polyhedra	$O(n\sqrt{k} \log n)$ $O(n^{1+\epsilon}\sqrt{k})$ $O(n^{2/3+\epsilon}k^{2/3} + n^{1+\epsilon})$	[SO92] [dBHO ⁺ 94] [AM93]
Arbitrary intersecting spheres	$O(n^{2+\epsilon})$	[AS00]
Nonintersecting spheres	$O(k + n^{3/2} \log n)$	[SO92]
Restricted-intersecting spheres	$O((n + k) \log^2 n)$	[KOS92]

28.8.2 BINARY SPACE PARTITION TREES

Binary Space Partition (BSP) trees have become a popular method of implementing the basic *painter’s algorithm*, which displays objects back-to-front to obtain proper occlusion of front-most surfaces. A *BSP* partitions \mathbb{R}^d into empty, open

FIGURE 28.8.1

A binary space partition tree for 3 segments.



convex sets by hyperplanes in a recursive fashion. A BSP for a set S of n line segments in \mathbb{R}^2 is a partition such that all the open regions corresponding to leaf nodes of the tree are empty of points from S : all the segments in S lie along the boundaries of the regions. An example is shown in Fig. 28.8.1. In general, a BSP for S will “cut up” the segments in S , in the sense that a particular $s \in S$ will not lie in the boundary of a single leaf region. In the figure, partitions 1 and 2 both cut segments, but partition 3 does not.

An attractive feature of BSPs is that an implementation to construct them is easy: In \mathbb{R}^3 , select a polygon, partition all objects by the plane containing it, and recurse. Bounding the size (number of leaves) of BSP trees has been a challenge. The long-standing conjecture that $O(n)$ size in \mathbb{R}^2 is achievable has recently been shown to be false. See Table 28.8.2 for selected results.

TABLE 28.8.2 BSP complexities.

DIM	CLASS	BOUND	SOURCE
2	segments	$O(n \log n)$	[PY90]
2	isothetic	$\Theta(n)$	[PY92]
2	fat	$\Theta(n)$	[dBdGO97]
2	segments	$\Omega(n[\log n / \log \log n])$	[Tót01c]
3	polyhedra	$O(n^2)$	[PY90]
3	polyhedra	$\Omega(n^2)$	Eppstein
3	isothetic	$\Theta(n^{3/2})$	[PY92]
3	fat orthog. rects.	$n^{O(\sqrt{\log n})}$	[AGMV00]

28.8.3 POLYHEDRAL TERRAINS

Polyhedral terrains are an important special class of 3D surfaces, arising in a variety of applications, most notably geographic information systems ([Chapter 58](#)).

GLOSSARY

Polyhedral terrain: A polyhedral surface that intersects every vertical line in at most a single point.

Perspective view: A view from a point.

Orthographic view: A view from infinity (parallel lines of sight).

Ray-shooting query: A query asking which terrain face is first hit by a ray shooting in a given direction from a given point. (See [Chapter 37](#).)

$\alpha(n)$: The inverse Ackermann function (nearly a constant). See [Section 47.4](#).

COMBINATORIAL BOUNDS

Several almost-tight bounds on the maximum number of combinatorially different views of a terrain have been obtained, as listed in Table 28.8.3.

TABLE 28.8.3 Bounds for polyhedral terrains.

VIEW TYPE	BOUND	SOURCE
Along vertical	$O(n^2 2^{\alpha(n)})$	[CS89]
Orthographic	$O(n^{5+\epsilon})$	[AS94]
Perspective	$O(n^{8+\epsilon})$	[AS94]

Bose et al. established that $\lfloor n/2 \rfloor$ vertex guards are sometimes necessary and always sufficient to guard a polyhedral terrain of n vertices [BSTZ97, BKL96].

ALGORITHMS

Algorithms seek to exploit the terrain constraints to improve on the same computations for general polyhedra:

1. To compute the orthographic view from above the terrain:
time $O((k + n) \log n \log \log n)$, where k is the output size [RS88].
2. To preprocess for $O(\log n)$ ray-shooting queries for rays with origin on a vertical line [BDEG94].

28.8.4 3D VISIBILITY GRAPHS

GLOSSARY

Aspect graph: A graph with a node for each combinatorially distinct view of a collection of polyhedra, with two nodes connected by an arc if the views can be reached directly from one another by a continuous movement of the viewpoint.

Isothetic: Edges parallel to Cartesian coordinate axes.

Box visibility graph: A graph realizable by disjoint isothetic boxes in 3D with orthogonal visibility.

K_n : The complete graph on n nodes.

There have been three primary motivations for studying visibility graphs of objects in three dimensions.

1. Computer graphics: Useful for accelerating interactive “walkthroughs” of complex polyhedral scenes [TS91], and for radiosity computations [TH93]. See [Chapter 49](#).
2. Computer vision: “Aspect graphs” are used to aid image recognition. The maximum number of nodes in an aspect graph for a polyhedron of n vertices depends on both convexity and the type of view. See Table 28.8.4. Note that the nonconvex bounds are significantly larger than those for terrains.

TABLE 28.8.4 Combinatorial complexity of visibility graphs.

CONVEXITY	ORTHOGRAPHIC	PERSPECTIVE	SOURCE
Convex polyhedron	$\Theta(n^2)$	$\Theta(n^3)$	[PD90]
Nonconvex polyhedron	$\Theta(n^6)$	$\Theta(n^9)$	[GCS91]

3. Combinatorics: It has been shown that K_{22} is realizable by disjoint isothetic rectangles in “ $2\frac{1}{2}D$ ” with vertical visibility (all rectangles are parallel to the xy -plane), but that K_{56} (and therefore all larger complete graphs) cannot be so represented [BEF⁺93]. It is known that K_{42} is a box visibility graph [BJMO94] but that K_{184} is not [FM99].

28.9 PENETRATING ILLUMINATION OF CONVEX BODIES

A rich vein of problems was initiated by Hadwiger, Levi, Gohberg and Markus; see [MS99] for the complex history. The problems employ a different notion of

exterior illumination, which could be called ***penetrating illumination*** (or perhaps “stabbing”), and focuses on a single convex body in \mathbb{R}^d .

GLOSSARY

Penetrating illumination: An exterior point x penetratingly illuminates a point y on the boundary ∂K of an object K if the ray from x through y has a non-empty intersection with the interior $\text{int } K$ of K .

Direction illumination: A point $y \in \partial K$ is illuminated from direction \mathbf{v} if the ray from the exterior through y with direction \mathbf{v} has a non-empty intersection with $\text{int } K$.

Affine symmetry: An object has affine symmetry if it unchanged after reflection through a point, reflection in a plane, or rotation about a line by angle $2\pi/n$, $n = 2, 3, \dots$

The central problem may be stated: What is the fewest number of exterior points sufficient to penetrantly illuminate any compact, convex body K in \mathbb{R}^d ? The problem is only completely solved in 2D: 4 lights are needed for a parallelogram, and 3 for all other convex bodies. In 3D it is known that 8 lights are needed for a parallelepiped (Fig. 28.9.1), and conjectured that 7 suffice for all other convex bodies. Bezdek proved that 8 lights suffice for any 3-polytope with an affine symmetry [Bez93]. Lassak proved that no more than 20 lights are needed for any compact, convex body in 3D [Bol81].

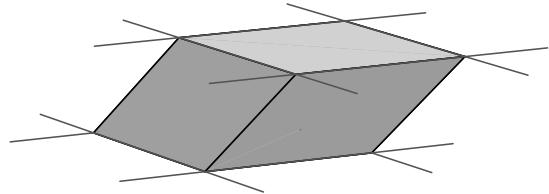


FIGURE 28.9.1

A parallelepiped requires $2^3 = 8$ lights for penetrating illumination of its boundary.

One reason for the interest in this problem is its connection to other problems, particularly covering problems. Define:

$I_0(K)$: the fewest number of points sufficient to penetrantly illuminate K .

$I_\infty(K)$: the fewest number of directions sufficient to direction-illuminate K .

$H(K)$: the fewest number of smaller homothetic copies of K that cover K .

$i(K)$: the fewest number of copies of $\text{int } K$ that cover K .

Remarkably,

$$I_0(K) = I_\infty(K) = H(K) = i(K).$$

as established by Boltjanski, Hadwiger, and Soltan; see again [MS99]. Several have conjectured that these quantities are $\leq 2^d$ for compact, convex bodies in \mathbb{R}^d , with equality only for the d -parallelotope.. The conjecture has been established only for special classes of bodies, e.g., [Bol01].

28.10 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

- [O'R87]: A monograph devoted to art gallery theorems and visibility algorithms.
- [She92]: A survey of art gallery theorems and visibility graphs, updating [O'R87].
- [O'R92]: A short update to [She92].
- [Urr00]: The latest art gallery results, updating [She92].
- [O'R93]: Survey of visibility graph results.
- [AGS00]: Survey of visibility algorithms in \mathbb{R}^2 .
- [MSD00]: Survey of link-distance algorithms.
- [Dor94]: A survey of hidden-surface removal algorithms, emphasizing recent theoretical developments.
- [Mur99]: A recent Ph.D. thesis on hidden-surface removal algorithms.
- [MS99]: Survey of illumination of convex bodies.

RELATED CHAPTERS

- Chapter 25: Triangulations
- Chapter 26: Polygons
- Chapter 27: Shortest paths and networks
- Chapter 37: Ray shooting and lines in space
- Chapter 38: Geometric intersection
- Chapter 49: Computer graphics
- Chapter 51: Pattern recognition
- Chapter 58: Geographic information systems

REFERENCES

- [AGMV00] P.K. Agarwal, E.F. Grove, T.M. Murali, and J.S. Vitter. Binary space partitions for fat rectangles. *SIAM J. Comput.*, 29:1422–1448, 2000.
- [AS94] P.K. Agarwal and M. Sharir. On the number of views of polyhedral terrains. *Discrete Comput. Geom.*, 12:177–182, 1994.
- [AAAS94] P.K. Agarwal, N. Alon, B. Aronov, and S. Suri. Can visibility graphs be represented compactly? *Discrete Comput. Geom.*, 12:347–365, 1994.
- [AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.

- [AS00] P.K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: The union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24:645–685, 2000.
- [Agg84] A. Aggarwal. *The art gallery problem: Its variations, applications, and algorithmic aspects*. Ph.D. thesis, Dept. of Comput. Sci., Johns Hopkins Univ., Baltimore, 1984.
- [AL93] M.H. Alsuwaiyel and D.T. Lee. Minimal link visibility paths inside a simple polygon. *Comput. Geom. Theory Appl.*, 3:1–25, 1993.
- [ADD⁺98] B. Aronov, A.R. Davis, T.K. Dey, S.P. Pal, and D.C. Prasad. Visibility with multiple reflections. *Discrete Comput. Geom.*, 20:61–78, 1998.
- [AGS00] Te. Asano, S.K. Ghosh, and T.C. Shermer. Visibility in the plane. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 829–876. Elsevier North-Holland, Amsterdam, 2000.
- [BDEG94] M. Bern, D.P. Dobkin, D. Eppstein, and R. Grossman. Visibility with a moving point of view. *Algorithmica*, 11:360–378, 1994.
- [Bez93] K. Bezdek. Hadwiger-Levi’s covering problem revisited. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 199–233. Springer-Verlag, Berlin, 1993.
- [Bol81] V. Boltjansky. Combinatorial geometry. *Algebra Topol. Geom.*, 19:209–274, 1981. In Russian. Cited in [MS99].
- [Bol01] V. Boltjansky. Solution of the illumination problem for bodies with md $m = 2$. *Discrete Comput. Geom.*, 26:527–541, 2001.
- [BDHS97] P. Bose, A.M. Dean, J.P. Hutchinson, and T.C. Shermer. On rectangle visibility graphs. *Proc. Graph Drawing*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 25–35. Springer-Verlag, Berlin, 1997.
- [BEF⁺93] P. Bose, H. Everett, S. Fekete, A. Lubiwi, H. Meijer, K. Romanik, T.C. Shermer, and S.H. Whitesides. On a visibility representation for graphs in three dimensions. *Proc. ALCOM Int. Work. Graph Drawing*. G. Di Battista, P. Eades, H. de Fraysseix, P. Rosenstiehl, and R. Tamassia, editors, pages 61–62, 1993.
- [BJMO94] P. Bose, A. Josephczyk, J. Miller, and J. O’Rourke. K_{42} is a box visibility graph. In *Snapshots in Comput. Geom.*, pages 88–91. Univ. Saskatchewan, 1994.
- [BKL96] P. Bose, D.G. Kirkpatrick, and Z. Li. Efficient algorithms for guarding or illuminating the surface of a polyhedral terrain. *Proc. Canad. Conf. Comput. Geom.*, 217–222, 1996.
- [BSTZ97] P. Bose, T.C. Shermer, G.T. Toussaint, B. Zhu. Guarding polyhedral terrains. *Comput. Geom. Theory Appl.*, 7: 173–185, 1997.
- [Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
- [CS89] R. Cole and M. Sharir. Visibility problems for polyhedral terrains. *J. Symbolic Comput.*, 7:11–30, 1989.
- [CRCU93] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia. Illuminating rectangles and triangles in the plane. *J. Combin. Theory Ser. B*, 57:1–17, 1993.
- [DN94] G. Das and G. Narasimhan. Optimal linear-time algorithm for the shortest illuminating line segment. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 259–266, 1994.
- [dBHO⁺94] M. de Berg, D. Halperin, M.H. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. *Algorithmica*, 12:30–53, 1994.

- [dBO92] M. de Berg and M.H. Overmars. Hidden surface removal for c -oriented polyhedra. *Comput. Geom. Theory Appl.*, 1:247–268, 1992.
- [dBdGO97] M. de Berg, M. de Groot, and M.H. Overmars. New results on binary space partitions in the plane. *Comput. Geom. Theory Appl.*, 8:317–333, 1997.
- [Dév86] F. Dévai. Quadratic bounds for hidden line elimination. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 269–275, 1986.
- [DLS92] H.N. Djidjev, A. Lingas, and J.-R. Sack. An $O(n \log n)$ algorithm for computing the link center of a simple polygon. *Discrete Comput. Geom.*, 8:131–152, 1992.
- [Dor94] S.E. Dorward. A survey of object-space hidden surface removal. *Internat. J. Comput. Geom. Appl.*, 4:325–362, 1994.
- [ECOUX95] V. Estivill-Castro, J. O'Rourke, J. Urrutia, and D. Xu. Illumination of polygons with vertex floodlights. *Inform. Process. Lett.*, 56:9–13, 1995.
- [Fej77] L. Fejes Tóth. Illumination of convex discs. *Acta Math. Acad. Sci. Hungar.*, 29(3–4):355–360, 1977.
- [FM99] S.P. Fekete and H. Meijer. Rectangle and box visibility graphs in 3d. *Internat. J. Comput. Geom. Appl.*, 9:1–27, 1999.
- [Gho97] S.K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete Comput. Geom.*, 17:143–162, 1997.
- [GM91] S.K. Ghosh and D.M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20:888–910, 1991.
- [GCS91] Z. Gigus, J.F. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:542–551, 1991.
- [GOH⁺02] A. García-Olaverri, F. Hurtado, M. Noy and J. Tejel. On the minimum size of visibility graphs. *Inform. Proc. Lett.*, 81: 223–230, 2002.
- [GSG92] M.T. Goodrich, S. Shauck, and S. Guha. Parallel methods for visibility and shortest path problems in simple polygons. *Algorithmica*, 8:461–486, 1992.
- [GHL⁺87] L.J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- [GHKS96] E. Győri, F. Hoffmann, K. Kriegel, and T.C. Shermer. Generalized guarding and partitioning for rectilinear polygons. *Comput. Geom. Theory Appl.*, 6:21–44, 1996.
- [HH00] L. Halbisen and N. Hungerbühler. On periodic billiard trajectories in obtuse triangles. *SIAM Rev.*, 42:657–670, 2000.
- [Her89] J. Hershberger. An optimal visibility graph algorithm for triangulated simple polygons. *Algorithmica*, 4:141–155, 1989.
- [HK93] F. Hoffmann and K. Kriegel. A graph coloring result and its consequences for some guarding problems. In *Proc. 4th Annu. Internat. Sympos. Algorithms Comput. (ISAAC 93)*, volume 762 of *Lecture Notes Comput. Sci.*, pages 78–87. Springer-Verlag, Berlin, 1993.
- [HK96] F. Hoffmann and K. Kriegel. A graph coloring result and its consequences for some guarding problems. *SIAM J. Discrete Math.*, 9:210–224, 1996.
- [HN01] F. Hurtado and M. Noy. On the number of visibility graphs of simple polygons. *Discrete Math.*, 232: 139–144, 2001.
- [HT01] M. Hoffmann and Cs. Tóth. Segment endpoint visibility graphs are Hamiltonian. In *Proc. 13th Canad. Conf. Comput. Geom.*, pages 109–112, 2001.

- [JS87] B. Joe and R.B. Simpson. Correction to Lee’s visibility polygon algorithm. *BIT*, 27:458–473, 1987.
- [KOS92] M.J. Katz, M.H. Overmars, and M. Sharir. Efficient hidden surface removal for objects with small union size. *Comput. Geom. Theory Appl.*, 2:223–234, 1992.
- [Kle69] V. Klee. Is every polygonal region illuminable from some point? *Amer. Math. Monthly*, 76:180, 1969.
- [KW91] V. Klee and S. Wagon. *Old and New Unsolved Problems in Plane Geometry*. Math. Assoc. Amer., 1991.
- [LP79] D.T. Lee and F.P. Preparata. An optimal algorithm for finding the kernel of a polygon. *J. Assoc. Comput. Mach.*, 26:415–421, 1979.
- [LC94] S.-Y. Lin and C. Chen. Planar visibility graphs. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 30–35, 1994.
- [MSD00] A. Maheshwari, J.-R. Sack, and H.N. Djidjev. Link distance problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 519–558. Elsevier North-Holland, Amsterdam, 2000.
- [MS99] H. Martini and V. Soltan. Combinatorial problems on the illumination of convex bodies. *Aequationes Math.*, 57:121–152, 1999.
- [McK87] M. McKenna. Worst-case optimal hidden-surface removal. *ACM Trans. Graph.*, 6:19–28, 1987.
- [Mur99] T.M. Murali. *Efficient Hidden-Surface Removal in Theory and in Practice*. Ph.D. thesis, Brown Univ., 1999.
- [NS91] B.J. Nilsson and S. Schuierer. An optimal algorithm for the rectilinear link center of a rectilinear polygon. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes Comput. Sci.*, pages 249–260. Springer-Verlag, Berlin, 1991.
- [O’R83] J. O’Rourke. Galleries need fewer mobile guards: a variation on Chvátal’s theorem. *Geom. Dedicata*, 14:273–283, 1983.
- [O’R87] J. O’Rourke. *Art Gallery Theorems and Algorithms. Internat. Series Monographs Computer Science*. Oxford University Press, New York, 1987.
- [O’R92] J. O’Rourke. Computational geometry column 15. *Internat. J. Comput. Geom. Appl.*, 2:215–217, 1992. Also in *SIGACT News*, 23:2, 1992.
- [O’R93] J. O’Rourke. Computational geometry column 18. *Internat. J. Comput. Geom. Appl.*, 3:107–113, 1993. Also in *SIGACT News*, 24:1:20–25, 1993.
- [OP01] J. O’Rourke and O. Petrovici. Narrowing light rays with mirrors. In *Proc. 13th Canad. Conf. Comput. Geom.*, pages 137–140, 2001.
- [PY90] M.S. Paterson and F.F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete Comput. Geom.*, 5:485–503, 1990.
- [PY92] M.S. Paterson and F.F. Yao. Optimal binary space partitions for orthogonal objects. *J. Algorithms*, 13:99–113, 1992.
- [PD90] H. Plantinga and C.R. Dyer. Visibility, occlusion, and the aspect graph. *Internat. J. Comput. Vision*, 5:137–160, 1990.
- [R03] D. Rappaport. The visibility graph of congruent discs is Hamiltonian. *Internat. J. Comput. Geom. Appl.*, 25:257–265, 2003.
- [RS88] J.H. Reif and S. Sen. An efficient output-sensitive hidden-surface removal algorithms and its parallelization. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 193–200, 1988.

- [SO92] M. Sharir and M.H. Overmars. A simple output-sensitive algorithm for hidden-surface removal. *ACM Trans. Graph.*, 11:1–11, 1992.
- [SE87] X. Shen and H. Edelsbrunner. A tight lower bound on the size of visibility graphs. *Inform. Process. Lett.*, 26:61–64, 1987.
- [She94] T.C. Shermer. A tight bound on the combinatorial edge guarding problem. In *Snapshots in Comput. Geom.*, pages 191–223. Univ. Saskatchewan, 1994.
- [She92] T.C. Shermer. Recent results in art galleries. *Proc. IEEE*, 80:1384–1399, 1992.
- [ST03] B. Speckmann and Cs. Tóth. Allocating vertex π -guards in simple polygons via pseudo-triangulations. *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 109–118, 2003.
- [Ste96] I. Stewart. Mathematical recreations. *Sci. Amer.*, 275:100–103, 1996. Includes light in circular forest problem due to J. Pach.
- [TH93] S. Teller and P. Hanrahan. Global visibility algorithms for illumination computations. In *Proc. ACM Conf. SIGGRAPH 93*, pages 239–246, 1993.
- [TS91] S.J. Teller and C.H. Séquin. Visibility preprocessing for interactive walkthroughs. *Proc. ACM Conf. SIGGRAPH 91*, pages 61–69, 1991.
- [Tok95] G.W. Tokarsky. Polygonal rooms not illuminable from every point. *Amer. Math. Monthly*, 102:867–879, 1995.
- [Tót00] Cs. Tóth. Art gallery problem with guards whose range of vision is 180° . *Comput. Geom. Theory Appl.*, 17:121–134, 2000.
- [Tót01a] Cs. Tóth. Illuminating both sides of line segments. In *Lecture Notes Comput. Sci.*, volume 2098, pages 370–380. Springer-Verlag, Berlin, 2001.
- [Tót01b] Cs. Tóth. Guarding disjoint triangles and claws in the plane. In *Abstracts 17th European Workshop Comput. Geom.*, pages 137–139. Freie Universität Berlin, 2001.
- [Tót01c] Cs. Tóth. A note on binary plane partitions. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 151–156, 2001.
- [Tót02] Cs. Tóth. Illumination in the presence of opaque line segments in the plane. *Comput. Geom. Theory Appl.*, 21:193–204, 2002.
- [Tót03a] Cs. Tóth. Illumination of polygons by 45° -floodlights. *Discrete Math.*, 265:251–260, 2003.
- [Tót03b] Cs. Tóth. Illuminating disjoint line segments in the plane. *Discrete Comput. Geom.*, 30:489–505, 2003.
- [Urr00] J. Urrutia. Art gallery and illumination problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 973–1027. Elsevier North-Holland, Amsterdam, 2000.

29 GEOMETRIC RECONSTRUCTION PROBLEMS

Steven S. Skiena

INTRODUCTION

Many problems from mathematics and engineering can be described in terms of reconstruction from geometric information. **Reconstruction** is the algorithmic problem of combining the results of one or more measurements of some aspect of a physical or mathematical object to obtain certain desired information about the object. Geometric reconstruction problems arise in a number of applications, such as robotics and computer-aided tomography, and also are of theoretical interest.

In this chapter, we consider three different classes of geometric reconstruction problems. In Section 29.1, we examine static reconstruction problems, where we are given a geometric structure G' derived from an original structure G , and seek to invert this transformation. In Section 29.2, we consider interactive reconstruction problems, where we are permitted to “probe” the object at arbitrary places and seek to reconstruct the desired structure using the fewest such probes. Finally, in Section 29.3, we provide pointers to the literature for work on (typically ill-defined) geometric reconstruction problems that often arise in practice.

29.1 STATIC RECONSTRUCTION PROBLEMS

Here we consider inverse problems of the following type. Let A be a geometric structure, and T a transformation such that $T(A) \rightarrow B$, where B is some different geometric structure. Now, given T and B , construct a structure A' such that $T(A') \rightarrow B$. If T is 1–1, then $A = A'$. If not, we may be interested in finding or counting all solutions.

An example of an important class of reconstruction problems is recognizing visibility graphs, i.e., given a graph G , construct a polygon P whose visibility graph is G . See [Section 28.2](#).

Results on static reconstruction problems are summarized in [Table 29.1.1](#). We characterize each problem by its input instance and desired inverted structure. Static reconstruction problems include reconstructing sets of points from inter-point distances, extended Gaussian images [GH95][GM03], points from Voronoi diagrams [AB85], and orthogonal polygons from points [O'R88].

A special class of problems concerns proximity drawability. Given a graph G , we seek a set of points corresponding to vertices of G such that two points are “sufficiently” close iff there is an edge in G for the corresponding vertices. Examples of proximity drawability problems include finding points to realize graphs as minimum spanning trees (MST), Delaunay triangulations ([Chapter 25](#)), Gabriel graphs, and relative neighborhood graphs (RNGs) ([Chapter 51](#)). Although many of the results are quite technical, Di Battista et al. [DLL95] provide an excellent survey

of results on these and other classes of proximity drawings; see also [Chapter 52](#).

To provide some intuition about the minimum spanning tree results, observe that very low degree graphs are easily embedded as point sets. If the maximum degree is 2, i.e., the graph is a simple path, then any straight line embedding will work. To realize a vertex v of degree 6 as a minimum spanning tree, a geometric argument shows that all adjacent points must be spaced at equal angles of 60 degrees around v , a very restrictive condition leading to the hardness result. Such equal spacing is not possible for degree larger than six.

GLOSSARY

Extended Gaussian image: A transform that maps each face of a convex polyhedron to a vector normal to the face whose length is proportional to the area of the face. These vectors uniquely represent convex polyhedra and have been applied to problems in robot vision.

Hammer's X-ray problem: Given a fixed set of X-ray projections of a convex body, can you reconstruct the body?

Determination: A class of sets is determined by n directions if there are n fixed directions such that all sets can be reconstructed from projections along these directions.

Verification: A class of sets is verified by n directions if, for each particular set, there are n projections that distinguish this set from any other.

Gabriel graph: A graph whose vertices are points in \mathbb{E}^2 , with edge (x, y) iff points x and y define the diameter of an empty circle.

Relative neighborhood graph: A graph whose vertices are points in \mathbb{E}^2 , with an edge (x, y) iff there exists no point z such that z is closer to x than y is and z is closer to y than x is. See [Section 51.2](#).

Interpoint distances: The complete set of $\binom{n}{2}$ distances defined between pairs of points in an n point set. The distance set is **labeled** if the identities of the two points defining the distance are associated with the distance, and **unlabeled** otherwise.

A final set of problems concern reconstructing objects from a fixed set of X-ray projections, conventionally called Hammer's X-ray problem. Different problems arise depending upon whether the X-rays originate from a point or line source, and whether we seek to verify or determine the object. A selection of results on parallel X-rays (line sources) are listed in [Table 29.1.2](#). For example, parallel X-rays in certain sets of four directions suffice to determine any convex body; the directions must not be a subset of the edges of an affinely regular polygon. If the directions do form such a subset, then there exist noncongruent polygons that are not distinguished by any number n of parallel X-rays in these directions. Nevertheless, any pair of nonparallel directions suffice to determine “most” (in the sense of Baire category) convex sets.

There is also a collection of results on *point source X-rays*. For example, convex sets in \mathbb{E}^2 are determined by directed X-rays from three noncollinear point sources. The substantial literature on such X-ray problems is most ably covered by Gardner's monograph [Gar95], from which several of the open problems listed below are drawn.

TABLE 29.1.1 Static reconstruction problems.

INPUT	INVERTED STRUCTURE	RESULT
Tree with max degree ≤ 5	points embedding it as MST	every tree realizable
Tree with max degree 6	points embedding it as MST	NP-hard
Tree with max degree ≥ 7	points embedding it as MST	no tree realizable
Planar graph	points embedding it as a Gabriel graph	partial characterization
Planar graph	points embedding it as a RNG	partial characterization
Triangulated graph	points embedding it as a Delaunay tri orthogonal polygon through them	partial characterization
Points in \mathbb{E}^2	points embedding it as a Voronoi diag	algorithm: $O(n \log n)$
Planar graph	convex polyhedra in \mathbb{E}^3	partial characterization
Extended Gaussian image	points realizing these in \mathbb{E}^d	algorithm: $O(n \log n)$ per iter
Labeled interpoint dists	points realizing these on \mathbb{E}^1	algorithm: $O(2^d n^2)$
Unlabeled interpoint dists	points realizing these in \mathbb{E}^d	algorithm: $O(2^n n \log n)$
Unlabeled interpoint dists		NP-hard

Discrete tomography is a new area of study inspired by the use of electron microscopy to reconstruct the positions of atoms in crystal structures. A typical problem is placing integers in a matrix so as to realize a given set of row and column sums. The problem becomes more complex when the reconstructed body must satisfy connectivity constraints or simultaneously satisfy row/column sums of multiple colors. A collection of survey articles on discrete tomography is presented by Herman and Kuba [HK99].

TABLE 29.1.2 Selected results on parallel X-rays (Hammer's problem).

DIM	PROBLEM	SETS	RESULT
2	verify	convex polygons	2 parallel X-rays do not suffice
	verify	convex set	3 parallel X-rays suffice
	determine	convex set	4 parallel X-rays suffice ($\not\subseteq$ affinely reg polygon)
	determine	convex set	n parallel X-rays do not suffice (\subseteq affinely reg polygon)
	determine	convex set	2 parallel X-rays “usually” suffice
	determine	star-shaped polygons	no finite set of directions suffice
3	determine	convex body	4 parallel X-rays suffice (coplanar directions)
	determine	convex body	4 parallel X-rays do not suffice (noncoplanar directions)
d	determine	convex body	2 parallel X-rays “usually” suffice
	determine	compact sets	no finite set of directions suffice

OPEN PROBLEMS

1. Give an algorithm (polynomial in n) to reconstruct a set of n points on a line from the set of $\binom{n}{2}$ unlabeled interpoint distances it defines. See [SSL90].
2. Do there exist two distinct n -point sets, $n \geq 7$, realizing identical unlabeled interpoint distance sets, where each distance is unique in the set? See [Blo77].
3. Characterize the convex sets in \mathbb{E}^2 that can be determined by two parallel X-rays [Gar95, Problem 1.1].
4. Are convex bodies in \mathbb{E}^3 determined by parallel X-rays in some set of five directions [Gar95, Problem 2.2]?
5. Find an algorithm to reconstruct a convex set from its directed X-rays from three noncollinear points [Gar95, Problem 5.5]. The uniqueness proof is non-constructive.
6. Given both the red and blue column sums of a matrix, color the matrix elements red, blue, and white so as to realize these sums. The problem is known to be polynomial for one color and NP-complete for three or more colors [CD01, Dur01].
7. Given the (single color) column sums of a matrix, find a convex polyomino which realizes these sums, if one exists. It is open as to whether there exists a polynomial-time algorithm for this problem [CD01, Dur01].

29.2 INTERACTIVE RECONSTRUCTION PROBLEMS

Geometric probing considers problems of determining a geometric structure or some aspect of that structure from the results of a mathematical or physical measuring device, a **probe**. A variety of problems from robotics, medical instrumentation, mathematical optimization, integral and computational geometry, graph theory, and other areas fit into this paradigm. The key issue is interaction, where the n th probe depends upon the outcome of the previous probes.

The problem of geometric probing was introduced by Cole and Yap [CY87] and inspired by work in robotics and tactile sensing (Section 48.7). A substantial body of work has followed it, which is extensively surveyed in [Ski92]. A collection of open problems in probing appears in [Ski89].

GLOSSARY

Determination: The algorithmic problem of computing how many probes of a particular model are necessary to completely determine or reconstruct an object drawn from a particular class of objects.

Verification: The algorithmic problem, given a supposed description of an object, of computing how many probes of a particular model are necessary to test if the description is valid.

Model-based: A problem where any object is constrained to be one of a known, finite set of m possible objects.

Point probe: An oracle that tests whether a given point is within an object or not.

Finger probe: An oracle that returns the first point of intersection between a directed line and an object.

Hyperplane probe: An oracle that measures the first time at which a hyperplane moving parallel to itself intersects an object.

X-ray probe: An oracle that measures the length of the intersection between a line and an object.

Silhouette probe: An oracle that returns a $(d-1)$ -dimensional projection (in a given direction) of a d -dimensional object.

Halfspace probe: An oracle that measures the area or volume of the intersection between a halfspace and an object.

Cut-set probe: An oracle that for a specified graph and partition of the vertices returns the size of the cut-set determined by the partition.

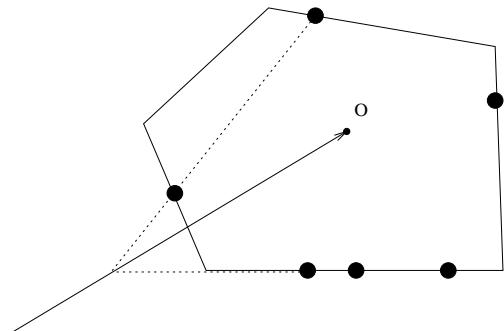


FIGURE 29.2.1

Determining the next edge of P using finger probes.

MAIN RESULTS

For a particular probing model, the determination problem asks how many probes are sufficient to completely reconstruct an object from a given class. For example, Cole and Yap's strategy for reconstructing a convex polygon P from finger probes is based on the observation that three collinear contact points must define an edge. The strategy, illustrated in Figure 29.2.1, aims probes at the intersection point between a confirmed edge (defined by three collinear points) and a conjectured edge (defined by two contact points). If this intersection point is indeed a contact point, another vertex is determined due to convexity; if not, the existence of another edge is known. Since we avoid probing the interior of any edge that has been determined, $\approx 3n$ probes suffice in total since no more than one edge can be hit four times.

Table 29.2.1 summarizes probing results for a wide variety of models. In the table, $f_i(P)$ denotes the number of i -dimensional faces of P .

TABLE 29.2.1 Upper and lower bounds for determination for various probing models.

PROBE	OBJECT	LOWER BOUND	UPPER BOUND
Finger	convex n -gon	$3n$	$3n$
Finger	convex n -gon w known n	$2n + 1$	$3n - 1$
Finger	convex polyhedra in \mathbb{E}^d	$df_0(P) + fd_{d-1}(P)$	$f_0(P) + (d+2)f_{d-1}(P)$
Finger	n -gon from m conv models	$n - 1$	$n + 3$
2 fingers	convex n -gon	$2n - 2$	$2n$
3 fingers	convex n -gon	$2n - 3$	$2n$
4 or 5 fingers	convex n -gon	$(4n - 5)/3$	$\lfloor(4n + 2)/3\rfloor$
$k \geq 6$ fingers	convex n -gon	n	$n + 1$
Enhanced fingers	n -gon w noncollinear edges	$3n - 3$	$3n - 3$
Line	convex n -gon	$3n + 1$	$3n + 1$
Line	n -gon from m conv models	$2n - 3$	$2n + 4$
Silhouette	convex n -gon	$3n - 2$	$3n - 2$
Silhouette	convex polyhedra in \mathbb{E}^3	$f_2(P)/2$	$5f_0(P) + f_2(P)$
X-ray	convex n -gon	$3n - 3$	$5n - 19$
Parallel X-ray	convex n -gon	3	3
Parallel X-ray	nondegenerate n -gon	$\lfloor \log n \rfloor - 2$	$2n + 2$
Halfplane	convex n -gon	$2n$	$7n + 7$
Cut-set	embedded graph	$\binom{n}{2}$	$\binom{n}{2}$
Cut-set	unembedded graph	$\Omega(n^2 / \log n)$	$O(n^2 / \log n)$

Cole and Yap's finger probing model is not powerful enough to determine nonconvex objects. There are three major reasons for this. A tiny crack in an edge can go forever undetected, since no finite strategy can explore the entire surface of the polygon. Second, it is easy to construct nonconvex polygons whose features cannot be entirely contacted with straight-line probes originating from infinity. Finally, for nonconvex polygons there exists no constant k such that k collinear probes determine an edge. To generalize the class of objects, enhanced finger probes have been considered. One such probe [ABY90] returns surface normals as well as contact points, eliminating the second problem. When restricted to polygons with no two edges defined by the same supporting line, the first and third problems are eliminated.

In the verification problem, we are given a description of a putative object, and charged with using a small number of probes to prove that the description is correct. Verification is clearly no harder than determination, since we are free to ignore the description in planning the probes, and could simply compare the determined object to its description. Sometimes significantly fewer probes suffice for verification. For example, we can verify a putative convex polygon in $2n$ probes by sending one finger probe to contact each vertex and the interior of each edge. This gives three contact points on each edge, which by convexity suffices to verify the polygon. Table 29.2.2 summarizes results in verification.

Of course, there are other classes of problems that do not fit so easily into the confines of these tables. Verification is closely related to approximate geometric testing; see [ABP⁺97, Rom95]. An interesting application of probing to nonconvex

polygons is presented in [HP99]. See [Ric97, Ski92] for discussions of probing with uncertainty and tactile sensing in robotics.

TABLE 29.2.2 Upper and lower bounds for verification for various probing models.

PROBE	OBJECT	LOWER BOUND	UPPER BOUND
Finger	convex n -gon	$2n$	$2n$
Finger	convex n -gon with known n	$3\lceil n/2 \rceil$	$3\lceil n/2 \rceil$
Line	convex n -gon	$2n$	$2n$
X-ray	convex n -gon	$3n/2$	$3n/2 + 6$
Halfplane	convex n -gon	$2n/3$	$n + 1$

OPEN PROBLEMS

1. Tighten the gap between the lower and upper bounds for determination for finger probes in higher dimensions [DEY86].
2. Tighten the bounds for determination of convex n -gons with X-ray probes. Does a finite number (i.e., $f(n)$) of parallel X-ray probes suffice to verify or determine simple n -gons? Since each parallel X-ray probe provides a representation of the complete polygon, there is hope to detect arbitrarily small cracks in a finite number of probes; but see [MS96].
3. Consider generalizations of halfplane probes to higher dimensions. How many probes are necessary to determine convex (or nonconvex) polyhedra?
4. Silhouette probes return the shadow cast by a polytope in a specified direction. These dualize to *cross-section probes* that return a slice of the polytope. Tighten the current bounds [DEY86] on determination with silhouettes in \mathbb{E}^3 .

29.3 ILL-POSED RECONSTRUCTION PROBLEMS

Many geometric reconstruction problems that arise in practice are inherently ill-defined. In this section, we mention a class of approximate reconstruction problems, typically inspired by practical problems, and describe a few approaches toward dealing with them. Specific results are not discussed, but pointers to the literature are provided.

COMPUTER VISION

Computer vision is an enormous research area, with the goal of enabling computers to understand and interpret features in digital images. There are a variety of com-

puter vision problems that can be framed as reconstruction problems, particularly those that try to use several fixed images or active sensing, where the robot is free to decide where to look next to obtain more information about its environment [Fau93, Hor86].

A particularly interesting class of active sensing problems involves navigating in an unfamiliar terrain, where we seek a short path to a goal but learn about obstacles only as we encounter them. See [BRS91] for approximation results on this problem, and Sections 27.3 and 47.7 of this Handbook.

Decision trees are a commonly used classification procedure for recognizing an object drawn from a known class of models. The classification procedure takes the form of a rooted tree, where the models are leaves and each internal node corresponds to a test or probe. All of the probing strategies discussed in previous sections can be reformulated in terms of decision trees, with the goal of minimizing the heights of the trees. The general problem of minimizing the height of a decision tree is NP-complete, but approximation algorithms for minimizing the height of geometric decision trees are known [AMM⁺98, AGM⁺93].

SURFACES FROM DATA POINTS

As described in the introduction to this section, interpolating a surface from a finite set of points in three dimensions can be considered a geometric reconstruction problem. These problems often arise in cartographic data, where we seek to construct a model of a mountain given a set of points on the surface. The issue also arises in surface simplification, where given a surface we seek to approximate it with another surface with fewer points such that the maximum difference between elevations is minimized; see [Chapter 54](#). Curve and surface reconstruction has recently been cast into a new, no longer ill-posed form, with theoretical guarantees. See [Chapter 30](#) for a thorough survey.

One common approach consists of projecting the points to the plane, triangulating them, and converting this into a triangulated surface by projecting each vertex back into three dimensions. Triangulation-based approaches to surface reconstruction are surveyed in [MSS92].

TOMOGRAPHY AND SURFACES FROM CROSS-SECTIONS AND PROJECTIONS

CAT scanners and other tomographic imaging systems represent a tremendous step forward in our ability to diagnose tumors and other medical problems. Herman [Her80] defines tomography as “the process of producing an image of a two-dimensional distribution (usually of some physical property) from estimates of its line integrals along a finite number of lines of known locations.” Tomographic scanners estimate line integrals by sending an energy pulse of some type through an object and measuring how much energy is absorbed. Surveys of tomography include [SK78, SSW77].

The most important reconstruction algorithms are transform methods, which are direct implementations of the Radon inversion formula, derived using Fourier transform methods.

Electrical impedance tomography (EIT) is a recently developed medical imaging technology that constructs a map of the electrical conductivity of a region of

the body using probes that measure the resistance between pairs of surface points. See <http://www.eit.org.uk/> for a comprehensive list of references on electrical impedance tomography.

An important related geometric problem concerns splicing a series of these parallel slices into polyhedra. The most natural way to proceed is to triangulate between the slices, but this is not always possible without adding extra vertices [GOS96]. Practical algorithms include [Boi88, BS94]; see also [Section 26.6](#).

SHAPE FROM DISTRIBUTION OF CROSS-SECTIONS

In such fields as biology and geology, it is often necessary to reconstruct the shape and size distributions of particles from the cross-sections of samples. For example, cross-sections of cubes can be polygons with 3, 4, 5, or 6 sides, and the probability of each such event is well defined if the cross-sections are taken uniformly at random, as would be the case with small crystals inside a large mineral sample. This has given rise to a field known as *stereology* [Eli67, Hau63, Wei83], where such distributions are studied. A subfield known as *local stereology*, where the set of cross-sections is taken through a common point, has a particularly close connection to geometric tomography. See [Jen98] for details and <http://www.stereologysociety.org/> for a comprehensive survey of the stereology literature.

3D MODELS FROM 2D IMAGES

In the field of computer vision, it is often desirable to reconstruct a 3D model of an object consistent with one or more two-dimensional images of the object. The model is not necessarily unique, as there may be features that do not appear in any of the images. These problems are surveyed in Section 51.2.

After edge detection has been applied to the image, the primary algorithmic problem concerns identifying whether edges correspond to protrusions or indentations of the main object. Huffman-Clowes labeling is a constraint-based approach resulting from a case analysis of the possible types of junctions and shadows in the scene. Recent articles of such methods include [ABC⁺90, WG93, Whi89, Sug86].

29.4 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference can be traced through these surveys:

[DLL95]: Survey on embedding proximity graphs ([Table 29.1.1](#)).

[Gar95]: Survey of Hammer's X-ray problem and related work in geometric tomography.

[HK99]: Survey on discrete tomography.

[Rom95]: Survey on geometric testing.

[Ski92]: Survey on geometric probing (Table 29.2.1).

RELATED CHAPTERS

- [Chapter 28: Visibility](#)
 - [Chapter 30: Curve and surface reconstruction](#)
 - [Chapter 48: Robotics](#)
 - [Chapter 52: Graph drawing](#)
 - [Chapter 60: Rigidity and scene analysis](#)
-

REFERENCES

- [AB85] P.F. Ash and E.D. Bolker. Recognizing Dirichlet tessellations. *Geom. Dedicata*, 19:175–206, 1985.
- [ABC⁺90] N. Ayache, J.-D. Boissonnat, L. Cohen, B. Geiger, J. Levy-Vehel, O. Monga, and P. Sander. Steps toward the automatic interpretation of 3D images. In K.H. Höhne, H. Fuchs, and S.M. Pizer, editors, *3D Imaging in Medicine*, volume 60 of *NATO Adv. Sci. Inst. Ser. F: Comput. Systems Sci.*, pages 107–120. Springer-Verlag, Berlin, 1990.
- [ABP⁺97] E.M. Arkin, P. Belleville, J.S.B. Mitchell, D.M. Mount, K. Romanik, S. Salzberg, and D.L. Souvaine. Testing simple polygons. *Comput. Geom. Theory Appl.*, 8:97–114, 1997.
- [ABY90] P.D. Alevizos, J.-D. Boissonnat, and M. Yvinec. Non-convex contour reconstruction. *J. Symbolic Comput.*, 10:225–252, 1990.
- [AGM⁺93] E.M. Arkin, M.T. Goodrich, J.S.B. Mitchell, D.M. Mount, C.D. Piatko, and S.S. Skiena. Point probe decision trees for geometric concept classes. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes Comput. Sci.*, pages 95–106. Springer-Verlag, New York, 1993.
- [AMM⁺98] E.M. Arkin, H. Meijer, J.S.B. Mitchell, D. Rappaport, and S.S. Skiena. Decision trees for geometric models. *Internat. J. Comput. Geom. Appl.*, 8:343–363, 1998.
- [Blo77] G. Bloom. A counterexample to a theorem of Piccard. *J. Comb. Theory Ser. A*, 22:378–379, 1977.
- [Boi88] J.-D. Boissonnat. Shape reconstruction from planar cross-sections. *Comput. Vision Graph. Image Process.*, 44:1–29, 1988.
- [BRS91] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 494–503, 1991.
- [BS94] G. Barequet and M. Sharir. Piecewise-linear interpolation between polygonal slices. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 93–102, 1994.
- [CD01] M. Chrobak and C. Durr. Reconstructing polyatomic structures from discrete X-rays: NP-completeness proof for three atoms. *Theoret. Comput. Sci.*, 259:81–98, 2001.
- [CY87] R. Cole and C.K. Yap. Shape from probing. *J. Algorithms*, 8:19–38, 1987.
- [DEY86] D.P. Dobkin, H. Edelsbrunner, and C.K. Yap. Probing convex polytopes. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 424–432, 1986.
- [DLL95] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability: A survey. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing (Proc. GD '94)*, volume 894 of *Lecture Notes Comput. Sci.*, pages 328–339. Springer-Verlag, New York, 1995.

- [Dur01] C. Durr. Open problems in discrete tomography, <http://www.lri.fr/~durr/Xray/Complexity/>. Dec. 2001.
- [Eli67] H. Elias, editor. *Proc. 2nd Internat. Congress Stereology*. Springer-Verlag, New York, 1967.
- [Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, 1993.
- [Gar95] R.J. Gardner. *Geometric Tomography*. Cambridge University Press, 1995.
- [GM03] R.J. Gardner and P. Milanfar. Reconstruction of convex bodies from brightness functions. *Discrete Comput. Geom.*, 29: 279–303, 2003.
- [GH95] P. Gritzmann and A. Hufnagel. A polynomial time algorithm for Minkowski reconstruction. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 1–9, 1995.
- [GOS96] C. Gitlin, J. O’Rourke, and V. Subramanian. On reconstructing polyhedra from parallel slices. *Internat. J. Comput. Geom. Appl.*, 6:103–122, 1996.
- [Hau63] H. Haug, editor. *Proc. 1st Internat. Congress Stereology*. Druck Congressprint, Kau-nitzgasse, 1963.
- [Her80] G.T. Herman. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. Academic Press, New York, 1980.
- [HK99] G.T. Herman and A. Kuba. *Discrete Tomography: Foundations, Algorithms, and Applications*. Springer-Verlag, 1999.
- [Hor86] B.K.P. Horn. *Robot Vision*. MIT Press, Cambridge, 1986.
- [HP99] K. Hunter and T. Pavlidis. Non-interactive geometric probing: Reconstructing non-convex polygon. *Comput. Geom. Theory Appl.* 14:221–240, 1999.
- [Jen98] E. Jensen. *Local Stereology*. World Scientific, Singapore, 1998.
- [MS96] H. Meijer and S.S. Skiena. Reconstructing polygons from X-rays. *Geometriae Dedicata*, 61:191–204, 1996.
- [MSS92] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Trans. Graph.*, 11:228–258, 1992.
- [O’R88] J. O’Rourke. Uniqueness of orthogonal connect-the-dots. In G.T. Toussaint, editor, *Computational Morphology*, pages 97–104. North-Holland, Amsterdam, 1988.
- [Ric97] T. Richardson. Approximation of Planar Convex Sets from Hyperplane Probes. *Discrete Comput. Geom.* 18:151–177, 1997.
- [Rom95] K. Romanik. Geometric Probing and Testing—A Survey, DIMACS Tech. Rep. 95-42 Rutgers Univ., New Brunswick, 1995.
- [SK78] L.A. Shepp and J.B. Kruskal. Computerized tomography: The new medical X-ray technology. *Amer. Math. Monthly*, 85:420–439, 1978.
- [Ski89] S.S. Skiena. Problems in geometric probing. *Algorithmica*, 4:599–605, 1989.
- [Ski92] S.S. Skiena. Interactive reconstruction via geometric probing. *Proc. IEEE*, 80:1364–1383, 1992.
- [SSL90] S.S. Skiena, W.D. Smith, and P. Lemke. Reconstructing sets from interpoint distances. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 332–339, 1990.
- [SSW77] K.T. Smith, D.C. Solomon, and D. Wagner. Practical and mathematical aspects of the problem of reconstructing objects from radiographs. *Bull. Amer. Math. Soc.*, 83:1227–1270, 1977.
- [Sug86] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, Cambridge, 1986.

- [Wei83] W. Weil. Stereology: A survey for geometers. In P. Gruber and J. Wills, editors, *Convexity and Its Applications*, pages 360–412. Birkhäuser, Basel, 1983.
- [WG93] W. Wang and G.G. Grinstein. A survey of 3D solid reconstruction from 2D projection line drawings. *Comput. Graph. Forum*, 12:137–158, 1993.
- [Whi89] W. Whiteley. A matroid on hypergraphs, with applications in scene analysis and geometry. *Discrete Comput. Geom.*, 4:75–95, 1989.

30 CURVE AND SURFACE RECONSTRUCTION

Tamal K. Dey

INTRODUCTION

The problem of reconstructing a shape from its sample appears in many scientific and engineering applications. Because of the variety in shapes and applications, many algorithms have been proposed over the last two decades, some of which exploit application-specific information and some of which are more general. We will concentrate on techniques that apply to the general setting and have proved to provide some guarantees on the quality of reconstruction.

GLOSSARY

Simplex: A k -simplex in \mathbb{R}^d , $0 \leq k \leq d$, is the convex hull of $k + 1$ affinely independent points in \mathbb{R}^d where $0 \leq k \leq d$. The 0-, 1-, 2-, and 3-simplices are also called *vertices*, *edges*, *triangles*, and *tetrahedra* respectively.

Simplicial complex: A simplicial complex \mathcal{K} is a collection of simplices with the conditions that, (i) if $\sigma_1, \sigma_2 \in \mathcal{K}$ intersect, then $\sigma_1 \cap \sigma_2 \in \mathcal{K}$, and (ii) all simplices spanned by the vertices of a simplex in \mathcal{K} are also in \mathcal{K} . The underlying space of \mathcal{K} is the set of all points in its simplices. (Cf. [Chapter 31](#).)

k -manifold: A k -manifold is a topological space where each point has a neighborhood homeomorphic to \mathbb{R}^k or the halfspace \mathbb{H}^k . The points with \mathbb{H}^k neighborhood constitute the boundary of the manifold.

Voronoi diagram: Given a point set $P \in \mathbb{R}^d$, the Voronoi diagram V_P of P is a collection of Voronoi cells V_p for each point $p \in P$, where

$$V_p = \{x \in \mathbb{R}^d \mid \|x - p\| \leq \|x - q\| \forall q \in P\}.$$

Delaunay triangulation: The Delaunay triangulation of a point set $P \in \mathbb{R}^d$ is a simplicial complex D_P so that a simplex with vertices $\{p_0, \dots, p_k\}$ is in D_P if and only if $\bigcap_{i=0,k} V_{p_i} \neq \emptyset$. (Cf. [Chapter 22](#).)

Shape: A shape Σ is a subset of an Euclidean space.

Sample: A sample P of a shape Σ is a finite set of points from Σ .

Medial axis: The medial axis of a shape $\Sigma \in \mathbb{R}^d$ is the closure of the set of points in \mathbb{R}^d that have more than one closest point in Σ . See [Figure 30.1.1\(a\)](#) for an illustration.

Local feature size: The local feature size for a shape $\Sigma \subseteq \mathbb{R}^d$ is a continuous function $f : \Sigma \rightarrow \mathbb{R}$ where $f(x)$ is the distance of $x \in \Sigma$ to the medial axis of Σ . See [Figure 30.1.1\(a\)](#).

Uniform sample: A sample P of a shape Σ is δ -uniform if for each $x \in \Sigma$ there is a sample point $p \in P$ so that $\|p - x\| \leq \delta f_{min}$ where $f_{min} = \min\{f(x), x \in \Sigma\}$ and $\delta > 0$ is a constant.

ϵ -sample: A sample P of a shape Σ is an ϵ -sample if for each $x \in \Sigma$ there is a sample point $p \in P$ so that $\|p - x\| \leq \epsilon f(x)$.

30.1 CURVE RECONSTRUCTION

In its simplest form the reconstruction problem appeared in applications such as pattern recognition ([Chapter 51](#)), computer vision, and cluster analysis, where a curve in two dimensions is to be approximated from a set of sample points. In the 1980s several geometric graphs connecting a set of points in plane were discovered which reveal a pattern among the points. The influence graph of Toussaint [AH85], the β -skeleton of Kirkpatrick and Radke [KR85], the α -shapes of Edelsbrunner, Kirkpatrick, Seidel [EKS83] are such graphs. Recently, several algorithms have been proposed that reconstruct a curve from its sample with guarantees under some sampling assumption.

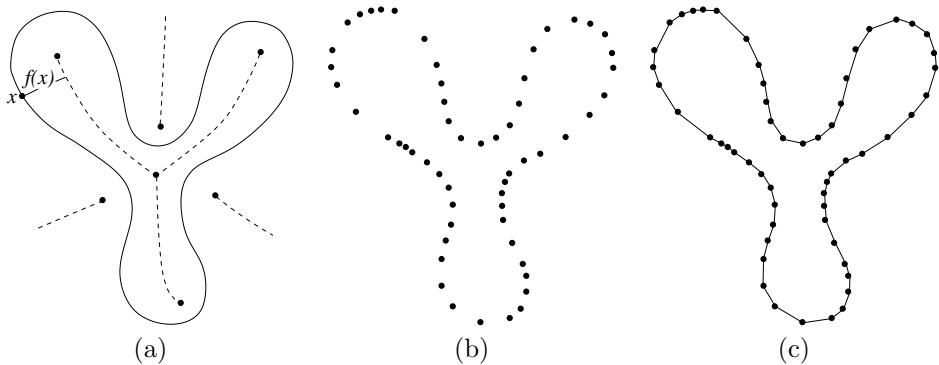


FIGURE 30.1.1
A smooth curve (solid), its medial axis (dashed) (a), sample (b), reconstruction (c).

GLOSSARY

Curve: A curve C in plane is a trace of a function $p : \mathbb{R} \rightarrow \mathbb{R}^2$ where $p(t) = (x(t), y(t))$ for $t \in [0, 1]$ and $p[t] \neq p[t']$ for any $t \neq t'$ except possibly $t, t' \in \{0, 1\}$. It is *smooth* if p is differentiable and the derivative $\frac{d}{dt}p(t) = (\frac{dx(t)}{dt}, \frac{dy(t)}{dt})$ does not vanish.

Boundary: A curve C is said to have no boundary if $p[0] = p[1]$; otherwise, it is a curve with boundary.

Reconstruction: The reconstruction of C from its sample P is a geometric graph $G = (P, E)$ where an edge pq belongs to E if and only if p and q are adjacent sample points on C . See Figure 30.1.1.

Semiregular curve: One for which the left tangent and right tangent exist at each point of the curve, though they may be different.

UNIFORM SAMPLE

α -shapes: Edelsbrunner, Kirkpatrick, and Seidel [EKS83] defined the α -shape of a point set $P \subseteq \mathbb{R}^2$ as the underlying space of a simplicial complex called the α -complex. The α -complex of P is defined by all simplices with vertices in P that have an empty circumscribing disk of radius α . Bernardini and Bajaj [BB97] show that the α -shapes reconstruct curves from δ -uniform samples if δ is sufficiently small and α is chosen appropriately.

r -regular shapes Attali considered r -regular shapes that are constructed using certain morphological operations with r as a parameter [Att97]. It turns out that these shapes are characterized by requiring that any circle passing through the points on the boundary has radius greater than r . A sample P from the boundary curve C of such a shape is called γ -sample if each point $x \in C$ has a sample point within γr distance. Let η_{pq} be the sum of the angles opposite to pq in the two incident Delaunay triangles at a Delaunay edge $pq \in D_P$. The main result in [Att97] is that if $\gamma < \sin \frac{\pi}{8}$, Delaunay edges with $\eta_{pq} < \pi$ reconstruct C .

EMST: Figueiredo and Gomes [FG95] show that the Euclidean minimum spanning tree (EMST) reconstructs curves with boundaries when the sample is sufficiently dense. The sampling density condition that is used to prove this result is equivalent to that of δ -uniform sampling for an appropriate $\delta > 0$. Of course, EMST cannot reconstruct curves without boundaries and/or multiple components.

NONUNIFORM SAMPLE

Crust: Amenta, Bern, and Eppstein [ABE98] proposed the first algorithm to reconstruct a curve from a non-uniform sample with guarantee. The algorithm computes the *crust* of P in two phases. The first phase computes the Voronoi diagram of the sample points in P . Let V be the set of Voronoi vertices in this diagram. The second phase computes the Delaunay triangulation of the larger set $P \cup V$. The Delaunay edges that connect only sample points in this triangulation constitute the crust; see [Figure 30.1.2](#).

The theoretical guarantee of the crust algorithm is based on the notion of dense sampling that respects features of the sampled curve. The important concepts of local feature size and ϵ -sampling were introduced by Amenta, Bern, and Eppstein [ABE98]. They prove that if P is an ϵ -sample of a curve C without boundary for $\epsilon \leq 0.252$, the crust reconstructs C . The two Voronoi diagram computations of the crust are reduced to one by Gold and Snoeyink [GS01].

Nearest neighbor: After the introduction of the crust, Dey and Kumar [DK99] proposed a curve reconstruction algorithm based on nearest neighbors. They showed that all nearest neighbor edges that connect a point to its Euclidean nearest neighbor must be in the reconstruction if the input is $\frac{1}{3}$ -sample. However, not all edges of the reconstruction are necessarily nearest neighbor edges. The remaining edges are characterized as follows. Let p be a sample point with only one nearest neighbor edge pq incident to it. Consider the halfplane with pq being an outward normal to its bounding line through p , and let r be the nearest to p among all sample points

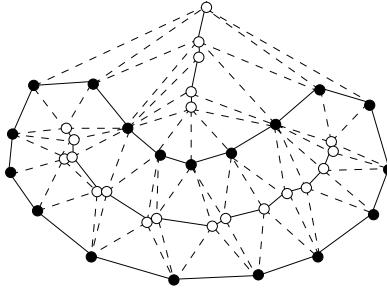


FIGURE 30.1.2

Crust edges (solid) among the Delaunay triangulation of a sample and their Voronoi vertices.

lying in this halfplane. Call pr the half-neighbor edge of p . Dey and Kumar show that all half-neighbor edges must also be in the reconstruction for a $\frac{1}{3}$ -sample.

The algorithm first computes all nearest neighbor edges and then computes the half-neighbor edges to complete the reconstruction. Since all edges in the reconstruction must be a subset of Delaunay edges if the sample is sufficiently dense, all nearest neighbor and half-neighbor edges can be computed from the Delaunay triangulation. Thus, as crust this algorithm runs in $O(n \log n)$ time for a sample of n points.

NONSMOOTHNESS, BOUNDARIES

The crust and nearest neighbor algorithms assume that the sampled curve is smooth and has no boundary. Nonsmoothness and boundaries make reconstruction harder.

Traveling Salesman Path: Giesen [Gie00] considered a fairly large class of non-smooth curves and showed that Traveling Salesman Path (or Tour) reconstructs them from sufficiently dense samples. A semiregular curve C is **benign** if the angle between the two tangents at each point is less than π . Giesen proved that, for a benign curve C , there exists a $\delta > 0$ so that if each $x \in C$ has a sample point p with $\|p - x\| \leq \delta$, then C is reconstructed by the Traveling Salesman Path (or Tour) in case C has boundary (or no boundary).

The uniform sampling condition for the Traveling Salesman approach was later removed by Althaus and Mehlhorn [AM02], who also gave a polynomial-time algorithm to compute the Traveling Salesman Path (or Tour) in this special case of curve reconstruction. Obviously, the Traveling Salesman approach cannot handle curves with multiple components. Also, the sample points representing the boundary need to be known a priori to choose between path or tour.

Conservative Crust: In order to allow boundaries in curve reconstruction, it is essential that the sample points representing boundaries are detected. Dey, Mehlhorn, and Ramos presented such an algorithm, called the *conservative crust* [DMR00].

Any algorithm for handling curves with boundaries faces a dilemma when an input point set samples a curve without boundary densely and simultaneously samples another curve with boundary densely. This dilemma is resolved in conservative

crust by a justification on the output. For any input point set P , the graph output by the algorithm is guaranteed to be the reconstruction of a smooth curve C possibly with boundary for which the input point set is a dense sample. The main idea of the algorithm is that an edge pq is chosen in the output only if there is a large enough ball centering the midpoint of pq which is empty of all Voronoi vertices in the Voronoi diagram of P . The rationale behind this choice is that these edges are small enough with respect to local feature size of C since the Voronoi vertices approximate its medial axis.

With a certain sampling condition tailored to handle nonsmooth curves, Funke, and Ramos used conservative crust to reconstruct nonsmooth curves that may have boundaries [FR01].

SUMMARIZED RESULTS

The strengths and deficiencies of the discussed algorithms are summarized in Table 30.1.1.

TABLE 30.1.1 Curve reconstruction algorithms.

ALGORITHM	SAMPLE	SMOOTHNESS	BOUNDARY	COMPONENTS
α -shape	uniform	required	none	multiple
r -regular shape	uniform	required	none	multiple
EMST	uniform	required	exactly two	single
Crust	non-uniform	required	none	multiple
Nearest neighbor	non-uniform	required	none	multiple
Traveling Salesman	non-uniform	not required	must be known	single
Conservative crust	non-uniform	required	any number	multiple

OPEN PROBLEM

All algorithms described above assume that the sampled curve does not cross itself. It is open to devise an algorithm that can reconstruct such curves under some reasonable sampling condition.

30.2 SURFACE RECONSTRUCTION

A number of surface reconstruction algorithms have been designed in different application fields in recent years. The problem appeared in medical imaging where a set of cross sections obtained via CAT scan or MRI needs to be joined with a surface. The points on the boundary of the cross sections are already joined by a polygonal curve and the output surface needs to join these curves in consecutive

cross sections. A dynamic programming based solution for two such consecutive curves was first proposed by Fuchs, Kedem, and Uselton [FKU77]. A negative result by Gitlin, O'Rourke, and Subramanian [GOS96] shows that, in general, two polygonal curves cannot be joined by a nonself-intersecting surface with only those vertices; even deciding its possibility is NP-hard. Several solutions with the addition of Steiner points have been proposed to overcome the problem, see [MSS92, BG93]. The most general version of the surface reconstruction problem does not assume any information about the input points other than their 3D coordinates, and requires a piecewise linear approximation of the surface from which the input point sample is derived; see Figure 30.2.1. In the context of computer graphics and vision, this problem has been investigated intensely in the past decade with emphasis on the practical effectiveness of the algorithms [BMR⁺99, Boi84, CL96, GKS00, HDD⁺92]. Lately, several algorithms have been designed mainly based on Voronoi/Delaunay diagrams that have theoretical guarantees. We focus mainly on them.

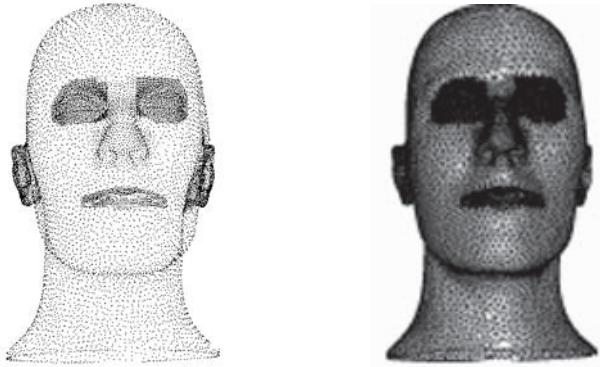


FIGURE 30.2.1
A point sample and the reconstructed surface.

GLOSSARY

Surface: A surface $S \subset \mathbb{R}^3$ is a 2-manifold embedded in \mathbb{R}^3 . Thus each point $p \in S$ has a neighborhood homeomorphic to \mathbb{R}^2 or halfplane \mathbb{H}^2 . The points with neighborhoods homeomorphic to \mathbb{H}^2 constitute the boundary of S .

Smooth Surface: A surface $S \subset \mathbb{R}^3$ is smooth if for each point $p \in S$ there is a neighborhood $W \subseteq \mathbb{R}^3$ and a map $\pi : U \rightarrow W \cap S$ of an open set $U \subset \mathbb{R}^2$ onto $W \cap S$ so that

- (i) π is differentiable,
- (ii) π is a homeomorphism,
- (iii) for each $q \in U$ the differential $d\pi_q$ is one-to-one.

Restricted Voronoi: Given a subspace $\mathbb{N} \subseteq \mathbb{R}^3$ and a point set $P \subseteq \mathbb{R}^3$, the restricted Voronoi diagram of V_P with respect to \mathbb{N} is $V_{P,\mathbb{N}} = V_P \cap \mathbb{N}$.

Restricted Delaunay: The dual of $V_{P,\text{IN}}$ is called the restricted Delaunay triangulation $D_{P,\text{IN}}$ defined as

$$D_{P,\text{IN}} = \{\sigma \mid \sigma = (p_0, \dots, p_k) \in D_P \text{ where } (\cap_{i=0,k} V_{p_i}) \cap \text{IN} \neq \emptyset\}.$$

Watertight surface: A 2-complex \mathcal{K} embedded in \mathbb{R}^3 is called watertight if the underlying space of \mathcal{K} is same as the boundary of the closure of a 3-manifold in \mathbb{R}^3 .

Steiner points: The points used by an algorithm that are not part of the finite input point set are called Steiner points.

α -SHAPES

Generalization of α -shapes to 3D by Edelsbrunner and Mücke [EM94] can be used for surface reconstruction in case the sample is more or less uniform. An alternate definition of α -shapes in terms of the restricted Delaunay triangulation is more appropriate for surface reconstruction. Let IN denote the space of all points covered by open balls of radius α around each sample point $p \in P$. The α -shape for P is the underlying space of the restricted Delaunay triangulation $D_{P,\text{IN}}$; see [Figure 30.3.1](#) below for an illustration in 2D. It is shown that the α -shape is always homotopic to IN , which in turn is homotopic to S if α is chosen appropriately for a sufficiently dense P [EM94]. Therefore, by transitivity of homotopy maps, the α -shape is homotopic to S if α is appropriate and the sample P is sufficiently dense.

The major drawback of α -shapes is that it requires a nearly uniform sample for reconstruction, and the value of α must be chosen appropriately. In a work under proprietary rights Edelsbrunner designed the WRAP algorithm based on Morse theory that overcomes the shortcoming of α -shapes [Ede03].

CRUST

The crust algorithm for curve reconstruction was generalized for surface reconstruction by Amenta and Bern [AB99]. In case of curves in 2D, Voronoi vertices for a dense sample lie close to the medial axis. That is why a second Voronoi diagram with the input sample points together with the Voronoi vertices is used to separate the Delaunay edges that reconstruct the curve. Unfortunately, Voronoi vertices in 3D can lie arbitrarily close to the sampled surface. One can place four arbitrarily close points on a smooth surface which lie near the diametric plane of the sphere defined by them. This sphere can be made empty of any other input point and thus its center as a Voronoi vertex lies close to the surface. With this important observation Amenta and Bern forsake the idea of putting all Voronoi vertices in the second phase of crust and instead identify a subset of Voronoi vertices called **poles** that lie far away from the surface, and in fact close to the medial axis.

Let P be an ϵ -sample of a compact smooth surface S without boundary. Let V_p be a Voronoi cell in the Voronoi diagram V_P . The farthest Voronoi vertex of V_p from p is called the positive pole of p . Call the vector from p to the positive pole the **pole vector** for p ; this vector approximates the surface normal \mathbf{n}_p at p . The Voronoi vertex of V_p that lies farthest from p in the opposite direction of the pole vector is called its negative pole. The opposite direction is specified by the

condition that the vector from p to the negative pole must make an angle more than $\frac{\pi}{2}$ with the pole vector. [Figure 30.2.2\(a\)](#) illustrates these definitions. If V_p is unbounded, the positive pole is taken at infinity and the direction of the pole vector is taken as the average of all directions of the unbounded Voronoi edges in V_p .

The crust algorithm in 3D proceeds as follows. First, it computes the Voronoi diagram V_P and then identifies the set of poles, say L . The Delaunay triangulation of the point set $P \cup L$ is computed and the set of Delaunay triangles, T , is filtered that have all three vertices only from P . This set of triangles almost approximates S but may not form a surface. Nevertheless, the set T includes all restricted Delaunay triangles in $D_{P,S}$. According to a result by Edelsbrunner and Shah [ES97], $D_{P,S}$ is homeomorphic to S if each Voronoi cell satisfies a topological condition called the “closed ball property.” Amenta and Bern show that if P is an ϵ -sample for $\epsilon \leq 0.06$, each Voronoi cell in V_P satisfies this property. This means that, if the triangles in $D_{P,S}$ can be extracted from T , we will have a surface homeomorphic to S . Unfortunately, it is impossible to detect the restricted Delaunay triangles of $D_{P,S}$ since S is unknown. However, the fact that T contains them is used in extracting a manifold out of T after a normal filtering step. This piecewise linear manifold surface is output as crust.

The crust guarantees that the output surface lies very close to S . In particular, each point p in the output has a point x in S so that $\|p - x\| \leq O(\epsilon)f(x)$. Also, each point x in S has a point p in the output so that the same bound holds.

COcone

The cocone algorithm was developed from the crust algorithm by Amenta, Choi, Dey, and Leekha [ACDL02]. It simplified the reconstruction algorithm and its proof of correctness.

A *cocone* C_p for a sample point p is defined as the complement of the double cone with p as apex and the pole vector as axis and an opening angle of $\frac{3\pi}{4}$; see Figure 30.2.2(b). Because the pole vector at p approximates the surface normal \mathbf{n}_p , the cocone C_p (clipped within V_p) approximates a thin neighborhood around the tangent plane at p . For each point p , the algorithm then determines all Voronoi edges in V_p that are intersected by the cocone C_p . The dual Delaunay triangles of these Voronoi edges constitute the set of candidate triangles T .

It is shown that the circumscribing circles of all candidate triangles are small [ACDL02]. Specifically, if $pqr \in T$ has circumradius r , then

$$(i) \quad r = O(\epsilon)f(x) \text{ where } f(x) = \min\{f(p), f(q), f(r)\}.$$

It turns out that any triangle with such small circumradius must lie flat to the surface, i.e., if \mathbf{n}_{pqr} is the normal to a candidate triangle pqr , then

$$(ii) \quad \angle(\mathbf{n}_{pqr}, \mathbf{n}_x) = O(\epsilon) \text{ up to orientation where } x \in \{p, q, r\}.$$

Also, it is proved that

$$(iii) \quad T \text{ includes all restricted Delaunay triangles in } D_{P,S}.$$

These three properties of the candidate triangles ensure that a manifold extraction step, as in crust algorithm, extracts a piecewise-linear surface N which is homeomorphic to the original surface S .

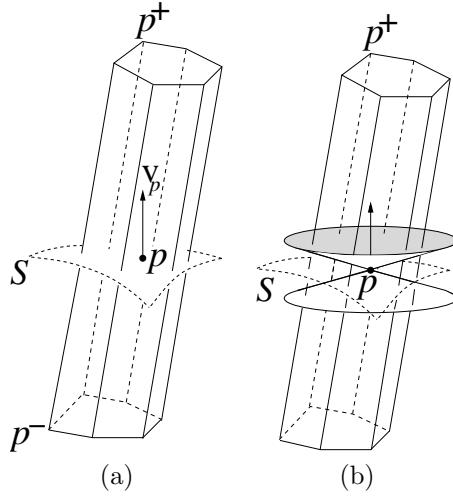


FIGURE 30.2.2

A long thin Voronoi cell V_p , the positive pole p^+ , the pole vector \mathbf{v}_p and the negative pole p^- (a), the cocone (b).

Cocone uses a single Voronoi diagram as opposed to two in the crust algorithm and also eliminates the normal filtering step. It guarantees that the output surface N is topologically equivalent to the sampled surface S for $\epsilon \leq 0.06$ and each point on N has a point x in S within $O(\epsilon)f(x)$ distance. Because of the Voronoi diagram computation, the cocone runs in $O(n^2)$ time and space. Funke and Ramos [FR02] improved its complexity to $O(n \log n)$ though the resulting algorithm seems impractical.

NATURAL NEIGHBOR

Boissonnat and Cazals [BC00] revisited the approach of Hoppe et al. [HDD⁺92] by approximating the sampled surface as the zero set of a signed distance function. They used natural neighbors and ϵ -sampling to provide output guarantees.

Given an input point set $P \subset \mathbb{R}^3$, the **natural neighbors** $N_{x,P}$ of a point $x \in \mathbb{R}^3$ are the Delaunay neighbors of x in $D_{P \cup x}$. Letting $V(x)$ denote the Voronoi cell of x in $V_{P \cup x}$, this means

$$N_{x,P} = \{p \in P \mid V(x) \cap V_p \neq \emptyset\}.$$

Let $A(x,p)$ denote the volume stolen by x from V_p , i.e.,

$$A(x,p) = V(x) \cap V_p.$$

The natural coordinate associated with a point p is a continuous function $\lambda_p : \mathbb{R}^3 \rightarrow \mathbb{R}$ where

$$\lambda_p(x) = \frac{A(x,p)}{\sum_{q \in P} A(x,q)}.$$

Some of the interesting properties of λ_p are that it is continuously differentiable except at p , and any point $x \in \mathbb{R}^3$ is a convex combination of its natural neighbors:

$\sum_{p \in N_{x,P}} \lambda_p(x)p = x$. Boissonnat and Cazals assume that each point p is equipped with a unit normal \mathbf{n}_p which can either be computed via pole vectors, or is part of the input. A distance function $h_p : \mathbb{R}^3 \rightarrow \mathbb{R}$ for each point p is defined as $h_p(x) = (p - x) \cdot \mathbf{n}_p$. A global distance function $h : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined by interpolating these local distance functions with natural coordinates. Specifically,

$$h(x) = \sum_{p \in P} \lambda_p^{1+\delta}(x) h_p(x).$$

The δ term in the exponent is added to make h continuously differentiable. By definition, $h(x)$ locally approximates the signed distance from the tangent plane at each point $p \in P$ and, in particular, $h(p) = 0$.

Since h is continuously differentiable, $\hat{S} = h^{-1}(0)$ is a smooth surface unless 0 is a critical value. A discrete approximation of \hat{S} can be computed from the Delaunay triangulation of P as follows. All Voronoi edges that intersect \hat{S} are computed via the sign of h at their two endpoints. The dual Delaunay triangles of these Voronoi edges constitute a piecewise linear approximation of \hat{S} . If the input sample P is an ϵ -sample for sufficiently small ϵ , then it can be shown that the output surface is geometrically close and is also topologically equivalent to the sampled surface.

UNDERSAMPLING

The assumption of ϵ -sampling for sufficiently small $\epsilon > 0$ often does not hold in practice. This undersampling may be caused by inadequate attention during the sampling process, or machine error, or nonsmoothness. Even boundaries in a surface may be viewed as the demarcation where appropriate sampling stops and the undersampling begins. Dey and Giesen [DG01] took this unified view to detect boundaries that identify the regions of undersampling.

Given a sample P of a surface S , the subset $S' \subseteq S$ is called **well-sampled** if each point x in S' has a sample point within $\epsilon f(x)$ distance. If P undersamples S , the well-sampled surface S' has boundaries. A point $p \in P$ is a **boundary sample** if V_p intersects the boundary of S' , otherwise p is *interior*. The algorithm of Dey and Giesen [DG01] works in two phases to detect all boundary samples. In the first phase, it selects a set $R \subseteq P$ based on two conditions. The first condition requires the Voronoi cell of a point in R be long and thin and the second requires its pole vector agree with those of all its neighbors on the surface (determined by cocones). These two conditions ensure that R consists of interior points only. In a second phase, the set R is expanded to include more points by relaxing the second condition. It is proved that under some mild assumptions on sampling, this algorithm determines all interior points and the remaining points are correctly detected as boundary.

Once the boundary samples are detected, the cocone algorithm is employed to filter the candidate triangles. Boundary samples are not allowed to choose any triangle. This produces the boundaries at the undersampled regions.

WATERTIGHT SURFACES

Most of the surface reconstruction algorithms face a difficulty while dealing with undersampled surfaces and noise. While the algorithm of [DG01] can detect undersampling, it leaves holes in the surface near the vicinity of undersampling. Although

this may be desirable for reconstructing surfaces with boundaries, many applications such as CAD designs require that the output surface be *watertight*, i.e., a surface that bounds a solid.

The natural neighbor algorithm of [BC00] can be adapted to guarantee a watertight surface. Recall that this algorithm approximates a surface \hat{S} implicitly defined by the zero set of a smooth map $h : \mathbb{R}^3 \rightarrow \mathbb{R}$. This surface is a smooth 2-manifold without boundary in \mathbb{R}^3 . However, if the input sample P is not dense for this surface, the reconstructed output may not be watertight. Boissonnat and Cazals suggest to sample more points on \hat{S} to obtain a dense sample for \hat{S} and then reconstruct it from the new sample.

Amenta, Choi, and Kolluri [ACK01] use the crust approach to design the *power crust* algorithm to produce watertight surfaces. This algorithm first distinguishes the inner poles that lie inside the solid bounded by the sampled surface S from the outer poles that lies outside. A consistent orientation of the pole vectors is used to decide between inner and outer poles. To prevent outer poles at infinity, eight corners of a large box containing the sample are added. The union of Delaunay balls with centers at the inner poles approximate the solid bounded by S . The union of Delaunay balls centered at the outer poles do not approximate the entire exterior of S although one of its boundary component approximates S . The implication is that the cells in the power diagram of the poles with the radius of the Delaunay ball as weights can be partitioned into two sets, with the boundary between approximating S . The facets in the power diagram that separate cells generated by inner poles from the ones generated by outer poles form this boundary which is output by power crust.

Recently Dey and Goswami [DG02] announced a water-tight surface reconstructor called *tight cocone*. This algorithm first computes the surface with cocone. Recall that cocone may leave some holes in the surface due to undersampling. A subsequent sculpting [Boi84] in the Delaunay triangulation of the input points recover triangles that fill the holes. Unlike power crust, tight cocone does not add Steiner points.

SUMMARIZED RESULTS

The properties of the above discussed surface reconstruction algorithms are summarized in [Table 30.2.1](#).

OPEN PROBLEMS

All guarantees given by various surface reconstruction algorithms depend on the notion of dense sampling. Watertight surface algorithms can guarantee a surface without holes, but no theoretical guarantees exists under any type of undersampling.

1. Design an algorithm that reconstructs nonsmooth surfaces under reasonable sampling conditions.
2. Design a surface reconstruction algorithm that handles noise gracefully, and with some guarantees.

TABLE 30.2.1 Surface reconstruction algorithms.

ALGORITHM	SAMPLE	PROPERTIES	SOURCE
α -shape	uniform	α to be determined.	[EM94]
Crust	non-uniform	Theoretical guarantees from Voronoi structures, two Voronoi computations.	[AB99]
Cocone	non-uniform	Simplifies crust, single Voronoi computation with topological guarantee, detects undersampling.	[ACDL02] [DG01]
Natural Neighbor	non-uniform	Theoretical guarantees using Voronoi diagram and implicit functions.	[BC00]
Power Crust	non-uniform	Watertight surface using power diagrams, introduces Steiner points.	[ACK01]
Tight Cocone	non-uniform	Watertight surface using Delaunay triangulation.	[DG02]

30.3 SHAPE RECONSTRUCTION

All algorithms discussed above are designed for reconstructing a shape of specific dimension from the samples. Thus, the curve reconstruction algorithms cannot handle samples from surfaces and the surface reconstruction algorithms cannot handle samples from curves. Therefore, if a sample is derived from shapes of mixed dimensions, i.e., both curves and surfaces in \mathbb{R}^3 , none of the curve and surface reconstruction algorithms is adequate. General shape reconstruction algorithms should be able to handle any shape embedded in Euclidean spaces. However, this goal may be too ambitious, as it is not clear what would be a reasonable definition of dense samples for general shapes that are nonsmooth or nonmanifold. The ϵ -sampling condition would require infinite sampling in these cases. We therefore distinguish two cases: (i) smooth manifold reconstruction for which a computable sampling criterion can be defined, (ii) shape reconstruction for which it is currently unclear how a computable sampling condition could be defined to guarantee reconstruction. This leads to a different definition for the general shape reconstruction problem in the glossary below.

GLOSSARY

Shape reconstruction: Given a set of points $P \subseteq \mathbb{R}^d$, compute a shape that best approximates P .

Manifold shape: A manifold shape is a collection of smooth manifolds $\{M_1, M_2, \dots, M_\ell\}$ embedded in an Euclidean space \mathbb{R}^d .

Manifold reconstruction: Compute a piecewise-linear approximation to each M_i , given a sample P from a manifold shape $\{M_1, M_2, \dots, M_\ell\}$.

MAIN RESULTS

Shape reconstruction: Not many algorithms are known to reconstruct shapes. The definition of α -shapes is general enough to be applicable to shape reconstruction. In Figure 30.3.1, the α -shape reconstructs a shape in \mathbb{R}^2 which is not a manifold. Similarly, it can reconstruct curves, surfaces and solids and their combinations in three dimensions. Melkemi [Mel97] proposed \mathcal{A} -shapes that can reconstruct shapes in \mathbb{R}^2 . Its class of shapes includes α -shapes. Given a set of points P in \mathbb{R}^2 , a member in this class of shapes is identified with another finite set $\mathcal{A} \subseteq \mathbb{R}^2$. The \mathcal{A} -shape of S is generated by edges that connect points $p, q \in P$ if there is a circle passing through p, q and a point in \mathcal{A} , and all other points in $P \cup \mathcal{A}$ lie outside the circle. The α -shape is a special case of \mathcal{A} -shapes where \mathcal{A} is the set of all points on Voronoi edges that span empty circles with points in P . The crust is also a special case of \mathcal{A} -shape where \mathcal{A} is the set of Voronoi vertices.

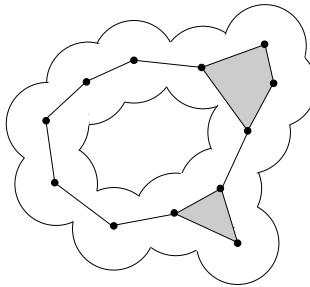


FIGURE 30.3.1
Alpha shape of a set of points in \mathbb{R}^2 .

Manifold reconstruction: When the sample P derives from smooth manifolds embedded in some Euclidean space \mathbb{R}^d , Dey et al. [DGGZ02] propose an algorithm *CoconeShape* for reconstruction. This algorithm first determines the dimension k of a sample point $p \in P$ if p is derived from a k -manifold. This dimension detection is accomplished by analyzing the structure of the Voronoi cell V_p . Subsequent to the dimension detection, a subset of k -dimensional Delaunay simplices incident to p are chosen in an output set T . This computation is performed with a generalized concept of cocones.

It is shown that the underlying space of T lies very close to the sampled manifold(s) although it may not be a triangulation of a manifold. A manifold extraction step as in the case of surfaces in \mathbb{R}^3 is necessary to clean T , but it is not clear how to do this effectively. In \mathbb{R}^2 and \mathbb{R}^3 , the manifold extraction step can be performed and hence the manifold reconstruction problem is solved in \mathbb{R}^2 and \mathbb{R}^3 .

OPEN PROBLEMS

1. Design an algorithm that outputs manifolds approximating sampled manifold shapes in \mathbb{R}^d , $d \geq 4$.
 2. Reconstruct shapes with guarantees.
-

30.4 SOURCES AND RELATED MATERIAL

SURVEYS

[Ede98]: Shape reconstruction with Delaunay complex.

[OR00]: Computational geometry column 38 (Recent results on curve reconstruction).

[MSS92]: Surfaces from contours.

[MM98]: Interpolation and approximation of surfaces from 3D scattered data points.

RELATED CHAPTERS

[Chapter 22: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 24: Triangulations and mesh generation](#)

[Chapter 28: Geometric reconstruction problems](#)

[Chapter 31: Computational topology](#)

[Chapter 49: Computer graphics](#)

[Chapter 51: Pattern recognition](#)

[Chapter 54: Surface simplification and 3D geometry compression](#)

REFERENCES

- [AM02] E. Althaus and K. Mehlhorn. Traveling salesman-based curve reconstruction in polynomial time. *SIAM J. Comput.*, 31: 27–66, 2002.
- [AB99] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22:481–504, 1999.
- [ACDL02] N. Amenta, S. Choi, T.K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Geom. Appl.*, 12:125–141, 2002.
- [ABE98] N. Amenta, M. Bern, and D. Eppstein. The crust and the β -skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60:125–135, 1998.
- [ACK01] N. Amenta, S. Choi, and R.K. Kolluri. The power crust, union of balls, and the medial axis transform. *Comput. Geom. Theory Appl.*, 19:127–153, 2001.

- [Att97] D. Attali. r -regular shape reconstruction from unorganized points. In *Proc. 13th Annu. Sympos. Comput. Geom.*, pages 248–253, 1997.
- [AH85] D. Avis and J. Horton. Remarks on the sphere of influence graph. In *Proc. Conf. Discr. Geom. Convexity*, J.E. Goodman et al., editors, *Ann. New York Acad. Sci.*, 440:323–327, 1985.
- [BB97] F. Bernardini and C.L. Bajaj. Sampling and reconstructing manifolds using α -shapes. In *Proc. 9th Canad. Conf. Comput. Geom.*, pages 193–198, 1997.
- [BMR⁺99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Visual. Comput. Graphics*, 5:349–359, 1999.
- [Boi84] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. Graphics*, 3:266–286, 1984.
- [BC00] J.-D. Boissonnat and F. Cazals. Smooth surface reconstruction via natural neighbor interpolation of distance functions. In *Proc. 16th Annu. Sympos. Comput. Geom.*, pages 223–232, 2000.
- [BG93] J.-D. Boissonnat and B. Geiger. Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation. In *Proc. Biomedical Image Process. Biomed. Visualization*, pages 964–975, 1993.
- [CL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. ACM Conf. SIGGRAPH 96*, pages 306–312, 1996.
- [DG01] T.K. Dey and J. Giesen. Detecting undersampling in surface reconstruction. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 257–263, 2001.
- [DGGZ02] T.K. Dey, J. Giesen, S. Goswami, and W. Zhao. Shape dimension and approximation from samples. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 772–780, 2002.
- [DG02] T.K. Dey and S. Goswami. Tight cocone: A water-tight surface reconstructor. In *Proc. 8th Annu. ACM Sympos. Solid Modeling Appl.*, pages 127–134, 2002.
- [DK99] T.K. Dey and P. Kumar. A simple provable curve reconstruction algorithm. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 893–894, 1999.
- [DMR00] T.K. Dey, K. Mehlhorn, and E.A. Ramos. Curve reconstruction: connecting dots with good reason. *Comput. Geom. Theory & Appl.*, 15:229–244, 2000.
- [Ede98] H. Edelsbrunner. Shape reconstruction with Delaunay complex. *LATIN 98: Theoretical Informatics, Lecture Notes Comput. Sci.*, volume 1380, pages 119–132, Springer-Verlag, Berlin, 1998.
- [Ede03] H. Edelsbrunner. Surface reconstruction by wrapping finite point sets in space. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift*, pages 379–404. Springer-Verlag, Berlin, 2003.
- [EKS83] H. Edelsbrunner, D.G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory*, 29:551–559, 1983.
- [EM94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graphics*, 13:43–72, 1994.
- [ES97] H. Edelsbrunner and N.R. Shah. Triangulating topological spaces. *Internat. J. Comput. Geom. Appl.*, 7:365–378, 1997.
- [FG95] L.H. de Figueiredo and J. de Miranda Gomes. Computational morphology of curves. *Visual Computer*, 11:105–112, 1995.

- [FKU77] H. Fuchs, Z.M. Kedem, and S.P. Uselton. Optimal surface reconstruction from planar contours. *Commun. ACM*, 20:693–702, 1977.
- [FR01] S. Funke and E.A. Ramos. Reconstructing curves with corners and endpoints. In *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 344–353, 2001.
- [FR02] S. Funke and E.A. Ramos. Smooth-surface reconstruction in near-linear time. In *13th ACM-SIAM Sympos. Discrete Algorithms*, pages 781–790, 2002.
- [Gie00] J. Giesen. Curve reconstruction, the traveling salesman problem and Menger’s theorem on length. *Discrete Comput. Geom.*, 24:577–603, 2000.
- [GOS96] C. Gitlin, J. O’Rourke, and V. Subramanian. On reconstruction of polyhedra from slices. *Internat. J. Comput. Geom. Appl.*, 6:103–112, 1996.
- [GS01] C.M. Gold and J. Snoeyink. Crust and anti-crust: A one-step boundary and skeleton extraction algorithm. *Algorithmica*, 30:144–163, 2001.
- [GKS00] M. Gopi, S. Krishnan, and C. Silva. Surface reconstruction based on lower dimensional localized Delaunay triangulation. In *Eurographics*, pages C467–C478, 2000.
- [HDD⁺92] H. Hoppe, T.D. DeRose, T. Duchamp, J. McDonald, and W. Stützle. Surface reconstruction from unorganized points. In *Proc. ACM Conf. SIGGRAPH 92*, pages 71–78, 1992.
- [KR85] D.G. Kirkpatrick and J.D. Radke. A framework for computational morphology. In G.T. Toussaint, editor, *Computational Geometry*, Elsevier North-Holland, Amsterdam, pages 217–248, 1985.
- [Mel97] M. Melkemi. \mathcal{A} -shapes of a finite point set. Correspondence in *Proc. 13th Annu. Sympos. Comput. Geom.*, 367–369, 1997.
- [MM98] R. Mencl and H. Müller. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *State of the Art Reports, Eurographics 98*, 51–67, 1998.
- [MSS92] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Trans. Graphics*, 11:228–258, 1992.
- [OR00] J. O’Rourke. Computational geometry column 38. *Internat. J. Comput. Geom. Appl.*, 10:221–223, 2000. Also in *SIGACT News*, 31:28–30 (Issue 114), 2000.

31 COMPUTATIONAL CONVEXITY

Peter Gritzmann and Victor Klee

INTRODUCTION

The subject of Computational Convexity draws its methods from discrete mathematics and convex geometry, and many of its problems from operations research, computer science, and other applied areas. In essence, it is the study of the computational and algorithmic aspects of high-dimensional convex sets (especially polytopes), with a view to applying the knowledge gained to convex bodies that arise in other mathematical disciplines or in the mathematical modeling of problems from outside mathematics.

The name *Computational Convexity* is of recent origin, having first appeared in print in 1989. However, results that retrospectively belong to this area go back a long way. In particular, many of the basic ideas of *Linear Programming* have an essentially geometric character and fit very well into the conception of Computational Convexity. The same is true of the subject of *Polyhedral Combinatorics* and of the *Algorithmic Theory of Polytopes and Convex Bodies*.

The emphasis in Computational Convexity is on problems whose underlying structure is the convex geometry of normed vector spaces of finite but generally *not* restricted dimension, rather than of fixed dimension. This leads to closer connections with the optimization problems that arise in a wide variety of disciplines. Further, in the study of Computational Convexity, the underlying model of computation is mainly the binary (Turing machine) model that is common in studies of computational complexity. This requirement is imposed by prospective applications, particularly in mathematical programming. For the study of algorithmic aspects of convex bodies that are not polytopes, the binary model is often augmented by additional devices called “oracles.” Some cases of interest involve other models of computation, but the present discussion focuses on aspects of computational convexity for which binary models seem most natural. Many of the results stated in this chapter are qualitative, in the sense that they classify certain problems as being solvable in polynomial time, or show that certain problems are NP-hard or harder. The tasks remain to find optimal exact algorithms for the problems that are polynomially solvable, and to find useful approximation algorithms or heuristics for those that are NP-hard. In most cases, the known algorithms, even when they run in polynomial time, appear to be far from optimal from the viewpoint of practical application. Hence, the qualitative complexity results should in many cases be regarded as a guide to future efforts but not as final words on the problems with which they deal.

Some of the important areas of computational convexity, such as linear and convex programming, polyhedral combinatorics, packing and covering, and pattern recognition, are covered in other chapters of this Handbook. Hence, after some remarks on presentations of polytopes in Section 31.1, the present discussion concentrates on areas that are not covered elsewhere in the Handbook. The

following sections are closely related to classical convex geometry: 31.2, Algorithmic Theory of Convex Bodies; 31.3, Volume Computations; 31.4, Mixed Volumes; 31.5, Containment Problems; 31.6, Radii. (Other such areas are geometric tomography [Gar95, GG94], discrete tomography [GG97], and the computational aspects of the following topics: projections of polytopes [Fil90, BGK96], sections of polytopes [Fil92], Minkowski addition of polytopes [GS93], and the Minkowski reconstruction of polytopes [GH99].) The final section, 31.7, Interval Matrices and Qualitative Matrices, is included as an illustration of material that, though not related to classical convex geometry, nevertheless falls under the general conception of computational convexity.

Because of the diversity of topics covered in this chapter, each section has a separate bibliography.

FURTHER READING

- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- [GK93b] P. Gritzmann and V. Klee. Mathematical programming and convex geometry. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 627–674. North-Holland, Amsterdam, 1993.
- [GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math.*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.
- [GK94b] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 373–466. Kluwer, Dordrecht, 1994.

RELATED CHAPTERS

[Chapter 7: Lattice points and lattice polytopes](#)

[Chapter 16: Basic properties of convex polytopes](#)

[Chapter 46: Mathematical programming](#)

REFERENCES

- [BGK96] T. Burger, P. Gritzmann, and V. Klee. Polytope projection and projection polytopes. *Amer. Math. Monthly*, 103:742–755, 1996.
- [Fil90] P. Filliman. Exterior algebra and projections of polytopes. *Discrete Comput. Geom.*, 5:305–322, 1990.
- [Fil92] P. Filliman. Volumes of duals and sections of polytopes. *Mathematika*, 39:67–80, 1992.
- [Gar95] R.J. Gardner. *Geometric Tomography*. Cambridge University Press, 1995.
- [GG94] R.J. Gardner and P. Gritzmann. Successive determination and verification of polytopes by their X-rays. *J. London Math. Soc.*, 50:375–391, 1994.

-
- [GG97] R.J. Gardner and P. Gritzmann. Discrete tomography: determination of finite sets by X-rays. *Trans. Amer. Math. Soc.*, 349:2271–2295, 1997.
- [GH99] P. Gritzmann and A. Hufnagel. On the algorithmic complexity of Minkowski’s reconstruction theorem. *J. London Math. Soc.* (2), 59:1081–1100, 1999.
- [GS93] P. Gritzmann and B. Sturmfels. Minkowski addition of polytopes: computational complexity and applications to Gröbner bases. *SIAM J. Discrete Math.*, 6:246–269, 1993.
-

31.1 PRESENTATIONS OF POLYTOPES

A convex polytope $P \subset \mathbb{R}^n$ can be represented in terms of its vertices or in terms of its facet inequalities. From a theoretical viewpoint, the two possibilities are equivalent. However, as the dimension increases, the number of vertices can grow exponentially in terms of the number of facets, and vice versa, so that different presentations may lead to different classifications concerning polynomial-time computability or NP-hardness. (See Sections 16.1, 17.3, and 22.3 of this Handbook.)

For algorithmic purposes it is usually not the polytope P as a *geometric* object that is relevant, but rather its *algebraic presentation*. The discussion here is based mainly on the *binary* or *Turing machine* model of computation, in which the *size of the input* is defined as the length of the binary encoding needed to present the input data to a Turing machine and the *time-complexity* of an algorithm is also defined in terms of the operations of a Turing machine. Hence the algebraic presentation of the objects at hand must be finite.

Among important special classes of polytopes, the zonotopes are particularly interesting because they can be so compactly presented.

GLOSSARY

Convex body in \mathbb{R}^n : A compact convex subset of \mathbb{R}^n .

\mathcal{K}^n : The family of all convex bodies in \mathbb{R}^n .

Proper convex body in \mathbb{R}^n : A convex body in \mathbb{R}^n with nonempty interior.

Polytope: A convex body that has only finitely many extreme points.

\mathcal{P}^n : The family of all convex polytopes in \mathbb{R}^n .

n -polytope: Polytope of dimension n .

Face of a polytope P : P itself, the empty set, or the intersection of P with some supporting hyperplane; $f_i(P)$ is the number of i -dimensional faces of P .

Facet of an n -polytope P : Face of dimension $n - 1$.

Simple n -polytope: Each vertex is incident to precisely n edges or, equivalently, to precisely n facets.

Simplicial polytope: A polytope in which each facet is a simplex.

V -presentation of a polytope P : A string $(n, m; v_1, \dots, v_m)$, where $n, m \in \mathbb{N}$ and $v_1, \dots, v_m \in \mathbb{R}^n$ such that $P = \text{conv}\{v_1, \dots, v_m\}$.

H -presentation of a polytope P : A string $(n, m; A, b)$, where $n, m \in \mathbb{N}$, A is a real $m \times n$ matrix, and $b \in \mathbb{R}^m$ such that $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

\mathcal{V} -polytope P : A string $(n, m; v_1, \dots, v_m)$, where $n, m \in \mathbb{N}$ and $v_1, \dots, v_m \in \mathbb{Q}^n$. P is usually identified with the geometric object $\text{conv}\{v_1, \dots, v_m\}$.

\mathcal{H} -polytope P : A string $(n, m; A, b)$, where $n, m \in \mathbb{N}$, A is a rational $m \times n$ matrix, $b \in \mathbb{Q}^m$, and the set $\{x \in \mathbb{R}^n \mid Ax \leq b\}$ is bounded. P is usually identified with this set.

Size of a \mathcal{V} - or an \mathcal{H} -polytope P : Number of binary digits needed to encode the string $(n, m; v_1, \dots, v_m)$ or $(n, m; A, b)$, respectively.

Zonotope: The vector sum (Minkowski sum) of a finite number of line segments; equivalently, a polytope of which each face has a center of symmetry.

S -presentation of a zonotope Z in \mathbb{R}^n : A string $(n, m; c; z_1, \dots, z_m)$, where $n, m \in \mathbb{N}$ and $c, z_1, \dots, z_m \in \mathbb{R}^n$, such that $Z = c + \sum_{i=1}^m [-1, 1]z_i$.

Parallelotope in \mathbb{R}^n : A zonotope $Z = c + \sum_{i=1}^m [-1, 1]z_i$, with z_1, \dots, z_m linearly independent.

S -zonotope Z in \mathbb{R}^n : A string $(n, m; c; z_1, \dots, z_m)$, where $n, m \in \mathbb{N}$ and $c, z_1, \dots, z_m \in \mathbb{Q}^n$. Z is usually identified with the geometric object $c + \sum_{i=1}^m [-1, 1]z_i$.

31.1.1 CONVERSION OF ONE PRESENTATION INTO THE OTHER

The following results indicate the difficulties that may be expected in converting the \mathcal{H} -presentation of a polytope into a \mathcal{V} -presentation or vice versa.

For \mathcal{H} -presented n -polytopes with m facets, the *maximum* possible number of vertices is

$$\mu(m, n) = \binom{m - \lfloor (n+1)/2 \rfloor}{m-n} + \binom{m - \lfloor (n+2)/2 \rfloor}{m-n},$$

and this is also the maximum possible number of facets for a \mathcal{V} -presented n -polytope with m vertices. The first maximum is attained within the family of simple n -polytopes, the second within the family of simplicial n -polytopes.

When n is fixed, the number of vertices is bounded by a polynomial in the number of facets, and vice versa, and it is possible to pass from either sort of presentation to the other in polynomial time. However, the degree of the polynomial goes to infinity with n . A consequence of this is that when the dimension n is permitted to vary in a problem concerning polytopes, the manner of presentation is often influential in determining whether the problem can be solved in polynomial time or is NP-hard. For the case of variable dimension, it is #P-hard even to determine the number of facets of a given \mathcal{V} -polytope, or to determine the number of vertices of a given \mathcal{H} -polytope.

For *simple* \mathcal{H} -presented n -polytopes with m facets, the *minimum* possible number of vertices is $(m-n)(n-1) + 2$. The large gap between this number and the above sum of binomial coefficients makes it clear that, from a practical standpoint, the worst-case behavior of any conversion algorithm should be measured in terms of *both* input size and output size.

The maximum number of j -dimensional faces of an n -dimensional zonotope formed as the sum of m segments is

$$2 \binom{m}{j} \sum_{k=0}^{n-1-j} \binom{m-1-j}{k},$$

and hence, the number of vertices or of facets (or of faces of any dimension) of an \mathcal{S} -zonotope is not bounded by any polynomial in the size of the \mathcal{S} -presentation.

We end this section by mentioning two other ways of presenting polytopes.

A result of Bröcker and Scheiderer (see [BCR98]) implies that for each n -polytope P in \mathbb{R}^n (no matter how complicated its facial structure may be), there exists a system of $n(n+1)/2$ polynomial inequalities that has P as its solution-set, and that n polynomial inequalities suffice to describe the interior of P . However, it is in general unknown whether one can *efficiently* produce such small polynomial presentations from the linear inequalities defining the facets of P .

For a polytope P in \mathbb{R}^n whose interior is known to contain the origin, [GKW95] shows that the entire face-lattice of P can be reconstructed with the aid of at most

$$f_0(P) + (n-1)f_{n-1}^2(P) + (5n-4)f_{n-1}(P)$$

queries to the *ray-oracle* of P . In each such query, one specifies a ray issuing from the origin and the oracle is required to tell where the ray hits the boundary of P . Related results were obtained in [DEY90].

For more on oracles, see [Section 31.2](#) of this Handbook.

FURTHER READING

- [BL93] M. Bayer and C. Lee. Combinatorial aspects of convex polytopes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 251–305. North-Holland, Amsterdam, 1993.
- [BCR98] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*. Springer-Verlag, Berlin, 1998.
- [Brø83] A. Brøndsted. *An Introduction to Convex Polytopes*. Springer-Verlag, New York, 1983.
- [GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math.*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.
- [Grü67] B. Grünbaum. *Convex Polytopes*. Wiley-Interscience, London, 1967. (Second edition prepared in collaboration with V. Kaibel, V. Klee, and G. Ziegler, Springer-Verlag, New York, 2003.)
- [KK95] V. Klee and P. Kleinschmidt. Convex polytopes and related complexes. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, Volume I, pages 875–917. North-Holland, Amsterdam, 1995.
- [MS71] P. McMullen and G.C. Shephard. *Convex Polytopes and the Upper Bound Conjecture*. Cambridge University Press, 1971.
- [Zie94] G.M. Ziegler. *Lectures on Polytopes*. Volume 152 of *Graduate Texts in Math.*, Springer-Verlag, New York, 1995.

RELATED CHAPTERS

- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 18: Face numbers of polytopes and complexes](#)
- [Chapter 22: Convex hull computations](#)

REFERENCES

- [DEY90] D.P. Dobkin, H. Edelsbrunner, and C. Yap. Probing convex polytopes. In I.J. Cox and T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, New York, pages 328–341, 1990.
- [GKW95] P. Gritzmann, V. Klee, and J. Westwater. Polytope containment and determination by linear probes. *Proc. London Math. Soc.*, 70:691–720, 1995.
-

31.2 ALGORITHMIC THEORY OF CONVEX BODIES

Polytopes may be \mathcal{V} -presented or \mathcal{H} -presented. However, a different approach is required to deal with convex bodies K that are not polytopes, since an enumeration of all the extreme points of K or of its polar is not possible. A convenient way to deal with the general situation is to assume that the convex body in question is given by an algorithm (called an *oracle*) that answers certain sorts of questions about the body. A small amount of a priori information about the body may be known, but aside from this, all information about the specific convex body must be obtained from the oracle, which functions as a “black box.” In other words, while it is assumed that the oracle’s answers are always correct, nothing is assumed about the manner in which it produces those answers. The algorithmic theory of convex bodies was developed in [GLS88] with a view to proper (i.e., n -dimensional) convex bodies in \mathbb{R}^n . For many purposes, provisions can be made to deal meaningfully with improper bodies as well, but that aspect is largely ignored in what follows.

GLOSSARY

Outer parallel body of a convex body K : $K(\epsilon) = K + \epsilon B^n$, where B^n is the Euclidean unit ball in \mathbb{R}^n .

Inner parallel body of a convex body K : $K(-\epsilon) = K \setminus ((\mathbb{R}^n \setminus K) + \epsilon B^n)$.

Weak membership problem for a convex body K in \mathbb{R}^n : Given $y \in \mathbb{Q}^n$, and a rational number $\epsilon > 0$, conclude with one of the following: *assert that* $y \in K(\epsilon)$; or *assert that* $y \notin K(-\epsilon)$.

Weak separation problem for a convex body K in \mathbb{R}^n : Given a vector $y \in \mathbb{Q}^n$, and a rational number $\epsilon > 0$, conclude with one of the following: *assert that* $y \in K(\epsilon)$; or *find a vector* $z \in \mathbb{Q}^n$ *such that* $\|z\|_\infty = 1$ and $z^T x < z^T y + \epsilon$ *for every* $x \in K(-\epsilon)$.

Weak (linear) optimization problem for a convex body K in \mathbb{R}^n : Given a vector $c \in \mathbb{Q}^n$ and a rational number $\epsilon > 0$, conclude with one of the following: *find a vector* $y \in \mathbb{Q}^n \cap K(\epsilon)$ *such that* $c^T x \leq c^T y + \epsilon$ *for every* $x \in K(-\epsilon)$; or *assert that* $K(-\epsilon) = \emptyset$.

Circumscribed convex body K : A positive rational number R is given explicitly such that $K \subset RB^n$.

Well-bounded convex body K : Positive rational numbers r, R are given explicitly such that $K \subset RB^n$ and K contains a ball of radius r .

Centered well-bounded convex body K : Positive rational numbers r, R and a vector $b \in \mathbb{Q}^n$ are given explicitly such that $b + rB^n \subset K$ and $K \subset RB^n$.

Weak membership oracle for a convex body K : Algorithm that solves the weak membership problem for K .

Weak separation oracle for K : Algorithm that solves the weak separation problem for K .

Weak (linear) optimization oracle for K : Algorithm that solves the weak (linear) optimization problem for K .

The three problems above are very closely related in the sense that when the classes of proper convex bodies are appropriately restricted to those that are circumscribed, well-bounded, or centered, and when input sizes are properly defined, an algorithm that solves any one of the problems in polynomial time can be used as a subroutine to solve the others in polynomial time also. The definition of input size involves the size of ϵ , the dimension of K , the given a priori information ($\text{size}(r)$, $\text{size}(R)$, and/or $\text{size}(b)$), and the input required by the oracle. The following theorem of [GLS88] contains a list of the precise relationships among the three basic oracles for proper convex bodies. The notation “ $(\mathcal{A}; \text{prop}) \rightarrow_{\pi} \mathcal{B}$ ” indicates the existence of an (oracle-) polynomial-time algorithm that solves problem \mathcal{B} for every proper convex body that is given by the oracle \mathcal{A} and has all the properties specified in prop . ($\text{prop} = \emptyset$ means that the statement holds for general proper convex bodies.)

(WEAK MEMBERSHIP; centered, well-bounded) \rightarrow_{π} WEAK SEPARATION;
 (WEAK MEMBERSHIP; centered, well-bounded) \rightarrow_{π} WEAK OPTIMIZATION;
 (WEAK SEPARATION; \emptyset) \rightarrow_{π} WEAK MEMBERSHIP;
 (WEAK SEPARATION; circumscribed) \rightarrow_{π} WEAK OPTIMIZATION;
 (WEAK OPTIMIZATION; \emptyset) \rightarrow_{π} WEAK MEMBERSHIP;
 (WEAK OPTIMIZATION; \emptyset) \rightarrow_{π} WEAK SEPARATION.

It should be emphasized that there are polynomial-time algorithms that, accepting as input a set P that is a proper \mathcal{V} -polytope, a proper \mathcal{H} -polytope, or a proper \mathcal{S} -zonotope, produce membership, separation, and optimization oracles for P , and also compute a lower bound on the inradius of P , an upper bound on its circumradius, and a “center” b_P for P . This implies that if an algorithm performs certain tasks for convex bodies given by some of the above (appropriately specified) oracles, then the same algorithm can also serve as a basis for procedures that perform these tasks for \mathcal{V} - or \mathcal{H} -polytopes and for \mathcal{S} -zonotopes. Hence the oracular framework, in addition to being applicable to convex bodies that are not polytopes, serves also to modularize the approach to algorithmic aspects of polytopes. On the other hand, there are lower bounds on the performance of approximate algorithms for the oracle model that do not carry over to the case of \mathcal{V} - or \mathcal{H} -polytopes or \mathcal{S} -zonotopes [BF87, BGK⁺03].

FURTHER READING

- [GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988, 1993.

RELATED CHAPTERS

Chapter 7: Lattice points and lattice polytopes

REFERENCES

- [BF87] I. Bárány and Z. Füredi. Computing the volume is difficult. *Discrete Comput. Geom.*, 2:319–326, 1987.
- [BGK⁺03] A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovász, and M. Simonovits. Deterministic and randomized polynomial-time approximation of radii. *Mathematika*, 48:63–105, 2001.
-

31.3 VOLUME COMPUTATIONS

It may be fair to say that the modern study of volume computations began with Kepler [Kep15] who derived the first *cubature formula* for measuring the capacities of wine barrels, and that it was the task of volume computation that motivated the general field of integration. The problem of computing or approximating volumes of convex bodies is certainly one of the basic problems in mathematics.

GLOSSARY

In the following, G is a subgroup of the group of all affine automorphisms of \mathbb{R}^n .

Dissection of an n -polytope P into n -polytopes P_1, \dots, P_k : $P = P_1 \cup \dots \cup P_k$, where the polytopes P_i have pairwise disjoint interiors.

Polytopes $P, Q \subset \mathbb{R}^n$ are **G -equidissectable**: For some k there exist dissections P_1, \dots, P_k of P and Q_1, \dots, Q_k of Q , and elements g_1, \dots, g_k of G , such that $P_i = g_i(Q_i)$ for all i .

Polytopes $P, Q \subset \mathbb{R}^n$ are **G -equicomplementable**: There are polytopes P_1, P_2 and Q_1, Q_2 such that P_2 is dissected into P and P_1 , Q_2 is dissected into Q and Q_1 , P_1 and Q_1 are G -equidissectable, and P_2 and Q_2 are G -equidissectable.

Decomposition of a set S : $S = S_1 \cup \dots \cup S_k$, where the sets S_i are pairwise disjoint.

Sets S, T are **G -equidecomposable**: For some k there are decompositions S_1, \dots, S_k of S and T_1, \dots, T_k of T , and elements g_1, \dots, g_k of G , such that $S_i = g_i(T_i)$ for all i .

Valuation on a family \mathcal{S} of subsets of \mathbb{R}^n : A functional $\varphi : \mathcal{S} \rightarrow \mathbb{R}$ with the property that $\varphi(S_1) + \varphi(S_2) = \varphi(S_1 \cup S_2) + \varphi(S_1 \cap S_2)$ whenever the sets $S_1, S_2, S_1 \cup S_2, S_1 \cap S_2 \in \mathcal{S}$.

G -invariant valuation φ : $\varphi(S) = \varphi(g(S))$ for all $S \in \mathcal{S}$ and $g \in G$.

Simple valuation φ : $\varphi(S) = 0$ whenever $S \in \mathcal{S}$ and S is contained in a hyperplane.

Monotone valuation φ : $\varphi(S_1) \leq \varphi(S_2)$ whenever $S_1, S_2 \in \mathcal{S}$ with $S_1 \subset S_2$.

Class \mathcal{P} of \mathcal{H} -polytopes is **near-simplicial**: There is a nonnegative integer σ such that $\mathcal{P} = \bigcup_{n \in \mathbb{N}} \mathcal{P}_{\mathcal{H}}(n, \sigma)$, where $\mathcal{P}_{\mathcal{H}}(n, \sigma)$ is the family of all n -dimensional \mathcal{H} -polytopes P in \mathbb{R}^n such that each facet of P has at most $n + 1 + \sigma$ vertices.

Class \mathcal{P} of \mathcal{V} -polytopes is **near-simple**: There is a nonnegative integer τ such that $\mathcal{P} = \bigcup_{n \in \mathbb{N}} \mathcal{P}_{\mathcal{V}}(n, \tau)$, where $\mathcal{P}_{\mathcal{V}}(n, \tau)$ is the family of all n -dimensional \mathcal{V} -polytopes P in \mathbb{R}^n such that each vertex of P is incident to at most $n + \tau$ edges.

Class \mathcal{P} of \mathcal{S} -zonotopes is **near-parallelotopal**: There is a nonnegative integer ζ such that $\mathcal{Z} = \bigcup_{n \in \mathbb{N}} \mathcal{Z}_{\mathcal{S}}(n, \zeta)$, where $\mathcal{Z}_{\mathcal{S}}(n, \zeta)$ is the family of all \mathcal{S} -zonotopes in \mathbb{R}^n that are represented as the sum of at most $n + \zeta$ segments.

V: The functional that associates with a convex body K its volume.

H-VOLUME: For a given \mathcal{H} -polytope P and a nonnegative rational ν , decide whether $V(P) \leq \nu$.

V-VOLUME, S-VOLUME: Similarly for \mathcal{V} -polytopes and \mathcal{S} -zonotopes.

λ -APPROXIMATION for some functional ρ : Given a positive integer n and a well-bounded convex body K given by a weak separation oracle, determine a positive rational μ such that

$$\frac{\rho(K)}{\mu} \leq 1 + \lambda \quad \text{and} \quad \frac{\mu}{\rho(K)} \leq 1 + \lambda.$$

EXPECTED VOLUME COMPUTATION: Given a positive integer n , a centered well-bounded convex body K in \mathbb{R}^n given by a weak membership oracle, and positive rationals β and ϵ . Determine a positive rational random variable μ such that

$$\text{prob} \left\{ \left| \frac{\mu}{V(K)} - 1 \right| \leq \epsilon \right\} \geq 1 - \beta.$$

31.3.1 CLASSICAL BACKGROUND, CHARACTERIZATIONS

The results in this subsection connect the subject matter of volume computation with related “classical” problems. In the following, G is a group of affine automorphisms of \mathbb{R}^n , as above, and D is the group of isometries.

- (i) Two polytopes are G -equidissectable if and only if they are G -equicomplementable.
- (ii) Two polytopes P and Q are G -equidissectable if and only if $\varphi(P) = \varphi(Q)$ for all G -invariant simple valuations on \mathcal{P}^n .
- (iii) Two plane polygons are of equal area if and only if they are D -equidissectable.
- (iv) If one agrees that an a -by- b rectangle should have area ab , and also agrees that the area function should be a D -invariant simple valuation, it then follows from the preceding result that the area of any plane polygon P can be determined (at least in theory) by finding a rectangle R to which P is equidissectable. This provides a satisfactorily geometric theory of area that does not require any limiting considerations. The third problem of Hilbert [Hil00] asked, in effect, whether such a result extends to 3-polytopes. A negative answer was supplied by [Deh00], who showed that a regular tetrahedron and a cube of the same volume are not D -equidissectable.

- (v) If P and Q are n -polytopes in \mathbb{R}^n , then for P and Q to be equidissectable under the group of all isometries of \mathbb{R}^n , it is necessary that $f^*(P) = f^*(Q)$ for each additive real function f such that $f(\pi) = 0$, where $f^*(P)$ is the so-called *Dehn invariant* of P associated with f . The condition is also sufficient for equidissectability when $n \leq 4$, but the matter of sufficiency is unsettled for $n \geq 5$.
- (vi) Two plane polygons are of equal area if and only if they are D -equidecomposable.
- (vii) In [Lac90], it was proved that any two plane polygons of equal area are equidecomposable under the group of translations. That paper also settled Tarski's old problem of "squaring the circle" by showing that a square and a circular disk of equal area are equidecomposable; there too, translations suffice. On the other hand, a disk and a square cannot be *scissors congruent*, i.e., there is no equidissection (with respect to rigid motions) into pieces that, roughly speaking, could be cut out with a pair of scissors.
- (viii) If X and Y are bounded subsets of \mathbb{R}^n (with $n \geq 3$), and each set has nonempty interior, then X and Y are D -equidecomposable. This is the famous *Banach-Tarski paradox*.
- (ix) Under the group of all volume-preserving affinities of \mathbb{R}^n , two n -polytopes are equidissectable if and only if they are of equal volume.
- (x) If φ is a translation-invariant, nonnegative, simple valuation on \mathcal{P}^n (resp. \mathcal{K}^n), then there exists a nonnegative real α such that $\varphi = \alpha V$.
- (xi) A translation-invariant valuation on \mathcal{P}^n that is homogeneous of degree n is a constant multiple of the volume.
- (xii) A continuous, rigid-motion-invariant, simple valuation on \mathcal{K}^n is a constant multiple of the volume.
- (xiii) A nonnegative simple valuation on \mathcal{P}^n (resp. \mathcal{K}^n) that is invariant under all volume-preserving linear maps of \mathbb{R}^n is a constant multiple of the volume.

31.3.2 SOME VOLUME FORMULAS

Since simplex volumes can be computed so easily, the most natural approach to the problem of computing the volume of a polytope P is to produce a *triangulation* of P (see [Chapter 17](#)). Then compute the volumes of the individual simplices and add them up to find the volume of P . (This uses the fact that the volume is a simple valuation.) As a simple consequence, one sees that when the dimension n is fixed, the volume of \mathcal{V} -polytopes and of \mathcal{H} -polytopes can be computed in polynomial time.

Another equally natural method is to dissect P into pyramids with common apex over its facets. Since the volume of such a pyramid is just $1/n$ times the product of its height and the $(n-1)$ -volume of its base, the volume can be computed recursively.

Another approach that has become a standard tool for many algorithmic questions in geometry is the *sweep-plane* technique. The general idea is to "sweep" a hyperplane through a polytope P , keeping track of the changes that occur when the

hyperplane sweeps through a vertex. As applied to volume computation, this leads to the volume formula given below that does not explicitly involve triangulations, [BN83, Law91].

Suppose that $(n, m; A, b)$ is an irredundant \mathcal{H} -presentation of a simple polytope P (see [Section 19.2](#)). Let $b = (\beta_1, \dots, \beta_m)^T$ and denote the row-vectors of A by a_1^T, \dots, a_m^T . Let $M = \{1, \dots, m\}$ and for each nonempty subset I of M , let A_I denote the submatrix of A formed by rows with indices in I and let b_I denote the corresponding right-hand side. Let $\mathcal{F}_0(P)$ denote the set of all vertices of the polytope $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. For each $v \in \mathcal{F}_0(P)$, there is a set $I = I_v \subset M$ of cardinality n such that $A_I v = b_I$ and $A_{M \setminus I} v \leq b_{M \setminus I}$. Since P is assumed to be simple and its \mathcal{H} -presentation to be irredundant, the set I_v is unique.

Let $c \in \mathbb{R}^n$ be such that $\langle c, v_1 \rangle \neq \langle c, v_2 \rangle$ for any pair of vertices v_1, v_2 that form an edge of P . Then it turns out that

$$V(P) = \frac{1}{n!} \sum_{v \in \mathcal{F}_0(P)} \frac{\langle c, v \rangle^n}{\prod_{i=1}^n e_i^T A_{I_v}^{-1} c |\det(A_{I_v})|}.$$

The ingredients of this volume formula are those that are computed in the (dual) simplex algorithm. More precisely, $\langle c, v \rangle$ is just the value of the objective function at the current basic feasible solution v , $\det(A_{I_v})$ is the determinant of the current basis, and $A_{I_v}^{-1} c$ is the vector of reduced costs, i.e., the (generally infeasible) dual point that belongs to v .

For practical computations, this volume formula has to be combined with some vertex enumeration technique. Its closeness to the simplex algorithm suggests the use of a *reverse search* method [AF92], which is based on the simplex method with Bland's pivoting rule.

As it stands, the volume formula does not involve triangulation. However, when interpreted in a polar setting, it is seen to involve the faces of the simplicial polytope P° that is the polar of P . Accordingly, generalization to nonsimple polytopes involves polar triangulation. In fact, for general polytopes P , one may apply a “lexicographic rule” for moving from one basis to another, but this amounts to a particular triangulation of P° .

Another possibility for computing the volume of a polytope P is to study the *exponential integral* $\int_P e^{\langle c, x \rangle} dx$, where c is an arbitrary vector of \mathbb{R}^n ; see [Bar93]. (Note that for $c = 0$, this integral just gives the volume of P .) Exponential integrals satisfy certain relations that make it possible to compute the integrals efficiently in some important cases. In particular, exponential sums can be used to obtain the tractability result for near-simple \mathcal{V} -polytopes stated in the next subsection.

31.3.3 TRACTABILITY RESULTS

The volume of a polytope P can be computed in polynomial time in the following cases:

- (i) when the dimension is fixed and P is a \mathcal{V} -polytope, an \mathcal{H} -polytope, or an \mathcal{S} -zonotope;
- (ii) when the dimension is part of the input and P is a near-simple \mathcal{V} -polytope, a near-simplicial \mathcal{H} -polytope, or a near-parallelotopal \mathcal{S} -zonotope.

31.3.4 INTRACTABILITY RESULTS

- (i) There is no polynomial-space algorithm for exact computation of the volume of \mathcal{H} -polytopes.
 - (ii) \mathcal{H} -VOLUME is $\#P$ -hard even for the intersections of the unit cube with one rational halfspace.
 - (iii) \mathcal{H} -VOLUME is $\#P$ -hard in the strong sense. (This follows from the result of [BW92] that the problem of computing the number of linear extensions of a given partially ordered set $\mathcal{O} = (\{1, \dots, n\}, \prec)$ is $\#P$ -complete, in conjunction with the fact that this number is equal to $n!V(P_{\mathcal{O}})$, where the set $P_{\mathcal{O}} = \{x = (\xi_1, \dots, \xi_n)^T \in [0, 1]^n \mid \xi_i \leq \xi_j \iff i \prec j\}$ is the *order polytope* of \mathcal{O} [Sta86].)
 - (iv) The problem of computing the volume of the convex hull of the regular \mathcal{V} -cross-polytope and an additional integer vector is $\#P$ -hard.
 - (v) \mathcal{S} -VOLUME is $\#P$ -hard.
-

31.3.5 DETERMINISTIC APPROXIMATION

- (i) There exists an oracle-polynomial-time algorithm that, for any convex body K of \mathbb{R}^n given by a weak optimization oracle, and for each $\epsilon > 0$, finds rationals μ_1 and μ_2 such that

$$\mu_1 \leq V(K) \leq \mu_2 \quad \text{and} \quad \mu_2 \leq n!(1 + \epsilon)^n \mu_1.$$

- (ii) Suppose that

$$\lambda(n) < \left(\frac{n}{\log n}\right)^{n/2} - 1 \quad \text{for all } n \in \mathbb{N}.$$

Then there exists no deterministic oracle-polynomial-time algorithm for λ -APPROXIMATION of the volume [BF87].

31.3.6 RANDOMIZED ALGORITHMS

[DFK89] proved that there is a randomized algorithm for EXPECTED VOLUME COMPUTATION that runs in time that is oracle-polynomial in n , $1/\epsilon$, and $\log(1/\beta)$.

The first step is a rounding procedure, using an algorithmic version of John's theorem; see Section 31.5.4. For the second step, one may therefore assume that $B^n \subset K \subset (n+1)\sqrt{n}B^n$. Now, let

$$k = \left\lceil \frac{3}{2}(n+1)\log(n+1) \right\rceil, \quad \text{and} \quad K_i = K \cap \left(1 + \frac{1}{n}\right)^i B^n \quad \text{for } i = 0, \dots, k.$$

Then it suffices to estimate each ratio $V(K_i)/V(K_{i-1})$ up to a relative error of order $\epsilon/(n \log n)$ with error probability of order $\beta/(n \log n)$.

The main step of the algorithm of [DFK89] is based on a method for sampling nearly uniformly from within certain convex bodies K_i . It superimposes a chessboard grid of small cubes (say of edge length δ) on K_i , and performs a random

walk over the set \mathcal{C}_i of cubes in this grid that intersect a suitable parallel body $K_i + \alpha B^n$, where α is small. This walk is performed by moving through a facet with probability $1/f_{n-1}(C_n) = (2n)^{-1}$ if this move ends up in a cube of \mathcal{C}_i , and staying at the current cube if the move would lead outside of \mathcal{C}_i . The random walk gives a *Markov chain* that is irreducible (since the moves are connected), aperiodic, and hence ergodic. But this implies that there is a unique stationary distribution, the limit distribution of the chain, which is easily seen to be a *uniform distribution*. Thus after a sufficiently large (but polynomially bounded) number of steps, the current cube in the random walk can be used to sample nearly uniformly from \mathcal{C}_i . Having obtained such a uniformly sampled cube, one determines whether it belongs to \mathcal{C}_{i-1} or to $\mathcal{C}_i \setminus \mathcal{C}_{i-1}$.

Now note that if ν_i is the number of cubes in \mathcal{C}_i , then the number $\mu_i = \nu_i/\nu_{i-1}$ is an estimate for the volume ratio $V(K_i)/V(K_{i-1})$. It is this number μ_i that can now be “randomly approximated” using the approximation constructed above of a uniform sampling over \mathcal{C}_i . In fact, a cube C that is reached after sufficiently many steps in the random walk will lie in \mathcal{C}_{i-1} with probability approximately $1/\mu_i$; hence this probability can be approximated closely by repeated sampling.

This algorithm has been improved by various authors; [KLS98] achieved a bound where n enters only to the fifth power.

FURTHER READING

- [Bol78] V.G. Boltyanskii. *Hilbert’s Third Problem* (Transl. by R. Silverman). Winston, Washington, 1978.
- [GW89] R.J. Gardner and S. Wagon. At long last, the circle has been squared. *Notices Amer. Math. Soc.*, 36:1338–1343, 1989.
- [GK94b] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 373–466. Kluwer, Dordrecht, 1994.
- [Had57] H. Hadwiger. *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie*. Springer-Verlag, Berlin, 1957.
- [McM93] P. McMullen. Valuations and dissections. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume B, pages 933–988. North-Holland, Amsterdam, 1993.
- [MS83] P. McMullen and R. Schneider. Valuations on convex bodies. In P.M. Gruber and J.M. Wills, editors, *Convexity and Its Applications*, pages 170–247. Birkhäuser, Basel, 1983.
- [Sah79] C.-H. Sah. *Hilbert’s Third Problem: Scissors Congruence*. Pitman, San Francisco, 1979.
- [Wag85] S. Wagon. *The Banach-Tarski Paradox*. Cambridge University Press, 1985.

RELATED CHAPTERS

- [Chapter 7: Lattice points and lattice polytopes](#)
- [Chapter 16: Basic properties of convex polytopes](#)
- [Chapter 17: Subdivisions and triangulations of polytopes](#)
- [Chapter 40: Randomization and derandomization](#)

REFERENCES

- [AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements of polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.
- [BF87] I. Bárány and Z. Füredi. Computing the volume is difficult. *Discrete Comput. Geom.*, 2:319–326, 1987.
- [Bar93] A. Barvinok. Computing the volume, counting integral points, and exponential sums. *Discrete Comput. Geom.*, 10:123–141, 1993.
- [BN83] H. Bieri and W. Nef. A sweep-plane algorithm for computing the volume of polyhedra represented in boolean form. *Linear Algebra Appl.*, 52/53:69–97, 1983.
- [BW92] G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8:225–242, 1992.
- [Deh00] M. Dehn. Über raumgleiche Polyeder. *Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl.*, 345–354, 1900.
- [DFK89] M.E. Dyer, A.M. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volumes of convex bodies. *J. Assoc. Comput. Mach.*, 38:1–17, 1989.
- [Hil00] D. Hilbert. Mathematische Probleme. *Nachr. Königl. Ges. Wiss. Göttingen Math.-Phys. Kl.*, 253–297, 1900; *Bull. Amer. Math. Soc.*, 8:437–479, 1902.
- [KLS98] R. Kannan, L. Lovász, and M. Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures Algorithms*, 11:1–90, 1998.
- [Kep15] J. Kepler. *Nova Stereometria doliorum vinariorum*. 1615. See M. Caspar, editor, *Johannes Kepler Gesammelte Werke*, Beck, Munich, 1940.
- [Lac90] M. Laczkovich. Equidecomposability and discrepancy: a solution of Tarski’s circle-squaring problem. *J. Reine Angew. Math.*, 404:77–117, 1990.
- [Law91] J. Lawrence. Polytope volume computation. *Math. Comp.*, 57:259–271, 1991.
- [Sta86] R. Stanley. Two order polytopes. *Discrete Comput. Geom.*, 1:9–23, 1986.

31.4 MIXED VOLUMES

The study of mixed volumes, the *Brunn-Minkowski theory*, forms the backbone of classical convexity theory. It is also useful for applications in other areas, including combinatorics and algebraic geometry. A relationship to solving systems of polynomial equations is described at the end of this section.

GLOSSARY

Mixed volume: Let K_1, \dots, K_s be convex bodies in \mathbb{R}^n , and let ξ_1, \dots, ξ_s be non-negative reals. Then the function $V(\sum_{i=1}^s \xi_i K_i)$ is a homogeneous polynomial of degree n in the variables ξ_1, \dots, ξ_s , and can be written in the form

$$V\left(\sum_{i=1}^s \xi_i K_i\right) = \sum_{i_1=1}^s \sum_{i_2=1}^s \cdots \sum_{i_n=1}^s \xi_{i_1} \xi_{i_2} \cdots \xi_{i_n} V(K_{i_1}, K_{i_2}, \dots, K_{i_n}),$$

where the coefficients $V(K_{i_1}, K_{i_2}, \dots, K_{i_n})$ are invariant under permutations of their argument. The coefficient $V(K_{i_1}, K_{i_2}, \dots, K_{i_n})$ is called the mixed volume of the convex bodies $K_{i_1}, K_{i_2}, \dots, K_{i_n}$.

31.4.1 MAIN RESULTS

Mixed volumes are nonnegative, monotone, multilinear, and continuous valuations.

They generalize the ordinary volume in that $V(K) = V(\overbrace{K, \dots, K}^n)$. If A is an affine transformation, then $V(A(K_1), \dots, A(K_n)) = |\det(A)|V(K_1, \dots, K_n)$.

Among the most famous inequalities in convexity theory is the *Aleksandrov-Fenchel inequality*,

$$V(K_1, K_2, K_3, \dots, K_n)^2 \geq V(K_1, K_1, K_3, \dots, K_n) V(K_2, K_2, K_3, \dots, K_n),$$

and its consequence, the *Brunn-Minkowski theorem*, which asserts that for each $\lambda \in [0, 1]$,

$$V^{\frac{1}{n}}((1 - \lambda)K_0 + \lambda K_1) \geq (1 - \lambda)V^{\frac{1}{n}}(K_0) + \lambda V^{\frac{1}{n}}(K_1).$$

OPEN PROBLEM 31.4.1

Provide a useful geometric characterization of the sequences (K_1, \dots, K_n) for which equality holds in the Aleksandrov-Fenchel inequality.

31.4.2 TRACTABILITY RESULTS

When n is fixed, there is a polynomial-time algorithm whereby, given s (\mathcal{V} - or \mathcal{H} -) polytopes P_1, \dots, P_s in \mathbb{R}^n , all the mixed volumes $V(P_{i_1}, \dots, P_{i_n})$ can be computed.

When the dimension is part of the input, it follows at least that mixed volume computation is not harder than volume computation. In fact, computation (for \mathcal{V} -polytopes or \mathcal{S} -zonotopes) or approximation (for \mathcal{H} -polytopes) of any single mixed volume is #P-easy.

31.4.3 INTRACTABILITY RESULTS

Since mixed volumes generalize the ordinary volume, it is clear that mixed volume computation cannot be easier, in general, than volume computation. In addition, there are hardness results for mixed volumes that do not trivially depend on the hardness of volume computations. One such result is described next.

As the term is used here, a *box* is a rectangular parallelotope with axis-aligned edges. Since the vector sum of boxes $V(Z_1, \dots, Z_n)$ is again a box, the volume of the sum is easy to compute. Nevertheless, computation of the mixed volume $V(Z_1, \dots, Z_n)$ is hard. This is in interesting contrast to the fact that the volume of a sum of segments (a zonotope) is hard to compute even though each of the mixed volumes can be computed in polynomial time.

31.4.4 RANDOMIZED ALGORITHMS

Since the mixed volumes of convex bodies K_1, \dots, K_s are coefficients of the polynomial $\varphi(\xi_1, \dots, \xi_s) = V(\sum_{i=1}^s \xi_i K_i)$, it seems natural to estimate these coefficients by combining an interpolation method with a randomized volume algorithm. However, there are significant obstacles to this approach, even for the case of two bodies. First, for a general polynomial φ there is *no* way of obtaining *relative* estimates of its coefficients from *relative* estimates of the values of φ . This can be overcome in the case of two bodies by using the special structure of the polynomial $p(x) = V(K_1 + xK_2)$. However, even then the absolute values of the entries of the “inversion” that is used to express the coefficients of the polynomial in terms of its approximate values are not bounded by a polynomial, while the randomized volume approximation algorithm is polynomial only in $\frac{1}{\tau}$ but not in $\text{size}(\tau)$.

Suppose that $\psi : \mathbb{N} \rightarrow \mathbb{N}$ is nondecreasing with

$$\psi(n) \leq n \quad \text{and} \quad \psi(n) \log \psi(n) = o(\log n).$$

Then there is a polynomial-time algorithm for the problem whose instance consists of $n, s \in \mathbb{N}$, $m_1, \dots, m_s \in \mathbb{N}$ with $m_1 + m_2 + \dots + m_s = n$ and $m_1 \geq n - \psi(n)$, of well-presented convex bodies K_1, \dots, K_s of \mathbb{R}^n , and of positive rational numbers ϵ and β , and whose output is a random variable $\hat{V}_{m_1, \dots, m_s} \in \mathbb{Q}$ such that

$$\text{prob} \left\{ \frac{|\hat{V}_{m_1, \dots, m_s} - V_{m_1, \dots, m_s}|}{V_{m_1, \dots, m_s}} \geq \epsilon \right\} \leq \beta,$$

where

$$V_{m_1, \dots, m_s} = V(\overbrace{K_1, \dots, K_1}^{m_1}, \dots, \overbrace{K_s, \dots, K_s}^{m_s}).$$

Note that the hypotheses above require that m_1 is close to n , and hence that the remaining m_i 's are relatively small. A special feature of an interpolation method as used for the proof of this result is that in order to compute a *specific* coefficient of the polynomial under consideration, it computes essentially *all previous* coefficients. Since there can be a polynomial-time algorithm for computing *all such* mixed volumes only if $\psi(n) \leq \log n$, the above result is essentially best possible for any interpolation method.

In terms of approximation, [Bar97] shows that a mixed volume of n proper convex bodies can be approximated by a randomized polynomial-time algorithm within a factor of $n^{O(n)}$, while [GS02] gives a deterministic algorithm for approximating mixed volumes up to such an error.

OPEN PROBLEM 31.4.2 [DGH98]

Is there a polynomial-time randomized algorithm that, for arbitrary given $n, s \in \mathbb{N}$, $m_1, \dots, m_s \in \mathbb{N}$ with $m_1 + m_2 + \dots + m_s = n$, well-presented convex bodies K_1, \dots, K_s in \mathbb{R}^n , and positive rationals ϵ and β , computes a random variable $\hat{V}_{m_1, \dots, m_s} \in \mathbb{Q}$ such that $\text{prob}\{|\hat{V}_{m_1, \dots, m_s} - V_{m_1, \dots, m_s}| / V_{m_1, \dots, m_s} \geq \epsilon\} \leq \beta$?

Even the case $s = n$, $m_1 = \dots = m_s = 1$ is open in general. See, however, [Bar97] for some partial results.

AN APPLICATION

Let S_1, S_2, \dots, S_n be subsets of \mathbb{Z}^n , and consider a system $F = (f_1, \dots, f_n)$ of Laurent polynomials in n variables, such that the exponents of the monomials in f_i are in S_i for all $i = 1, \dots, n$. For $i = 1, \dots, n$, let

$$f_i(x) = \sum_{q \in S_i} c_q^{(i)} x^q,$$

where $f_i \in \mathbb{C}[x_1, x_1^{-1}, \dots, x_n, x_n^{-1}]$, and x^q is an abbreviation for the monomial $x_1^{q_1} \cdots x_n^{q_n}$; $x = (x_1, \dots, x_n)$ is the vector of indeterminates and $q = (q_1, \dots, q_n)$ the vector of exponents. Further, let $\mathbb{C}^* = \mathbb{C} \setminus \{0\}$.

Now, if the coefficients $c_q^{(i)}$ ($q \in S_i$) are chosen “generically,” then the number $L(F)$ of distinct common roots of the system F in $(\mathbb{C}^*)^n$ depends only on the **Newton polytopes** $P_i = \text{conv}(S_i)$ of the polynomials. More precisely,

$$L(F) = n! \cdot V(P_1, P_2, \dots, P_n).$$

In general, $L(F) \leq n! \cdot V(P_1, P_2, \dots, P_n)$. These connections can be utilized to develop a numerical continuation method for computing the isolated solutions of sparse polynomial systems. For this, see [Emi94, HS95, Roj94, Ver96].

FURTHER READING

- [BZ88] Y.D. Burago and V.A. Zalgaller. *Geometric Inequalities*. Springer-Verlag, Berlin, 1988.
- [GK94b] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: II. Volume and mixed volumes. In T. Bisztriczky, P. McMullen, R. Schneider, and A.I. Weiss, editors, *Polytopes: Abstract, Convex and Computational*, volume 440 of *NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci.*, pages 373–466. Kluwer, Dordrecht, 1994.
- [San93] J.R. Sangwine-Yager. Mixed volumes. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, Volume A, pages 43–72. North-Holland, Amsterdam, 1993.
- [Sch93] R. Schneider. *Convex Bodies: The Brunn-Minkowski Theory*. Volume 44 of *Encyclopedia Math. Appl.*, Cambridge University Press, 1993.
- [Stu02] B. Sturmfels. *Solving Systems of Polynomial Equations*. Volume 97 of *CBMS Regional Conf. Ser. in Math.*, Amer. Math. Soc., Providence, 2002.

RELATED CHAPTERS

- Chapter 16: Basic properties of convex polytopes
- Chapter 40: Randomization and derandomization

REFERENCES

- [Bar97] A. Barvinok. Computing mixed discriminants, mixed volumes and permanents. *Discrete Comput. Geom.*, 18:205–237, 1997.

-
- [DGH98] M.E. Dyer, P. Gritzmann, and A. Hufnagel. On the complexity of computing mixed volumes. *SIAM J. Comput.*, 27:356–400, 1998.
- [Emi94] I.Z. Emiris. *Sparse Elimination and Applications in Kinematics*. Ph.D. Thesis, Univ. of California, Berkeley, 1994.
- [GS02] L. Gurvits and A. Samorodnitsky. A deterministic algorithm for approximating the mixed discriminant and mixed volume, and a combinatorial corollary. *Discrete Comput. Geom.*, 27:531–550, 2002.
- [HS95] B. Huber and B. Sturmfels. A polyhedral method for solving sparse polynomial systems. *Math. Comput.*, 64:1541–1555, 1995.
- [Roj94] J.M. Rojas. *Cohomology, Combinatorics, and Complexity Arising from Solving Polynomial Systems*. Ph.D. Thesis, Univ. of California, Berkeley, 1994.
- [Ver96] J. Verschelde. *Homotopy Continuation Methods for Solving Polynomial Systems*. Ph.D. Thesis, Katholieke Universiteit Leuven, 1996.
-

31.5 CONTAINMENT PROBLEMS

Typically, containment problems involve two fixed sequences, Γ and Ω , that are given as follows: for each $n \in \mathbb{N}$, let \mathcal{C}_n denote a family of closed convex subsets of \mathbb{R}^n , and let $\omega_n : \mathcal{C}_n \rightarrow \mathbb{R}$ be a functional that is nonnegative and is monotone with respect to inclusion. Then $\Gamma = (\mathcal{C}_n)_{n \in \mathbb{N}}$ and $\Omega = (\omega_n)_{n \in \mathbb{N}}$.

GLOSSARY

(Γ, Ω) -INBODY: Accepts as input a positive integer n , a body K in \mathbb{R}^n that is given by an oracle or is an \mathcal{H} -polytope, a \mathcal{V} -polytope, or an \mathcal{S} -zonotope, and a positive rational λ . It answers the question of whether there is a $C \in \mathcal{C}_n$ such that $C \subset K$ and $\omega_n(C) \geq \lambda$.

(Γ, Ω) -CIRCUMBODY is defined similarly for $C \supset K$.

j -simplex S bound to a polytope P : Each vertex of S is a vertex of P .

Largest j -simplex in a given polytope: One of maximum j -measure.

31.5.1 THE GENERAL CONTAINMENT PROBLEM

The general containment problem deals with the question of computing, approximating, or measuring extremal bodies of a given class that are contained in or contain a given convex body. Since [GK94a] contains a broad survey of containment problems, the present account is confined to some selected examples.

31.5.2 OPTIMAL CONTAINMENT UNDER HOMOTHETY

The results on (Γ, Ω) -INBODY and (Γ, Ω) -CIRCUMBODY are summarized below for the case in which each C_n is a fixed polytope,

$$\mathcal{C}_n = \{g(C_n) \mid g \text{ is a homothety}\},$$

and

$$\omega_n(g(C_n)) = \rho, \quad \text{when } g(C_n) = a + \rho C_n \text{ for some } a \in \mathbb{R}^n \text{ and } \rho \geq 0.$$

As an abbreviation, these specific problems are denoted by $\mathcal{E}^{\text{Hom}}\text{-INBODY}$ and $\mathcal{E}^{\text{Hom}}\text{-CIRCUMBODY}$, respectively, where $\mathcal{E} = (C_n)_{n \in \mathbb{N}}$ and a subscript (\mathcal{V} or \mathcal{H}) is used to indicate the manner in which each C_n is presented.

There are polynomial-time algorithms for the following problems:

$\mathcal{E}_{\mathcal{V}}^{\text{Hom}}\text{-INBODY}$ for \mathcal{V} -polytopes P ;	$\mathcal{E}_{\mathcal{V}}^{\text{Hom}}\text{-CIRCUMBODY}$ for \mathcal{V} -polytopes P ;
$\mathcal{E}_{\mathcal{V}}^{\text{Hom}}\text{-INBODY}$ for \mathcal{H} -polytopes P ;	$\mathcal{E}_{\mathcal{H}}^{\text{Hom}}\text{-CIRCUMBODY}$ for \mathcal{V} -polytopes P ;
$\mathcal{E}_{\mathcal{H}}^{\text{Hom}}\text{-INBODY}$ for \mathcal{H} -polytopes P ;	$\mathcal{E}_{\mathcal{H}}^{\text{Hom}}\text{-CIRCUMBODY}$ for \mathcal{H} -polytopes P .

These positive results are best possible in the sense that the cases not listed above contain instances of NP-hard problems. In fact, the problem $\mathcal{E}_{\mathcal{H}}^{\text{Hom}}\text{-INBODY}$ is coNP-complete even when C_n is the standard unit \mathcal{H} -cube while P is restricted to the class of all affinely regular \mathcal{V} -cross-polytopes centered at the origin. The problem $\mathcal{E}_{\mathcal{V}}^{\text{Hom}}\text{-CIRCUMBODY}$ is coNP-complete even when C_n is the standard \mathcal{V} -cross-polytope while P is restricted to the class of all \mathcal{H} -parallelotopes centered at the origin.

There are some results for bodies that are more general than polytopes. Suppose that for each $n \in \mathbb{N}$, C_n is a centrally symmetric body in \mathbb{R}^n , and that there exists a number μ_n whose size is bounded by a polynomial in n and an n -dimensional \mathcal{S} -parallelotope Z that is strictly inscribed in $\mu_n C_n$ (i.e., the intersection of Z with the boundary of $\mu_n C_n$ consists of the vertex set of Z), the size of the presentation being bounded by a polynomial in n . Then with $\mathcal{E} = (C_n)_{n \in \mathbb{N}}$, (an appropriate variant of) the problem $\mathcal{E}^{\text{Hom}}\text{-CIRCUMBODY}$ is NP-hard for the classes of all centrally symmetric $(n-1)$ -dimensional \mathcal{H} -polytopes in \mathbb{R}^n . With the aid of polarity, similar results for $\mathcal{E}^{\text{Hom}}\text{-INBODY}$ can be obtained.

31.5.3 OPTIMAL CONTAINMENT UNDER AFFINITY: SIMPLICES

This section focuses on the problem of finding a largest j -dimensional simplex in a given n -dimensional polytope, where *largest* means of maximum j -measure.

When an n -polytope P has m vertices, it contains at most $\binom{m}{j+1}$ bound j -simplices. There is always a largest j -simplex that is bound, and hence there is a finite algorithm for finding a largest j -simplex contained in P .

Each largest j -simplex in P contains at least two vertices of P . However, there are polytopes P of arbitrarily large dimension, with an arbitrarily large number of vertices, such that some of the largest n -simplices in P have only two vertices in the vertex-set of P . Hence for $j \geq 2$ it is not clear whether there is a finite algorithm for producing a useful presentation of *all* the largest j -simplices in a given n -polytope.

The problem of finding a largest j -simplex in a \mathcal{V} - or \mathcal{H} -polytope can be solved in polynomial time when the dimension n of the polytope is fixed. Further, for fixed j , the volumes of all j -simplices in a given \mathcal{V} -polytope can be computed in polynomial time (even for varying n).

Suppose that the functions $\psi : \mathbb{N} \rightarrow \mathbb{N}$ and $\gamma : \mathbb{N} \rightarrow \mathbb{N}$ are both of order $\Omega(n^{1/k})$ for some $k \in \mathbb{N}$, and that $1 \leq \gamma(n) \leq n$ for each $n \in \mathbb{N}$. Then the following problem is NP-complete: Given $n, \lambda \in \mathbb{N}$, and the vertex set V of an n -dimensional \mathcal{V} -polytope $P \subset \mathbb{R}^n$ with $|V| \leq n + \psi(n)$, and given $j = \gamma(n)$, decide whether P contains a j -simplex S such that $(j!)^2 \text{vol}(S)^2 \geq \lambda$. Note that the conditions for γ

are satisfied when $\gamma(n) = \max\{1, n - \mu\}$ for a nonnegative integer constant μ , and also when $\gamma(n) = \max\{1, \lfloor \mu n \rfloor\}$ for a fixed rational μ with $0 < \mu \leq 1$.

A similar hardness result holds for \mathcal{H} -polytopes. There the question is the same, but the growth condition on the function γ is that $1 \leq \gamma(n) \leq n$ and that there exists a function $f : \mathbb{N} \rightarrow \mathbb{N}$, bounded by a polynomial in n , such that for each $n \in \mathbb{N}$, $f(n) - \gamma(f(n)) = n$. Note that such an f exists when the function γ is constant, and also when $\gamma(n) = \lfloor \mu n \rfloor$ for fixed rational μ with $0 < \mu < 1$.

The “dual” problem of finding smallest simplices containing a given polytope P seems even harder, since the relationship between a smallest such simplex and the faces of P is much weaker.

CONJECTURE 31.5.1 [GKL95]

For each function $\gamma : \mathbb{N} \rightarrow \mathbb{N}$ with $1 \leq \gamma(n) \leq n$, the problem of finding a largest j -simplex in a given n -dimensional \mathcal{H} -polytope P is NP-hard, even for the case in which P is a parallelotope.

With the restriction to parallelotopes, this conjecture is still open. However, under the assumption that the function $f : \mathbb{N} \rightarrow \mathbb{N}$ is such that $f(n) = \Omega(n^{1/k})$ for some fixed $k > 0$, [Pac02] establishes the NP-hardness of four problems, for each of which an instance consists of $n \in \mathbb{N}$, a rational \mathcal{H} -polytope or \mathcal{V} -polytope P in \mathbb{R}^n , and $\lambda > 0$, and the question is one of the following: Does there exist an $f(n)$ -simplex $S \subset P$ with $V^2(S) \geq \lambda$? Does there exist an $f(n)$ -dimensional simplicial cylinder $C \subset P$ with $V^2(C) \leq \lambda$?

Some approximation results can be found in [BGK00a].

APPLICATIONS

The paper [HKL96] is in part a survey of the problem of finding largest j -simplices in an n -dimensional cube. As outlined in [GK94a], applications of this problem and its relatives include the Hadamard determinant problem, finding optimal weighing designs, and bounding the growth of pivots in Gaussian elimination with complete pivoting.

31.5.4 OPTIMAL CONTAINMENT UNDER AFFINITY: ELLIPSOIDS

For an arbitrary proper body K in \mathbb{R}^n , there is a unique ellipsoid E_0 of maximum volume contained in K , and it is concentric with the unique ellipsoid E of minimum volume containing K . If a is the common center, then $K \subset a + n(E_0 - a)$, where the factor n can be replaced by \sqrt{n} when K is centrally symmetric. E is called the **Löwner-John ellipsoid** of K , and it plays an important role in the algorithmic theory of convex bodies.

Algorithmic approximations of the Löwner-John ellipsoid can be obtained by use of the ellipsoid method [GLS88]: There exists an oracle-polynomial-time algorithm that, for any well-bounded body K of \mathbb{R}^n given by a weak separation oracle, finds a point a and a linear transformation A such that

$$a + A(B^n) \subset K \subset a + (n+1)\sqrt{n}A(B^n).$$

Further, the dilatation factor $(n+1)\sqrt{n}$ can be replaced by $\sqrt{n(n+1)}$ when K is symmetric, by $(n+1)$ when K is an \mathcal{H} -polytope, and by $\sqrt{n+1}$ when K is a

symmetric \mathcal{H} -polytope.

[TKE88] and [KT93] give polynomial-time algorithms for approximating the ellipsoid of maximum volume E_0 that is contained in a given \mathcal{H} -polytope. For each rational $\gamma < 1$, there exists a polynomial-time algorithm that, given $n, m \in \mathbb{N}$ and $a_1, \dots, a_m \in \mathbb{Q}^n$, computes an ellipsoid $E = a + A(B^n)$ such that

$$E \subset P = \{x \in \mathbb{R}^n \mid \langle a_i x \rangle \leq 1, \text{ for } i = 1, \dots, m\} \quad \text{and} \quad \frac{V(E)}{V(E_0)} \geq \gamma.$$

The running time of the algorithm is

$$O(m^{3.5} \log(mR/(r \log(1/\gamma))) \log(nR/(r \log(1/\gamma)))) ,$$

where the numbers r and R are, respectively, a lower bound on the inradius of P and an upper bound on its circumradius.

It is not known whether a similar result holds for \mathcal{V} -polytopes.

As shown in [TKE88], an approximation of E_0 of the kind given above leads to the following inclusion:

$$a + A(B^n) \subset K \subset a + \frac{n(1 + 3\sqrt{1 - \gamma})}{\gamma} A(B^n).$$

FURTHER READING

- [GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math.*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.
- [GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988, 1993.

RELATED CHAPTERS

[Chapter 46: Mathematical programming](#)

REFERENCES

- [BGK00a] A. Brieden, P. Gritzmann, and V. Klee. Oracle-polynomial-time approximation of largest simplices in convex bodies. *Discrete Math.*, 221:79–92, 2000.
- [GKL95] P. Gritzmann, V. Klee, and D.G. Larman. Largest j -simplices in n -polytopes. *Discrete Comput. Geom.*, 13:477–515, 1995.
- [HKL96] M. Hudelson, V. Klee, and D. Larman. Largest j -simplices in d -cubes: Some relatives of the Hadamard maximum determinant problem. *Linear Algebra Appl.*, 241–243:519–598, 1996.
- [KT93] L. Khachiyan and M. Todd. On the complexity of approximating the maximal inscribed ellipsoid in a polytope. *Math. Programming*, 61:137–160, 1993.
- [Pac02] A. Packer. NP-hardness of largest contained and smallest containing simplices for V - and H -polytopes. *Discrete Comput. Geom.*, 28:349–377, 2002.

-
- [TKE88] S.P. Tarasov, L.G. Khachiyan, and I.I. Erlikh. The method of inscribed ellipsoids. *Soviet Math. Dokl.*, 37:226–230, 1988.
-

31.6 RADII

The diameter, width, circumradius, and inradius of a convex body are classical functionals that play an important role in convexity theory and in many applications. For other applications, generalizations have been introduced. The underlying space is a **Minkowski space** (finite-dimensional normed space) $\mathbb{M} = (\mathbb{R}^n, \|\cdot\|)$. Let B denote its unit ball, j a positive integer, and K a convex body.

GLOSSARY

Outer j -radius $R_j(K)$ of K : Infimum of the positive numbers ρ such that the space contains an $(n-j)$ -flat F for which $K \subset F + \rho B$.

j -ball of radius ρ : Set of the form $(q + \rho B) \cap F = \{x \in F \mid \|x - q\| \leq \rho\}$ for some j -flat F in \mathbb{R}^n and point $q \in F$.

Inner j -radius $r_j(K)$ of K : Maximum of the radii of the j -balls contained in K .

Diameter of K : $2r_1(K)$.

Width of K : $2R_1(K)$.

Inradius of K : $r_n(K)$.

Circumradius of K : $R_n(K)$.

For the case of variable dimension (i.e., the dimension is part of the input), Tables 31.6.1, 31.6.2, and 31.6.3 provide a rapid indication of the main complexity results for the most important radii: r_1, R_1, r_n , and R_n ; and for the three most important ℓ_p spaces: $\mathbb{R}_2^n, \mathbb{R}_1^n$, and \mathbb{R}_∞^n . The designations P, NPC, and NPH indicate respectively polynomial-time computability, NP-completeness, and NP-hardness. The tables provide only a rough indication of results. They are imprecise in the following respects: (i) the diameter and width are actually equal to $2r_1$ and $2R_1$ respectively; (ii) the results for \mathbb{R}_2^n involve the square of the radius rather than the radius itself; (iii) some of the P entries are based on polynomial-time approximability rather than polynomial-time computability; (iv) the designations NPC and NPH do not refer to computability per se, but to the appropriately related decision problems involving the establishment of lower or upper bounds for the radii in question.

For inapproximability results in the Turing machine model see [BGK00b] and [Bri02]; for sharp bounds on the approximation error of polynomial-time algorithms in the oracle model see [BGK⁺03].

APPLICATIONS

Applications of radii include conditioning in global optimization, sensitivity analysis of linear programs, orthogonal minimax regression, computer graphics and com-

 TABLE 31.6.1 Complexity of radii in \mathbb{R}_2^n .

Polytope functional		\mathcal{H} -polytopes		\mathcal{V} -polytopes	
		general	symmetric	general	symmetric
Diameter	r_1^2	NPC	NPC	P	P
Inradius	r_n^2	P	P	NPH	NPC
Width	R_1^2	NPC	P	NPC	NPC
Circumradius	R_n^2	NPC	NPC	P	P

 TABLE 31.6.2 Complexity of radii in \mathbb{R}_1^n .

Polytope functional		\mathcal{H} -polytopes		\mathcal{V} -polytopes	
		general	symmetric	general	symmetric
Diameter	r_1	NPC	NPC	P	P
Inradius	r_n	P	P	P	P
Width	R_1	P	P	P	P
Circumradius	R_n	NPC	NPC	P	P

 TABLE 31.6.3 Complexity of radii in \mathbb{R}_∞^n .

Polytope functional		\mathcal{H} -polytopes		\mathcal{V} -polytopes	
		general	symmetric	general	symmetric
Diameter	r_1	P	P	P	P
Inradius	r_n	P	P	NPC	NPC
Width	R_1	NPC	P	NPC	NPC
Circumradius	R_n	P	P	P	P

puter vision, chromosome classification, set separation, and design of membranes and sieves; see [GK93a].

FURTHER READING

- [BGK⁺03] A. Brieden, P. Gritzmann, R. Kannan, V. Klee, L. Lovász, and M. Simonovits. Deterministic and randomized polynomial-time approximation of radii. *Mathematika*, 48:63–105, 2001.
- [GK94a] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity: I. Containment problems. *Discrete Math.*, 136:129–174, 1994. Reprinted in W. Deuber, H.-J. Prömel, and B. Voigt, editors, *Trends in Discrete Mathematics*, pages 129–174. *Topics in Discrete Math.*, North-Holland, Amsterdam, 1994.

REFERENCES

- [Bri02] A. Brieden. On geometric optimization problems likely not contained in APX. *Discrete Comput. Geom.*, 28:201–209, 2002.

-
- [BGK00b] A. Brieden, P. Gritzmann and V. Klee. Inapproximability of some geometric and quadratic optimization problem. In: P.M. Pardalos (editor), *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, pages 96–115, Kluwer, Dordrecht, 2000.
- [GK93a] P. Gritzmann and V. Klee. Computational complexity of inner and outer j -radii of polytopes in finite dimensional normed spaces. *Math. Programming*, 59:163–213, 1993.
-

31.7 INTERVAL MATRICES, QUALITATIVE MATRICES

The mathematical modeling of practical problems often involves real matrices whose entries are not known precisely, but are known only to lie in specified bounded closed intervals or to be of specified sign. The interval case arises in many applications, while the study of the sign case was motivated by questions concerning the modeling of problems in economics. The associated complexity results and problems can be formulated in terms of systems of parallelotopes or systems of sign cones, and there have been some extensions to more general systems of convex sets. Attention is confined here to the two most-studied topics, solvability of linear algebraic systems and stability of linear dynamical systems.

GLOSSARY

$m \times n$ system: A sequence $\mathcal{A} = (A_1, \dots, A_n)$ of n nonempty subsets of \mathbb{R}^m .

Matrices associated with an $m \times n$ system: The set $\mathcal{M}(\mathcal{A})$ of all $m \times n$ matrices $A = [a_1, \dots, a_n]$ such that $a_j \in A_j$ for each j , where a_j denotes the j th column of A .

L-system: A system \mathcal{A} such that for each $A \in \mathcal{M}(\mathcal{A})$, the columns of A are linearly independent.

S-system: A system \mathcal{A} such that for each $A \in \mathcal{M}(\mathcal{A})$, the n columns of A are the vertices of an $(n-1)$ -simplex in \mathbb{R}^m whose relative interior includes the origin. (Equivalently, the nullspace of A is a line in \mathbb{R}^n that passes through the origin and penetrates the positive orthant of \mathbb{R}^n .)

Sign cone: A subset of \mathbb{R}^m that, for some sequence of m signs $(-, 0, +)$, consists of all points of \mathbb{R}^m that exhibit the specified sign pattern.

Qualitative matrix: An $m \times n$ matrix A in which each entry is one of the intervals $(-\infty, 0)$, $\{0\}$, and $(0, \infty)$. This may be viewed instead as an $m \times n$ system $\mathcal{A} = (A_1, \dots, A_n)$ in which each A_j is a sign cone.

L-matrix, S-matrix: A qualitative matrix that gives rise to an L-system or an S-system, respectively.

Interval matrix: An $m \times n$ matrix $A = ([\alpha_{ij}, \beta_{ij}])$ in which each entry is a bounded closed interval in \mathbb{R} . This may be viewed instead as an $m \times n$ system $\mathcal{A} = (A_1, \dots, A_n)$ in which each A_j is the parallelotope $[\alpha_{1j}, \beta_{1j}] \times \dots \times [\alpha_{mj}, \beta_{mj}]$.

Nonsingularity of a system: An $n \times n$ system \mathcal{A} is nonsingular if every member of $\mathcal{M}(\mathcal{A})$ has this property.

Sign-nonsingular: When \mathcal{A} is a system of sign cones, the preceding notion is called *sign-nonsingularity*. In other words, a sign-nonsingular matrix is a square matrix whose sign pattern guarantees nonsingularity.

Matrix stability: A square real matrix A is *semistable* (resp. *stable*) if each of its eigenvalues has nonnegative (resp. positive) real part. It is *quasistable* if it is semistable and, in addition, each eigenvalue with zero real part is a simple root of the minimum polynomial of A . These terms are used for an $n \times n$ system \mathcal{A} when they apply to *every* $A \in \mathcal{M}(\mathcal{A})$.

Matrix sign-stability: When \mathcal{A} is a system of sign cones, the preceding notion is called *sign-stability*. In other words, a *sign-stable* matrix is a square matrix whose sign pattern guarantees stability. *Sign-semistability* and *sign-quasistability* are defined similarly.

Sign-solvability: A system of linear equations, $Ax = b$, is sign-solvable if both the solvability of the system and the sign pattern of the solution x are implied by the sign patterns of A and b .

BASIC FACTS

The problem of testing a square matrix for sign-nonsingularity is polynomially equivalent to the problem of testing a digraph for the presence of a (simple) cycle that has an even number of edges. If either recognition problem admits a polynomial-time algorithm, then so does the other.

The study of sign-solvability can in a sense be decomposed into the study of L-matrices and the study of S-matrices—equivalently, into the study of L-systems and S-systems of sign cones. This result can be extended to more general $m \times n$ systems $\mathcal{A} = (A_1, \dots, A_n)$ under the assumption that each A_j has nonempty interior relative to the smallest canonical subspace of \mathbb{R}^m that contains it.

For an $n \times n$ real matrix A , consider the system $x' = Ax$ of linear differential equations with constant coefficients. For each point $p_0 \in \mathbb{R}^n$, there is a unique *positive trajectory* $x : [0, \infty) \rightarrow \mathbb{R}^n$ of this system that has $x(0) = p_0$. The matrix A is stable if and only if each positive trajectory converges to the origin, is quasistable if and only if each positive trajectory is bounded, and is semistable if and only if no positive trajectory runs off to infinity at an exponential rate.

TRACTABILITY RESULTS

There is an $O(n^2)$ algorithm for deciding whether a given $n \times (n + 1)$ matrix is an S-matrix. In the general case of systems of polyhedral cones presented in terms of their generators, an algorithm based on linear programming can recognize S-systems in polynomial time [KVL93].

There is a polynomial-time algorithm for deciding whether a square matrix is sign-nonsingular. This follows from independent deep studies of [McC97] and [RST99] that contain many other interesting results.

For a properly presented square matrix A , sign-stability, sign-quasistability, and sign-semistability can all be detected in time that is proportional to the number of nonzero entries of A [JKV87].

INTRACTABILITY RESULTS

Deciding whether a given rectangular sign matrix is an L-matrix is NP-hard, and this is true even when the matrix is “almost square” in a certain sense [KLM84].

Testing the nonsingularity of a symmetric square interval matrix is NP-hard, as is testing the stability of such a matrix [Roh94].

FURTHER READING

- [BS95] R.A. Brualdi and B.L. Shader. *Matrices of Sign-Solvable Linear Systems*. Cambridge University Press, 1995.
-

REFERENCES

- [McC97] W. McCuaig. Pólya’s permanent problem. Manuscript, 1997.
- [JKV87] C. Jeffries, V. Klee, and P. Van Den Driessche. Qualitative stability of linear systems. *Linear Algebra Appl.*, 87:1–48, 1987.
- [KLM84] V. Klee, R. Ladner, and R. Manber. Sign-solvability revisited. *Linear Algebra Appl.*, 59:131–157, 1984.
- [KVL93] V. Klee, B. Von Hohenbalken, and T. Lewis. On the recognition of S-systems. *Linear Algebra Appl.*, 192:187–204, 1993.
- [Roh94] J. Rohn. Checking positive definiteness or stability of symmetric interval matrices is NP-hard. *Comment. Math. Univ. Carolin.*, 35:795–797, 1994.
- [RST99] N. Robertson, P.D. Seymour, and R. Thomas. Permanents, Pfaffian orientations, and even directed circuits. *Ann. of Math.*, 150:929–975, 1999.

32 COMPUTATIONAL TOPOLOGY

Gert Vegter

INTRODUCTION

Topology studies point sets and their invariants under continuous deformations, invariants such as the number of connected components, holes, tunnels, or cavities. Metric properties such as the position of a point, the distance between points, or the curvature of a surface, are irrelevant to topology. A high level description of the main concepts and problems in topology is given in Section 32.1. Computational topology deals with the complexity of such problems, and with the design of efficient algorithms for their solution, in case these problems are tractable. These algorithms can deal only with spaces and maps that have a finite representation. To this end we consider simplicial complexes and maps (Section 32.2) and CW-complexes (Section 32.3). Section 32.4 deals with algebraic invariants of topological spaces, which are in general easier to compute than topological invariants. Mapping (embedding) a topological space 1–1 into another space may reveal some of its topological properties. Several types of embeddings are considered in Section 32.5. Section 32.6 deals with the classification of immersions of a space into another space. These maps are only *locally* 1–1, and hence more general than embeddings. Section 32.7 constitutes a brief introduction to Morse theory.

Many computational problems in topology are undecidable (in the sense of complexity theory). The mathematical literature of this century contains many (beautiful) topological algorithms, usually reducing to decision procedures, in many cases with exponential-time complexity. The quest for efficient algorithms for topological problems has started rather recently. Most of the problems in computational topology still await an efficient solution.

32.1 TOPOLOGICAL SPACES AND MAPS

Topology deals with the classification of spaces that are the same up to some equivalence relation. We introduce these notions, and describe some classes of topological problems.

GLOSSARY

Space: In this chapter a *topological space* (or *space*, for short) is a subset of some Euclidean space \mathbb{R}^d , endowed with the topology of \mathbb{R}^d .

Map: A function $f : X \rightarrow Y$ from a space X to a space Y is a map if f is continuous.

Homeomorphism: A 1–1 map $h : X \rightarrow Y$, with a continuous inverse, is called

a homeomorphism from X to Y (or: between X and Y).

Topological equivalence: Two spaces are topologically equivalent (or *homeomorphic*) if there is a homeomorphism between them.

Embedding: A map $f : X \rightarrow Y$ is an embedding if f is a homeomorphism onto its image. We say that X can be (topologically) embedded in Y .

Homotopy of maps: Two maps $f_0, f_1 : X \rightarrow Y$ are homotopic if there is a map $F : X \times [0, 1] \rightarrow Y$ such that $F(x, 0) = f_0(x)$ and $F(x, 1) = f_1(x)$, for all $x \in X$.

Homotopy equivalence: Two spaces X and Y are homotopy-equivalent if there are maps $f : X \rightarrow Y$ and $g : Y \rightarrow X$ such that gf and fg are homotopic to the identity mappings on X and Y , respectively. Obviously topological equivalence implies homotopy equivalence.

Topological/homotopy invariant: A map ζ associating a number, or a group, $\zeta(X)$ to a space X , is a topological invariant (resp. homotopy invariant) if $\zeta(X_1)$ and $\zeta(X_2)$ are equal, or isomorphic, for topologically equivalent (resp. homotopy-equivalent) spaces X_1 and X_2 .

Contractibility: A space is contractible if it is homotopy-equivalent to a point.

Unit interval \mathbb{I} : The interval $[0, 1]$ in \mathbb{R} .

Ball: Open d -ball: $\mathbb{B}^d = \{(x_1, \dots, x_d) \in \mathbb{R}^d \mid x_1^2 + \dots + x_d^2 < 1\}$. Closed d -ball: $\overline{\mathbb{B}}^d$ is the closure of \mathbb{B}^d .

Half ball: $\mathbb{B}_+^d = \{(x_1, \dots, x_d) \in \mathbb{R}^d \mid x_1^2 + \dots + x_d^2 < 1 \text{ and } x_d \geq 0\}$.

Sphere: $\mathbb{S}^d = \{(x_1, \dots, x_{d+1}) \in \mathbb{R}^{d+1} \mid x_1^2 + \dots + x_{d+1}^2 = 1\}$ is the d -sphere. It is the boundary of the $(d+1)$ -ball.

Manifold: A space X is a d -dimensional (topological) manifold (also: d -manifold) if every point of X has a neighborhood homeomorphic to \mathbb{B}^d . X is a d -manifold **with boundary** if every point has a neighborhood homeomorphic to \mathbb{B}^d or \mathbb{B}_+^d .

Surface: A 2-dimensional manifold, with or without boundary. A **closed surface** is a surface without boundary.

Curve: A curve in X is a continuous map $\mathbb{I} \rightarrow X$. For $x_0 \in X$, a x_0 -based **closed curve** c is a curve for which $c(0) = c(1) = x_0$.

BASIC TOPOLOGICAL PROBLEMS AND APPLICATIONS

Topological equivalence and classification: Decide whether a space belongs to (is topologically equivalent to an element of) a class of known objects.

Application: Object recognition in computer vision.

Homotopy equivalence: Decide whether two spaces are homotopy-equivalent, or whether a curve in X is contractible (the *contractibility problem*).

Applications: α -hull, skeletons; see [Ede94]. Concurrent computing; see [HS94a].

Embedding: Decide whether X can be embedded in Y . If so, construct an embedding.

Application: Graph drawing (Chapter 52), VLSI-layout, and wire routing.

Extension of maps: Let A be a subspace of X . Decide whether a map $f : A \rightarrow Y$ can be extended to X (i.e., whether there is a map $F : X \rightarrow Y$ whose restriction to A is f).

Lifting of maps: Let $f : A \rightarrow X$ and $p : Y \rightarrow X$ be maps. Decide whether there is a map $F : A \rightarrow Y$ such that $pF = f$.

Application: Inverse kinematics problems and tracking algorithms in robotics; see [Bak90] and [Section 48.1](#).

32.2 SIMPLICIAL COMPLEXES

Computation requires finite representation of topological spaces. Representing a space by a simplicial complex corresponds to the idea of building the space from simplices. Simplicial complexes may be considered as combinatorial objects, with a straightforward data structure for their representation. See also [Section 18.1](#).

GLOSSARY

Geometric simplex: A geometric k -simplex σ_k is the convex hull of a set A of $k + 1$ independent points a_0, \dots, a_k in some Euclidean space \mathbb{R}^d (so $d \geq k$). A is said to **span** the simplex σ_k . A simplex spanned by a subset A' of A is called a **face** of σ_k . The face is proper if $\emptyset \neq A' \neq A$. The **dimension** of the face is $|A'| - 1$. A 0-dimensional face is called a **vertex**, a 1-dimensional face is called an **edge**. The union σ_k^i , $0 \leq i \leq k$, of all faces of dimension at most i is called the **i -skeleton** of σ_k . In particular σ_k^0 is the set of vertices, and $\sigma_k^k = \sigma_k$. An **orientation** of σ_k is induced by an ordering of its vertices, denoted by $\langle a_0, \dots, a_k \rangle$, as follows: For any permutation π of $0, \dots, k$, the orientation $\langle a_{\pi(0)}, \dots, a_{\pi(k)} \rangle$ is equal to $(-1)^{\text{sign}(\pi)} \langle a_0, \dots, a_k \rangle$, where $\text{sign}(\pi)$ is the number of transpositions of π (so any simplex has two distinct orientations). If τ is a $(k-1)$ -dimensional face of σ , obtained by omitting the vertex a_i , then the **induced orientation** on τ is $(-1)^i \langle a_0, \dots, \hat{a}_i, \dots, a_k \rangle$, where the hat indicates omission of a_i .

Geometric simplicial complex K : A finite set of simplices in some Euclidean space \mathbb{R}^m , such that (i) if σ is a simplex of K and τ is a face of σ , then τ is a simplex of K , and (ii) if σ and τ are simplices of K , then $\sigma \cap \tau$ is either empty or a common face of σ and τ . The **dimension** of K is the maximum of the dimensions of its simplices. The **underlying space** of K , denoted by $|K|$, is the union of all simplices of K , endowed with the subspace topology of \mathbb{R}^m . The **i -skeleton** of K , denoted by K^i , is the union of all simplices of K of dimension at most i . A **subcomplex** L of K is a subset of K that is a simplicial complex.

Combinatorial simplicial complex: A pair $\mathcal{K} = (V, \Sigma)$, where V contains finitely many elements, called vertices, and Σ is a collection of subsets of V , called (combinatorial) simplices, with the property that any subset of a simplex is a simplex. The dimension of a simplex is one less than the number of vertices it contains. The dimension of \mathcal{K} is the maximum of the dimensions of its simplices.

Geometric realization: A geometric simplicial complex K in \mathbb{R}^m is called a geometric realization (in \mathbb{R}^m) of the combinatorial simplicial complex $\mathcal{K} = (V, \Sigma)$ if there is a 1-1 correspondence $f : V \rightarrow K^0$, such that $A \subset V$ is a simplex of \mathcal{K} iff $f(A)$ spans a simplex of K . Furthermore \mathcal{K} is called the **abstraction** of K .

Triangulation: A triangulation of a topological space X is a pair (K, h) , where K is a geometric simplicial complex and h is a homeomorphism from the underlying space $|K|$ to X .

Barycentric subdivision: The **barycenter** (center of mass) of a geometric k -simplex with vertices a_0, \dots, a_k in \mathbb{R}^m is the point $1/(k+1) \sum_{i=0}^k a_i$. The barycentric subdivision of a geometric simplicial complex K is defined inductively: (i) the barycentric subdivision of the 0-skeleton σ^0 is σ^0 itself; (ii) if σ is an i dimensional face of K , $i > 0$, then σ is subdivided into the collection of simplices $C(b, \tau)$, for all simplices τ in the barycentric subdivision of the $(i-1)$ -skeleton of σ . Here $C(b, \tau)$ is the convex hull of $b \cup \tau$ and b the barycenter of σ .

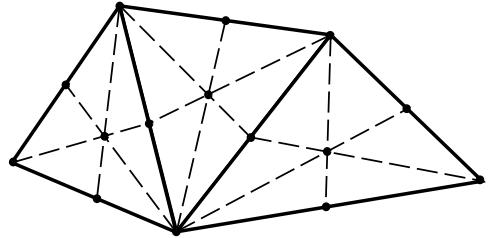


FIGURE 32.2.1
Barycentric subdivision.

The (first) barycentric subdivision of a simplicial complex K is the simplicial complex $s(K)$ obtained by barycentric subdivision of all simplices of K ; see Figure 32.2.1. The i th barycentric subdivision of K , $i > 1$, is defined inductively as $s(s^{i-1}(K))$. A simplicial complex L is called a **refinement** of K if $L = s^i(K)$, for some $i \geq 0$.

Simplicial map: A simplicial map between simplicial complexes K and L is a function $f : |K| \rightarrow |L|$ such that (i) if a is a vertex of K then $f(a)$ is a vertex of L ; (ii) if a_0, \dots, a_k are vertices of a simplex of K , then the convex hull of $f(a_0), \dots, f(a_k)$ is a simplex of L (whose dimension may be less than k); and (iii) f is linear on each simplex: if $x = \sum_{i=0}^k \lambda_i a_i$ is a point in a simplex with vertices a_0, \dots, a_k , then $f(x) = \sum_{i=0}^k \lambda_i f(a_i)$.

Simplicial equivalence: Two simplicial complexes K and L are simplicially equivalent iff there are simplicial maps $f : |K| \rightarrow |L|$ and $g : |L| \rightarrow |K|$ such that gf is the identity on $|K|$ and fg is the identity on $|L|$.

Piecewise linear (PL)-equivalence: Two simplicial complexes K and L are called piecewise linearly equivalent (PL-equivalent, for short) if there is a refinement K' of K and L' of L such that K' and L' are simplicially equivalent.

Orientation of a simplicial manifold: An orientation of a simplicial complex K , whose underlying space is a d -manifold, is a choice of orientation for each simplex of K , such that, if τ is a $(d-1)$ -face of two distinct d -simplices σ_1 and σ_2 , then the orientation on τ induced by σ_1 is the opposite of the orientation induced by σ_2 . The manifold is called **orientable** if it has a triangulation that has an orientation, otherwise it is **nonorientable**.

Euler characteristic: (Combinatorial definition; cf. Section 32.4) The Euler characteristic of a simplicial d -complex K , denoted by $\chi(K)$, is the number $\sum_{i=0}^d (-1)^i \alpha_i$, where α_i is the number of i -simplices of K .

Polygonal schema for a surface: Let $\mathcal{M}_g(a_1, b_1, \dots, a_g, b_g)$ be a regular $4g$ -gon, whose successive edges are labeled $a_1, b_1, \bar{a}_1, \bar{b}_1, \dots, a_g, b_g, \bar{a}_g, \bar{b}_g$. Edge x is directed counterclockwise, edge \bar{x} clockwise. The space obtained by identifying edges x and \bar{x} , as indicated by their direction, is a closed oriented surface, denoted by \mathbb{M}_g ; see, e.g., [Sti93, Chapter 1.4]. This surface, called the orientable surface of genus g , is homeomorphic to a 2-sphere with g handles.

Let $\mathcal{N}_g(a_1, \dots, a_g)$ be the regular $2g$ -gon whose successive edges are labeled $a_1, a_1, \dots, a_g, a_g$. Identifying edges in pairs, as indicated by their oriented labels, yields a closed nonorientable surface, denoted by \mathbb{N}_g . This surface, called the nonorientable surface of genus g , is homeomorphic to a 2-sphere with g cross-caps.

The labeled polygon \mathcal{M}_g (\mathcal{N}_g) is called the polygonal schema of \mathbb{M}_g (\mathbb{N}_g). \mathbb{M}_1 is the **torus**, \mathbb{N}_1 is the **projective plane**, \mathbb{N}_2 is the **Klein bottle**.

Minimal triangulation: A triangulation of a surface is called minimal if it has no contractible edges (i.e., contracting an edge yields a subdivision that is not a triangulation).

EXAMPLES

1. A **graph** is a 1-dimensional simplicial complex. The complete graph with n vertices is the 1-skeleton of an $(n-1)$ -simplex: $K_n = \sigma_{n-1}^1$.
 2. Every connected, compact 1-manifold is topologically equivalent to \mathbb{S}^1 or \mathbb{I} .
 3. The Delaunay triangulation of a set of points in general position in \mathbb{R}^d is a simplicial complex.
-

BASIC PROPERTIES

1. Every triangulation of an orientable manifold has an orientation (i.e., the definition of orientability does not depend on the particular triangulation).
2. The Euler characteristic is a homotopy (and hence a topological) invariant (cf. Section 32.4).
3. A simplicial 2-complex is (topologically equivalent to) a closed surface iff every edge is incident with two faces, and the faces around a vertex can be ordered as f_0, \dots, f_{k-1} so that there is exactly one edge incident with both f_i and f_{i+1} (indices modulo k).
4. An oriented closed surface X is topologically equivalent to \mathbb{S}^2 if $\chi(X) = 2$, or to \mathbb{M}_g if $\chi(X) \neq 2$, where g is uniquely determined by $\chi(X) = 2 - 2g$. A nonorientable closed surface X is topologically equivalent to \mathbb{N}_g , with $\chi(X) = 2 - g$. The number g is called the **genus** of the surface.
5. Every surface has finitely many minimal triangulations. (This number is 1 for \mathbb{S}^2 , 2 for the projective plane, and 22 for the torus; cf. Section 21.2.)
6. A simplicial complex is a 3-manifold without boundary iff every 2-simplex is incident with exactly two 3-simplices and $\chi(M) = 0$. See [Fom91, p. 184].

7. Every combinatorial simplicial d -complex has a geometric realization in \mathbb{R}^{2d+1} .
 8. Two geometric realizations K_1 and K_2 of a combinatorial simplicial complex \mathcal{K} are simplicially equivalent (therefore the topology of K does not depend on the Euclidean space in which \mathcal{K} is geometrically realized).
 9. A simplicial map $f : |K| \rightarrow |L|$ is continuous. Hence both simplicial equivalence and PL-equivalence imply topological equivalence.
 10. **Hauptvermutung:** Two simplicial complexes are PL-equivalent iff their underlying spaces are topologically equivalent. The Hauptvermutung is true if the underlying spaces are manifolds of dimension ≤ 3 , and open for manifolds of dimension exceeding 3. It is false for general simplicial complexes, see Milnor [Mil61]. (*Reidemeister torsion* is a PL-invariant, but not a topological invariant [DFN90, pp. 156, 372].)
-

ALGORITHMS, DATA STRUCTURES, AND COMPLEXITY

Representation of spaces: The *Delaunay complex* D_X is a geometric simplicial complex which is, under some conditions, homotopically (or even topologically) equivalent to a given subspace X of some Euclidean space \mathbb{R}^d . See [ES94]. For applications of simplicial complexes to geometric modeling, see [Ede94].

Classification of surfaces: The Euler characteristic and orientability of a triangulated surface with n simplices can both be computed in $O(n)$ time.

Polygonal schema for a surface of genus $g > 0$: Given a triangulation of a closed orientable (nonorientable) surface of genus $g > 0$ with n triangles, there is a sequence of $O(n)$ elementary transformations (called *cross-cap* or *handle normalizations*) that turns the triangulation into a polygonal schema of the form $\mathcal{M}_g(\mathcal{N}_g)$. This sequence of transformations can be computed in $O(n \log n)$ time [VY90].

Minimal triangulations of a surface: For a triangulation of a surface of genus g with n triangles, a sequence of $O(n)$ edge contractions leading to a minimal triangulation, can be computed in $O(n \log n)$ time [Sch91]. Therefore the classification problem for triangulated surfaces with n -triangles can be solved in $O(n)$ time; see property (4) above.

Isomorphism (simplicial equivalence): The homeomorphism problem for 2-complexes is equivalent to the graph-isomorphism problem [ÓWW00]. It is unknown whether the graph-isomorphism problem is solvable in polynomial time (in the size of the graphs). See [vL90].

PL-equivalence: Deciding whether two arbitrary simplicial d -manifolds are PL-equivalent is unsolvable for $d \geq 4$ [Sti93, Chapter 9].

OPEN PROBLEMS

1. Design an algorithm that determines whether a simplicial 3-manifold is topologically equivalent to \mathbb{S}^3 . This is a hard problem; see [VKF74] for partial results.

2. Design an algorithm that computes all minimal triangulations for a surface of genus g .
 3. Determine the minimal size of a triangulation for a triangulable d -manifold [BK87, Sar87].
-

32.3 CELL COMPLEXES

Although simplicial complexes are convenient representations of topological spaces from an algorithmic point of view, they usually have many simplices. If a representation with a smaller number of cells is desirable, CW-complexes seem appropriate. See also [Section 18.4](#).

GLOSSARY

Attaching cells to a space: Let X and Y be topological spaces, such that $X \subset Y$. We say that Y is obtained by attaching a (finite) collection of k -cells to X if $Y \setminus X$ is the disjoint union of a finite number of open k -balls $\{e_i^k \mid i \in I\}$, with the property that, for each i in the index set I , there is a map $f_i : \overline{\mathbb{B}}^k \rightarrow \overline{e}_i^k$, called the **characteristic map** of the cell e_i^k , such that $f_i(\mathbb{S}^{k-1}) \subset X$ and the restriction $f_i \mid \mathbb{B}^k$ is a homeomorphism $\mathbb{B}^k \rightarrow e_i^k$. (Note: $\overline{\mathbb{B}}^k$ need not be homeomorphic to \overline{e}_i^k .)

Cell complex (CW complex): A (finite) CW-decomposition of a topological space X is a finite sequence

$$\emptyset = X^{-1} \subset X^0 \subset X^1 \subset \cdots \subset X^d = X \quad (32.3.1)$$

such that (i) X^0 is a finite set of points, called the 0-cells of X ; (ii) for $k > 0$, X^k is obtained from X^{k-1} by attaching a finite number of k -cells to X^{k-1} . The connected components of $X^k \setminus X^{k-1}$ are called the k -cells of X . The space X is called a (finite) CW-complex. The dimension of X is the maximal dimension of the cells of X . A finite CW-complex is called **regular** if the characteristic map of each cell is a homeomorphism. (“CW” stands for “Closure-finite with the Weak topology.”)

EXAMPLES AND ELEMENTARY PROPERTIES

1. The d -sphere ($d > 0$) is a CW-complex, obtained by attaching a d -cell to a point p (so $X^k = \{p\}$, for $0 \leq k < d$, and $X^d = \mathbb{S}^d$). This CW-complex is not regular: the characteristic map of the d -cell maps the boundary of \mathbb{B}^d to a single point.
2. The **orientable surface** M_g of genus $g > 1$ is a CW-complex with one 0-cell, $2g$ 1-cells, and one 2-cell. Let the 1-cells be $a_1, b_1, \dots, a_g, b_g$, endowed with an orientation (direction). The characteristic map of the 2-cell is uniquely determined by attaching the labeled 4g-gon $M_g(a_1, b_1, \dots, a_g, b_g)$ (cf. Section 32.2) to the 1-skeleton by mapping an edge to the 1-cell with the same

label, so that the directions of the edge and the 1-cell correspond. See [VY90]. The $2g$ 1-cells are curves on the surface, disjoint except at their common endpoint (which is the 0-cell). These curves are called *canonical generators* of the surface (see [Section 32.4](#) for a justification of this nomenclature). The total number of cells is $2g + 2$, whereas the total number of simplices in a triangulation is at least $10g - 10 + \Theta(\sqrt{g})$ [JR80].

3. The *nonorientable surface* \mathbb{N}_g of genus $g > 1$ is a CW-complex, with one 0-cell, g 1-cells, and one 2-cell. The characteristic map of the 2-cell is obtained from the polygonal schema represented by the $2g$ -gon $\mathcal{N}_g(a_1, \dots, a_g)$.
4. A geometric simplicial complex is a regular CW-complex.
5. The dual map of a triangulation of a surface is a regular CW-complex, but not a simplicial complex.
6. Examples of CW-complexes arising in computational geometry are: arrangements of hyperplanes in \mathbb{R}^d (after addition of a point at infinity), the visibility complex [PV93], the free space of a polygonal robot moving amid polygonal obstacles (see [SS83] and [Chapter 47](#) of this Handbook), and the zero-set of a generic polynomial defined on $\mathbb{S}^d \subset \mathbb{R}^{d+1}$.

ALGORITHMS AND DATA STRUCTURES

Representation: A data structure for the representation and manipulation of a finite, d -manifold CW-complex is described in [Bri93].

CW-decomposition of surfaces from triangulations: For a triangulated surface of genus g , with a total of n simplices, a set of canonical generators (cf. property (2)) can be computed in $O(gn)$ time, which is optimal in the worst case [VY90]. Two algorithms achieving this time complexity have been implemented; see [LPVV01].

Each of the g or $2g$ canonical generators is represented by a polygonal curve whose vertices are on the 1-skeleton, while its other points are in the interior of a 2-simplex. In some cases the total number of edges of a single generator is $O(n)$. This method can be used to construct covering surfaces of m sheets in time $O(gnm)$ time and space; see also [Section 32.4](#).

CW-decomposition in motion planning: A general method to solve motion planning problems is the construction of a cell decomposition (Equation 32.3.1) of the free space X of the robot, together with a *retraction* $r : X \rightarrow X^k$ of X onto a low-dimensional skeleton, such that there is a motion from initial position $x_0 \in X$ to final position $x_1 \in X$ iff there is a motion from $r(x_0)$ to $r(x_1)$. This may be regarded as a reduction of the degrees of freedom of the robot. Because in general the complexity of the motion planning problem is exponential in the number of degrees of freedom, this approach simplifies the problem. For more details on the cell decomposition method in motion planning, see [Section 47.1](#).

32.4 ALGEBRAIC TOPOLOGY

In algebraic topology one associates homotopy-invariant groups (homology and homotopy groups) to a space, and homotopy-invariant homomorphisms to maps

between spaces. In passing from topology to algebra one may lose information since topologically distinct spaces may give rise to identical algebraic invariants. However, one gains on the algorithmic side, since the algebraic counterpart of an intractable topological problem may be tractable.

32.4.1 SIMPLICIAL HOMOLOGY GROUPS

Historically speaking, simplicial homology groups were among the first invariants associated with topological spaces. They are conceptually and algorithmically appealing. Modern algebraic topology usually deals with singular and cellular homology groups, which are more convenient from a mathematical point of view.

GLOSSARY

Ordered simplex: Let the vertices of a simplicial complex K be ordered v_0, \dots, v_m . A k -simplex of K with vertices v_{i_0}, \dots, v_{i_k} , $i_0 < \dots < i_k$ is represented by the symbol $[v_{i_0}, \dots, v_{i_k}]$, and called an ordered simplex.

Simplicial chain: If G is an abelian group, then an (ordered) simplicial k -chain is a formal sum of the form $\sum_j a_j \sigma_j$, with $a_j \in G$ and σ_j the symbol of a k -simplex in K . With the obvious definition for addition, the set of all (ordered) simplicial k -chains forms a (free) abelian group $C_k(K, G)$, called the group of (ordered) simplicial k -chains of K . If $G = \mathbb{Z}$, the group of integers, an element of $C_k(K, G)$ is called an **integral k -chain**.

Boundary operator: The boundary operator $\partial_k : C_k(K, G) \rightarrow C_{k-1}(K, G)$ is defined as follows. For a single (ordered) k -simplex $\sigma = [v_{i_0}, \dots, v_{i_k}]$, let $\partial_k \sigma = \sum_{h=0}^k (-1)^h [v_{i_0}, \dots, \hat{v}_{i_h}, \dots, v_{i_k}]$, and then let ∂_k be extended linearly, viz., $\partial_k(\sum_j a_j \sigma_j) = \sum_j a_j \partial_k \sigma_j$. The boundary operator is a homomorphism of groups. It satisfies $\partial_k \partial_{k+1} = 0$.

Simplicial k -cycles: $Z_k(K, G) = \ker \partial_k$ is called the group of (ordered) simplicial k -cycles.

Simplicial k -boundaries: $B_k(K, G) = \text{im } \partial_{k+1}$ is called the group of (ordered) simplicial k -boundaries. Since the boundary of a boundary is 0, B_k is a subgroup of $Z_k(K, G)$.

Simplicial homology groups: The group $H_k(K, G) = Z_k(K, G)/B_k(K, G)$ is the k th (simplicial) homology group of K . This is a purely combinatorial object, since in fact it is defined for abstract simplicial complexes. If $G = \mathbb{Z}$, these groups are called **integral homology groups**, usually denoted by $H_k(K)$. If G is a field (such as \mathbb{R}), then $H_k(K, G)$ is a vector space.

Homology groups of a triangulable topological space: $H_k(X, G) = H_k(K, G)$, if K is a simplicial complex triangulating X . This definition is independent of the triangulation K : if $h_i : K_i \rightarrow X$, $i = 1, 2$, are two triangulations of X , then $H_k(K_1, G) = H_k(K_2, G)$.

Betti numbers: The k th Betti number $\beta_k(K)$ of a simplicial complex K is the dimension of the real vector space $H_k(K, \mathbb{R})$. (For an alternative definition, see [Bre93, Chapter IV.1].)

Euler characteristic: The Euler characteristic $\chi(K)$ of a simplicial d -complex K is defined by $\chi(K) = \sum_{i=0}^d (-1)^i \beta_i(K)$. This definition is equivalent to the one of Section 32.2.

EXAMPLES

1. The *n-sphere* ($n > 0$): $H_k(\mathbb{S}^n, \mathbb{Z}) = \mathbb{Z}$, if $k = 0$ or n , and 0 otherwise.
 2. *Orientable surface*: For $g \geq 0$, $H_0(\mathbb{M}_g, G) = H_2(\mathbb{M}_g, G) = G$, $H_1(\mathbb{M}_g, G) = \bigoplus_{i=1}^{2g} G$, $H_k(\mathbb{M}_g, G) = 0$ for $k > 2$. Taking $G = \mathbb{R}$ we see that $\chi(\mathbb{M}_g) = 2 - 2g$.
 3. *Nonorientable surface*: For $g \geq 0$, $H_0(\mathbb{N}_g, \mathbb{Z}) = \mathbb{Z}$, $H_1(\mathbb{N}_g, \mathbb{Z}) = \bigoplus_{i=1}^{g-1} \mathbb{Z} \oplus \mathbb{Z}_2$, $H_k(\mathbb{N}_g, \mathbb{Z}) = 0$ for $k \geq 2$. $H_0(\mathbb{N}_g, \mathbb{R}) = \mathbb{R}$, $H_1(\mathbb{N}_g, \mathbb{R}) = \bigoplus_{i=1}^{g-1} \mathbb{R}$, $H_2(\mathbb{N}_g, \mathbb{R}) = 0$. Hence, $\chi(\mathbb{N}_g) = 2 - g$.
-

BASIC PROPERTIES

1. Homology is a homotopy invariant: if X_1 and X_2 are homotopy-equivalent, then $H_k(X_1) = H_k(X_2)$ for all k . In particular, Betti numbers and the Euler characteristic are homotopy invariants.
 2. For a simplicial d -complex K : $H_k(K, G) = 0$ for $k > d$.
 3. Let $\alpha_i(K)$ be the number of i -simplices of a simplicial d -complex K . Then $\chi(K) = \sum_{i=0}^d (-1)^i \alpha_i(K)$. This justifies the definition of χ in Section 32.2.
-

COMPUTING BETTI NUMBERS AND HOMOLOGY GROUPS

See Table 32.4.1 for the algorithmic complexity of computing the Betti numbers of several important types of spaces. The paper [DG98] also presents a method of computing a basis for the first and second homology groups of a complex in \mathbb{R}^3 of size n , in time $O(\bar{g}n^2)$, where the integer \bar{g} is an invariant of the complex, with $\bar{g} < n$.

Bounds on the sum of the Betti numbers of closed semialgebraic sets are given in [Bas99], as well as a single-exponential-time algorithm for computing the Euler characteristic of arbitrary closed semialgebraic sets.

TABLE 32.4.1 Complexity of computing Betti numbers.

TYPE OF SPACE	COMPLEXITY	SOURCE
Simplicial subcomplex of \mathbb{S}^3 of size n	$O(n\alpha(n))$	[DE95]
Simplicial complex in \mathbb{R}^3 of size n	$O(n)$	[DG98]
Sparse simplicial complex of size n	$O(n^2)$ (probabilistic)	[DC91]
Semialgebraic set, defined by m poly's ($\deg \leq d$) on \mathbb{R}^n , n fixed	polynomial in m, d	[SS83]

32.4.2 HOMOTOPY GROUPS

Homotopy groups usually provide more information than homology groups, but are generally harder to compute. The main object is the fundamental group, whose computation requires some combinatorial group theory.

GLOSSARY

Fundamental group: The space of x_0 -based curves on X is endowed with a group structure by (group multiplication) $(u_1 \cdot u_2)(t) = u_1(2t)$, if $0 \leq t \leq \frac{1}{2}$, and $u_2(2t - 1)$ if $\frac{1}{2} \leq t \leq 1$, and (inverse) $u^{-1}(t) = u(1 - t)$.

This group structure can be extended to homotopy classes of x_0 -based curves: If u, v are homotopic, then u^{-1} and v^{-1} are homotopic, and if u_i and v_i , $i = 1, 2$, are homotopic, then $u_1 \cdot u_2$ and $v_1 \cdot v_2$ are homotopic (homotopies respect the basepoint x_0). The group of homotopy classes of closed x_0 -based curves is called the fundamental group (or, the **first homotopy group**) of (X, x_0) , and is denoted by $\pi_1(X, x_0)$. If X is connected, the definition is independent of the basepoint. Then the fundamental group is denoted by $\pi_1(X)$.

Combinatorial definition of the fundamental group: If X is a connected space with triangulation K and vertices a_0, \dots, a_m , then the fundamental group has generators g_{ij} , one per ordered 1-simplex $[a_i, a_j]$, and relations $g_{ij}g_{jk}g_{ik}^{-1} = 1$, one for each ordered 2-simplex $[a_i, a_j, a_k]$ [Mau70, Chapter 3]. See [Sti93] for an introduction to combinatorial group theory.

k th homotopy group: Let $s_0 \in \mathbb{S}^k$, for $k \geq 1$. The space of homotopy classes of basepoint-preserving maps $(\mathbb{S}^k, s_0) \rightarrow (X, x_0)$ can be endowed with a group structure. The group is called the k th homotopy group of (X, x_0) , and is denoted by $\pi_k(X, x_0)$.

Word problem for a group G : Given a (finitely generated) group generated by g_1, \dots, g_k (the alphabet), and a finite set of relations of the form $g_1^{m_1} \cdots g_k^{m_k} = 1$ (rewrite rules) with $m_i \in \mathbb{Z}$, decide whether a given word of the form $g_1^{n_1} \cdots g_k^{n_k}$ represents the unit element 1.

Covering space: A continuous map $p : Y \rightarrow X$ is a covering map if every point $x \in X$ has a connected neighborhood U such that for each connected component V of $p^{-1}(x)$ the restriction of p to V is a homeomorphism $V \rightarrow U$. Y is called a covering space of X . If the cardinality n of $p^{-1}(U)$ is finite, Y is called an n -sheeted cover of X . This number is the same for all $x \in X$.

Universal covering space: A connected covering space Y of X is called universal if $\pi_1(Y) = 0$.

EXAMPLES

1. *The n -sphere ($n > 0$):* $\pi_1(\mathbb{S}^n, s_0) = \mathbb{Z}$ if $n = 1$, and 0 otherwise.
2. *Orientable surface of genus $g \geq 1$:* $\pi_1(\mathbb{M}_g)$ is generated by $2g$ generators $a_1, b_1, \dots, a_g, b_g$, with the single relation $a_1b_1a_1^{-1}b_1^{-1} \cdots a_gb_ga_g^{-1}b_g^{-1} = 1$.
3. *Nonorientable surface of genus $g \geq 1$:* $\pi_1(\mathbb{N}_g)$ is generated by g generators a_1, \dots, a_g , with the single relation $a_1a_1 \cdots a_ga_g = 1$.
4. *Universal covering space:* The universal covering space of \mathbb{S}^1 is \mathbb{R} , with covering map $p : \mathbb{R} \rightarrow \mathbb{S}^1$ defined by $p(t) = (\cos t, \sin t)$. The universal covering space of the projective plane \mathbb{P} is \mathbb{S}^2 , the covering map being antipodal identification. The plane is the universal covering space of \mathbb{M}_g and \mathbb{N}_g , $g > 0$.

BASIC PROPERTIES

1. The homotopy groups are homotopy invariants.
2. The first integral homology group is the abelianized fundamental group.
3. The fundamental group of a simplicial complex is the fundamental group of its 2-skeleton.
4. For every finitely generated group G there is a finite simplicial 2-complex K and a 4-manifold M such that $\pi_1(K) = G$ and $\pi_1(M) = G$.
5. Homotopy invariants are topological invariants, but not vice versa. For example, the *lens spaces* $L(5, 1)$ and $L(5, 2)$ are not homotopy-equivalent, but do have isomorphic homology and homotopy groups [Bre93, Chapter VI].
6. Let Y be a universal covering space of X with covering map $p : Y \rightarrow X$, and let $y_0 \in Y$ and $x_0 = p(y_0) \in X$. Every curve $c : \mathbb{I} \rightarrow X$ with $c(0) = x_0$ has a unique lift $\bar{c} : \mathbb{I} \rightarrow Y$ with $\bar{c}(0) = y_0$. Furthermore, a closed curve c is contractible in X iff \bar{c} is a closed curve in Y , i.e., $\bar{c}(1) = y_0$ [Sti93, Chapter 6]. This is the basis of Dehn's algorithm for the contractibility problem on surfaces (see below).

ALGORITHMS AND COMPLEXITY

Undecidability of homeomorphism problem: The word problem for general groups is undecidable. Hence the contractibility problem for general simplicial 2-complexes, and for manifolds of dimension ≥ 4 , is undecidable [Sti93]. A slight variation even proves that the homeomorphism problem for 4-manifolds is undecidable.

Contractibility problem for surfaces: Determine whether a curve with k edges on a triangulated surface \mathbb{M}_g of size n is contractible, and, if so, construct a contraction.

Dey and Schipper [DS95] implement Dehn's algorithm in $O(n + k \log g)$ time and $O(n + k)$ space by constructing a finite portion of the covering surface of \mathbb{M}_g , for $g > 1$, and determining whether the lift of the curve to the covering space is closed. These algorithms can also be applied to solving the homotopy problem for curves on a surface.

The paper [DG99] presents an algorithmic solution of the word problem for fundamental groups of the orientable surfaces \mathbb{M}_g , if $g \neq 2$, and of the nonorientable surfaces \mathbb{N}_g , if $g \neq 3, 4$. This algorithm yields a method to decide whether a curve on such a surface is contractible in $O(n + k)$ time and space, which is optimal.

Representation problem: There is an algorithm that decides whether a homotopy class of curves contains a simple closed curve. The algorithm of [Chi72] can be turned into a polynomial-time algorithm using methods similar to those of [Sch92] and [VY90]. (Poincaré had already given a condition for a *homology* class of a curve on a surface to contain a simple closed curve. This can also be turned into a polynomial algorithm along similar lines.)

Homotopy of polygonal paths among points in the plane: Several algorithms determine whether two polygonal paths in the plane with n points removed

are homotopic. Hershberger and Snoeyink [HS94b] construct part of the covering space to compute minimum length curves that are homotopy-equivalent to a given curve, in $\Theta(n^2)$ time, where n is the number of point-shaped holes and the input curve consists of at most n edges. Cabello, Liu, Mantler, and Snoeyink [CLMS02] present an $O(n \log n)$ algorithm to test whether two *simple* paths, with the same endpoints, are homotopic. See [Section 27.2](#).

32.5 EMBEDDING SIMPLICIAL COMPLEXES

Embeddability problems are important for their own sake, but also for computations. Especially important algorithmically is the problem of embedding a simplicial complex in a Euclidean space of lowest dimension. See also [Section 21.1](#).

GLOSSARY

Simplicial embedding of a simplicial complex K in simplicial complex L : A simplicial map $f : |K| \rightarrow |L|$ that is a topological embedding.

Geometric embedding of a simplicial complex K in \mathbb{R}^d : A simplicial equivalence $f : K \rightarrow L$, where L is a geometric simplicial complex in \mathbb{R}^d .

Piecewise-linear (PL) embedding of a simplicial complex K in a simplicial complex L : A simplicial embedding of a refinement K' of K in a refinement L' of L . If L is a geometric simplicial complex in \mathbb{R}^d , we say that K can be PL-embedded in \mathbb{R}^d .

PL-minimality: A simplicial complex is PL-minimal in \mathbb{R}^d if it is not PL-embeddable in \mathbb{R}^d , but every proper subcomplex can be PL-embedded in \mathbb{R}^d .

Genus of a graph: The orientable (nonorientable) genus of a graph G is the minimal genus of an orientable (nonorientable) surface in which G is PL-embeddable.

Book: A book with p pages is a simplicial complex consisting of p triangles sharing a common edge (and nothing else).

Page number of a graph: Minimal number of pages of a book in which the graph is PL-embeddable.

32.5.1 PL-EMBEDDINGS

BASIC RESULTS

1. A simplicial d -complex that is topologically embeddable in \mathbb{R}^{2d} is also PL-embeddable in \mathbb{R}^{2d} [Web67].
2. For $d \geq 3$, a simplicial d -complex K is PL-embeddable in \mathbb{R}^{2d} iff its *van Kampen obstruction class* $o(K) = 0$. ($o(K)$ is an element of the $2d$ th cohomology group of the symmetric product of K minus the diagonal; see [vK33, Sha57].) If K is a triangulation of a d -manifold, then $o(K) = 0$, so K can be embedded in \mathbb{R}^{2d} [Whi44].

3. **Kuratowski's theorem:** a graph G is PL-embeddable in the plane iff K_5 and $K_{3,3}$ are not PL-embeddable in G . The graphs K_5 and $K_{3,3}$ are called **forbidden minors** for planarity.
4. Every orientable triangulated surface can be PL-embedded in \mathbb{R}^3 . Every nonorientable triangulated surface can be PL-embedded in \mathbb{R}^4 , but not in \mathbb{R}^3 (for a simple proof of the latter, see [Mae93]).
5. Kuratowski's theorem can be rephrased by saying that K_5 and $K_{3,3}$ are the only PL-minimal 1-complexes in \mathbb{R}^2 . For each $n \geq 2$ and each d , with $n+1 \leq d \leq 2n$, there are countably many nonhomeomorphic n -complexes that are all PL-minimal in \mathbb{R}^d [Zak69].
6. There is a finite set of forbidden minors for PL-embeddability in a surface of fixed genus g [RS90].
7. The page-number of a graph is $O(g)$ [HI92].

ALGORITHMS AND COMPLEXITY

PL-embeddability of graphs: It can be decided in $O(n \log n)$ time whether a graph with n vertices is planar (PL-embeddable in the plane). In $O(n \log n)$ time a geometric embedding in the plane can be constructed [HT74].

Graph genus: The graph genus problem is NP-complete [Tho89].

OPEN PROBLEMS

1. Give an efficient algorithm that computes the van Kampen obstruction $o(K)$ for a simplicial d -complex K with a total of n simplices. Find an algorithm that constructs a PL-embedding (of reasonable complexity) for K in case $o(K) = 0$.
2. Design an efficient algorithm that determines whether a simplicial d -complex can be PL-embedded in \mathbb{R}^k , for $d \leq k < 2d$.

32.5.2 GEOMETRIC EMBEDDINGS

MAIN RESULTS

1. Every simplicial d -complex can be geometrically embedded in \mathbb{R}^{2d+1} .
2. Every simplicial 1-complex (graph) that is PL-embeddable in \mathbb{R}^2 can be geometrically embedded in \mathbb{R}^2 (**Fáry's theorem**).
3. For each $d \geq 2$ there is a simplicial d -complex that is PL-embeddable in \mathbb{R}^{d+1} , but not geometrically embeddable in \mathbb{R}^{d+1} [Duk70].
4. All minimal triangulations of the 2-sphere and the torus can be geometrically embedded in \mathbb{R}^3 [BW93]. All minimal triangulations of the projective plane can be geometrically embedded in \mathbb{R}^4 [BW93].

ALGORITHMS

Geometric embeddability of a graph: It can be decided in $O(n \log n)$ time whether a simplicial 1-complex (graph) with n cells (edges and vertices) can be geometrically embedded in the plane. If such an embedding exists, it can be constructed in $O(n \log n)$ time [HT74].

OPEN PROBLEMS

1. Can every minimal triangulation (see [Section 32.2](#)) of the surface of genus g be geometrically embedded in \mathbb{R}^3 (cf. [BW93])?
2. Design an efficient (polynomial-time) algorithm that determines whether a simplicial d -complex can be geometrically embedded in \mathbb{R}^k , for $d \leq k \leq 2d$.
3. Prove or disprove: If a simplicial d -complex is PL-embeddable in \mathbb{R}^{2d}

- [Mil61] J. Milnor. Two complexes which are homeomorphic but combinatorially distinct. *Ann. of Math.*, 74:575–590, 1961.
- [Mil73] J. Milnor. *Morse Theory*, volume 51 of *Ann. of Math. Stud.* Princeton University Press, 1973.
- [ÓWW00] C. Ó'Dúnlaing, C. Watt, and D. Wilkins. Homeomorphism of 2-complexes is equivalent to graph isomorphism. *Internat. J. Comput. Geom. Appl.*, 10:453–476, 2000.
- [Phi66] A. Phillips. Turning a surface inside out. *Sci. Amer.*, 214:112–120, May 1966.
- [PV93] M. Pocchiola and G. Vegter. The visibility complex. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 328–337, 1993.
- [RS90] N. Robertson and P.D. Seymour. Graph minors VIII. A Kuratowski theorem for general surfaces. *J. Combin. Theory Ser. B*, 48:255–288, 1990.
- [Sar87] K.S. Sarkaria. Heawood inequalities. *J. Combin. Theory Ser. A*, 46:50–78, 1987.
- [Sch91] H. Schipper. Generating triangulations of 2-manifolds. In *Computational Geometry—Methods, Algorithms and Applications: Proc. 7th Internat. Workshop Comput. Geom. (CG '91)*, volume 553 of *Lecture Notes in Comput. Sci.*, pages 237–248. Springer-Verlag, Berlin, 1991.
- [Sch92] H. Schipper. Determining contractibility of curves. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 358–367, 1992.
- [Sha57] A. Shapiro. Obstructions to the imbedding of a complex in a euclidean space. I. The first obstruction. *Ann. of Math.*, 66:256–269, 1957.
- [Sma58a] S. Smale. Regular curves on Riemannian manifolds. *Trans. Amer. Math. Soc.*, 87:492–512, 1958.
- [Sma58b] S. Smale. A classification of immersions of the two-sphere. *Trans. Amer. Math. Soc.*, 90:281–290, 1958.
- [SS83] J.T. Schwartz and M. Sharir. On the piano mover's problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. in Appl. Math.*, 4:298–351, 1983.
- [ST34] H. Seifert and W. Threlfall. *Lehrbuch der Topologie*. Teubner, Leipzig, 1934. English translation, *A Textbook of Topology*. Academic Press, New York, 1980.
- [Sti93] J. Stillwell. *Classical Topology and Combinatorial Group Theory*, volume 72 of *Grad. Texts in Math.* Springer-Verlag, New York, 1993.
- [Tho89] C. Thomassen. The graph genus problem is NP-complete. *J. Algorithms*, 10:568–576, 1989.
- [Veg89] G. Vegter. Kink-free deformations of polygons. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 61–68, 1989.
- [vK33] E.R. van Kampen. Komplexe in euklidischen Räumen. *Abh. Math. Sem. Hamb.*, 9:72–78 and 152–153, 1933.
- [VKF74] I.A. Volodin, V.E. Kuznetsov, and A.T. Fomenko. The problem of discriminating algorithmically the standard three-dimensional sphere. *Russian Math. Surveys*, 29:71–172, 1974.
- [vL90] J. van Leeuwen. Graph algorithms. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, pages 525–631. Elsevier, Amsterdam, 1990.
- [VY90] G. Vegter and C.K. Yap. Computational complexity of combinatorial surfaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 102–111, 1990.
- [Web67] C. Weber. Plongement des polyèdres dans le domaine métastable. *Comment. Math. Helv.*, 42:1–27, 1967.
- [Whi44] H. Whitney. The self-intersections of a smooth n -manifold in $2n$ -space. *Ann. of Math.*, 45:221–246, 1944.
- [Zak69] J. Zaks. On minimal complexes. *Pacific J. Math.*, 28:721–727, 1969.

33 COMPUTATIONAL REAL ALGEBRAIC GEOMETRY

Bhubaneswar Mishra

INTRODUCTION

Computational real algebraic geometry studies various algorithmic questions dealing with the *real solutions* of a system of equalities, inequalities, and inequations of polynomials over the real numbers. This emerging field is largely motivated by the power and elegance with which it solves a broad and general class of problems arising in robotics, vision, computer-aided design, geometric theorem proving, etc.

The algorithmic problems that arise in this context are formulated as decision problems for the *first-order theory of reals* and the related problems of *quantifier elimination* (Section 33.1). The associated geometric structures are then examined via an exploration of the *semialgebraic sets* (Section 33.2). Algorithmic problems for semialgebraic sets are considered next. In particular, Section 33.3 discusses real algebraic numbers and their representation, relying on such classical theorems as Sturm's theorem and Thom's lemma (Section 33.3). This discussion is followed by a description of semialgebraic sets using the concept of *cylindrical algebraic decomposition* (CAD) in both one and higher dimensions (Sections 33.4 and 33.5). This leads to brief descriptions of two algorithmic approaches for the decision and quantifier elimination problems (Section 33.6): namely, Collins's algorithm based on CAD, and some more recent approaches based on critical points techniques and on reducing the multivariate problem to easier univariate problems. These new approaches rely on the work of several groups of researchers: Grigor'ev and Vorob'ev [Gri88, GV88], Canny [Can88a, Can90], Heintz et al. [HRS90], Renegar [Ren91, Ren92a, Ren92b, Ren92c], and Basu et al. [BPR96]. A few representative applications of computational algebra conclude this chapter (Section 33.7).

33.1 FIRST-ORDER THEORY OF REALS

The *decision problem* for the first-order theory of reals is to determine if a *Tarski sentence* in the first-order theory of reals is true or false. The *quantifier elimination problem* is to determine if there is a logically equivalent quantifier-free formula for an arbitrary Tarski formula in the first-order theory of reals. As a result of Tarski's work, we have the following theorem.

THEOREM 33.1.1 [Tar51]

- Let Ψ be a Tarski sentence. There is an effective decision procedure for Ψ .
- Let Ψ be a Tarski formula. There is a quantifier-free formula ϕ logically equiv-

alent to Ψ . If Ψ involves only polynomials with rational coefficients, then so does the sentence ϕ .

Tarski formulas are formulas in a first-order language (defined by Tarski in 1930 [Tar51]) constructed from equalities, inequalities, and inequations of polynomials over the reals. Such formulas may be constructed by introducing logical connectives and universal and existential quantifiers to the atomic formulas. *Tarski sentences* are Tarski formulas in which all variables are bound by quantification.

GLOSSARY

Term: A constant, variable, or term combining two terms by an arithmetic operator: $\{+, -, \cdot, /\}$. A constant is a real number. A variable assumes a real number as its value. A term contains finitely many such algebraic variables: x_1, x_2, \dots, x_n .

Atomic formula: A formula comparing two terms by a binary relational operator: $\{=, \neq, >, <, \geq, \leq\}$.

Quantifier-free formula: An atomic formula, a negation of a quantifier-free formula given by the unary Boolean connective $\{\neg\}$, or a formula combining two quantifier-free formulas by a binary Boolean connective: $\{\Rightarrow, \wedge, \vee\}$. *Example:* The formula $(x^2 - 2 = 0) \wedge (x > 0)$ defines the (real algebraic) number $+\sqrt{2}$.

Tarski formula: If $\phi(y_1, \dots, y_r)$ is a quantifier-free formula, then it is also a Tarski formula. All the variables y_i are *free* in ϕ . Let $\Phi(y_1, \dots, y_r)$ and $\Psi(z_1, \dots, z_s)$ be two Tarski formulas (with free variables y_i and z_i , respectively); then a formula combining Φ and Ψ by a Boolean connective is a Tarski formula with free variables $\{y_i\} \cup \{z_i\}$. Lastly, if \mathcal{Q} stands for a quantifier (either universal \forall or existential \exists) and if $\Phi(y_1, \dots, y_r, x)$ is a Tarski formula (with free variables x and y), then

$$(\mathcal{Q} x) [\Phi(y_1, \dots, y_r, x)]$$

is a Tarski formula with only the y 's as free variables. The variable x is *bound* in $(\mathcal{Q} x)[\Phi]$.

Tarski sentence: A Tarski formula with no free variable.

Example: $(\exists x) (\forall y) [y^2 - x < 0]$. This Tarski sentence is false.

Prenex Tarski formula: A Tarski formula of the form

$$(\mathcal{Q} x_1) (\mathcal{Q} x_2) \cdots (\mathcal{Q} x_n) [\phi(y_1, y_2, \dots, y_r, x_1, \dots, x_n)],$$

where ϕ is quantifier-free. The string of quantifiers $(\mathcal{Q} x_1) (\mathcal{Q} x_2) \cdots (\mathcal{Q} x_n)$ is called the *prefix* and ϕ is called the *matrix*.

Prenex form of a Tarski formula, Ψ : A prenex Tarski formula logically equivalent to Ψ . For every Tarski formula, one can find its prenex form using a simple procedure that works in four steps: (1) eliminate redundant quantifiers; (2) rename variables so that the same variable does not occur as free and bound; (3) move negations inward; and finally, (4) push quantifiers to the left.

Extension of a Tarski formula, $\Phi(y_1, \dots, y_r)$ with free variables $\{y_1, \dots, y_r\}$: The set of all $\langle \zeta_1, \dots, \zeta_r \rangle \in \mathbb{R}^r$ such that

$$\Phi(\zeta_1, \dots, \zeta_r) = \text{True}.$$

THE DECISION PROBLEM

The general **decision problem** for the first-order theory of reals is to determine if a given Tarski sentence is true or false. A particularly interesting special case of the problem is when all the quantifiers are existential. We refer to the decision problem in this case as the **existential problem** for the first-order theory of reals.

The general decision problem was shown to be decidable by Tarski [Tar51]. However, the complexity of Tarski's original algorithm could only be given by a very rapidly growing function of the input size (e.g., a function that could not be expressed as a bounded tower of exponents of the input size). The first algorithm with substantial improvement over Tarski's algorithm was due to Collins [Col75]; it has a doubly-exponential time complexity in the number of variables appearing in the sentence. Further improvements have been made by a number of researchers (Grigor'ev-Vorobjov [Gri88, GV88], Canny [Can88b, Can93], Heintz et al. [HRS89, HRS90], Renegar [Ren92a,b,c]) and most recently by Basu et al. [BPR98].

In the following, we assume that our Tarski sentence is presented in its prenex form:

$$(\mathcal{Q}_1 \mathbf{x}^{[1]}) (\mathcal{Q}_2 \mathbf{x}^{[2]}) \cdots (\mathcal{Q}_\omega \mathbf{x}^{[\omega]}) [\psi(\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[\omega]})],$$

where the \mathcal{Q}_i 's form a sequence of alternating quantifiers (i.e., \forall or \exists , with every pair of consecutive quantifiers distinct), with $\mathbf{x}^{[i]}$ a partition of the variables

$$\bigcup_{i=0}^{\omega} \mathbf{x}^{[i]} = \{x_1, x_2, \dots, x_n\} \triangleq \mathbf{x}, \quad \text{and} \quad |\mathbf{x}^{[i]}| = n_i,$$

and where ψ is a quantifier-free formula with atomic predicates consisting of polynomial equalities and inequalities of the form

$$g_i(\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[\omega]}) \geqslant 0, \quad i = 1, \dots, m.$$

Here, g_i is a multivariate polynomial (over \mathbb{R} or \mathbb{Q} , as the case may be) of total degree bounded by d . There are a total of m such polynomials. The special case $\omega = 1$ reduces the problem to that of the existential problem for the first-order theory of reals.

If the polynomials of the basic equalities, inequalities, inequations, etc., are over the rationals, then we assume that their coefficients can be stored with at most L bits. Thus the arithmetic complexity can be described in terms of n , n_i , ω , m , and d , and the bit complexity will involve L as well.

Table 33.1.1 highlights a representative set of known bit-complexity results for the decision problem.

QUANTIFIER ELIMINATION PROBLEM

Formally, given a Tarski formula of the form,

$$\Psi(\mathbf{x}^{[0]}) = (\mathcal{Q}_1 \mathbf{x}^{[1]}) (\mathcal{Q}_2 \mathbf{x}^{[2]}) \cdots (\mathcal{Q}_\omega \mathbf{x}^{[\omega]}) [\psi(\mathbf{x}^{[0]}, \mathbf{x}^{[1]}, \dots, \mathbf{x}^{[\omega]})],$$

where ψ is a quantifier-free formula, the **quantifier elimination problem** is to construct another quantifier-free formula, $\phi(\mathbf{x}^{[0]})$, such that $\phi(\mathbf{x}^{[0]})$ holds if and

 TABLE 33.1.1 Selected time complexity results.

GENERAL OR EXISTENTIAL	TIME COMPLEXITY	SOURCE
General	$L^3(md)^{2^O(\sum n_i)}$	[Col75]
Existential	$L^{O(1)}(md)^{O(n^2)}$	[GV92]
General	$L^{O(1)}(md)^{(O(\sum n_i))^{4\omega-2}}$	[Gri88]
Existential	$L^{1+o(1)}(m)^{(n+1)}(d)^{O(n^2)}$	[Can88b, Can93]
General	$(L \log L \log \log L)(md)^{(2^O(\omega))^{\Pi n_i}}$	[Ren92a,b,c]
Existential	$(L \log L \log \log L)m(m/n)^n(d)^{O(n)}$	[BPR96]
General	$(L \log L \log \log L)(m)^{\Pi(n_i+1)}(d)^{\Pi O(n_i)}$	[BPR96]

only if $\Psi(\mathbf{x}^{[0]})$ holds. Such a quantifier-free formula takes the form

$$\phi(\mathbf{x}^{[0]}) \equiv \bigvee_{i=1}^I \bigwedge_{j=1}^{J_i} \left(f_{i,j}(\mathbf{x}^{[0]}) \geqslant 0 \right),$$

where $f_{i,j} \in \mathbb{R}[\mathbf{x}^{[0]}]$ is a multivariate polynomial with real coefficients.

Significantly improved bounds were given by Basu et al. [BPR96] and are summarized as follows:

$$\begin{aligned} I &\leq (m) \prod_{i>0} (n_i+1) (d) \prod_{i>0} O(n_i) \\ J_i &\leq (m) \prod_{i>0} (n_i+1) (d) \prod_{i>0} O(n_i). \end{aligned}$$

The total degrees of the polynomials $f_{i,j}(\mathbf{x}^{[0]})$ are bounded by

$$(d) \prod_{i>0} O(n_i).$$

Nonetheless, comparing the above bounds to the bounds obtained in *semilinear geometry*, it appears that the “combinatorial part” of the complexity of both the formula and the computation could be improved to $(m) \prod_{i>0} (n_i+1)$. As a consequence of some recent results of Basu [Bas99], the best bound for the size of the equivalent quantifier-free formula is now

$$I, J_i \leq (m) \prod_{i>0} (n_i+1) (d)^{n'_0} \prod_{i>0} O(n_i),$$

where $n'_0 = \min(n_0, \tau \prod_{i>0} (n_i+1))$ and τ is a bound on the number of free variables occurring in any polynomial in the original Tarski formula. The total degrees of the polynomials $f_{i,j}(\mathbf{x}^{[0]})$ are still bounded by

$$(d) \prod_{i>0} O(n_i).$$

Furthermore, the algorithmic complexity of Basu’s new procedure involves only $(m) \prod_{i>0} (n_i+1) (d)^{n'_0} \prod_{i>0} O(n_i)$ arithmetic operations.

Lower bound results for the quantifier elimination problem can be found in Davenport and Heintz [DH88]. They showed that for every n , there exists a Tarski

formula Ψ_n with n quantifiers, of length $O(n)$, and of constant degree, such that any quantifier-free formula ψ_n logically equivalent to Ψ_n must involve polynomials of

$$\text{degree} = 2^{2^{\Omega(n)}} \quad \text{and} \quad \text{length} = 2^{2^{\Omega(n)}}.$$

Note that in the simplest possible case (i.e., $d = 2$ and $n_i = 2$), upper and lower bounds are doubly exponential and match well. This result, however, does not imply a similar lower bound for the decision problems.

33.2 SEMIALGEBRAIC SETS

Every quantifier-free formula composed of polynomial inequalities and Boolean connectives defines a semialgebraic set. Thus, these semialgebraic sets play an important role in real algebraic geometry.

GLOSSARY

Semialgebraic set: A subset $S \subseteq \mathbb{R}^n$ defined by a set-theoretic expression involving a system of polynomial inequalities

$$S = \bigcup_{i=1}^I \bigcap_{j=1}^{J_i} \left\{ \langle \xi_1, \dots, \xi_n \rangle \in \mathbb{R}^n \mid \operatorname{sgn}(f_{i,j}(\xi_1, \dots, \xi_n)) = s_{i,j} \right\},$$

where the $f_{i,j}$'s are multivariate polynomials over \mathbb{R} and the $s_{i,j}$'s are corresponding sets of signs in $\{-1, 0, +1\}$.

Real algebraic set: A subset $Z \subseteq \mathbb{R}^n$ defined by a system of algebraic equations.

$$Z = \left\{ \langle \xi_1, \dots, \xi_n \rangle \in \mathbb{R}^n \mid f_1(\xi_1, \dots, \xi_n) = \dots = f_m(\xi_1, \dots, \xi_n) = 0 \right\},$$

where the f_i 's are multivariate polynomials over \mathbb{R} .

Semialgebraic map: A map $\theta : S \rightarrow T$, from a semialgebraic set $S \subseteq \mathbb{R}^m$ to a semialgebraic set $T \subseteq \mathbb{R}^n$, such that its graph $\{ \langle s, \theta(s) \rangle \in \mathbb{R}^{m+n} : s \in S \}$ is a semialgebraic set in \mathbb{R}^{m+n} . Note that projection, being linear, is a semialgebraic map.

TARSKI-SEIDENBERG THEOREM

Equivalently, semialgebraic sets can be defined as

$$S = \left\{ \langle \xi_1, \dots, \xi_n \rangle \in \mathbb{R}^n \mid \psi(\xi_1, \dots, \xi_n) = \text{True} \right\},$$

where $\psi(x_1, \dots, x_n)$ is a quantifier-free formula involving n algebraic variables. As a direct corollary of Tarski's theorem on quantifier elimination, we see that extensions of Tarski formulas are also semialgebraic sets.

While real algebraic sets are quite interesting and would be natural objects of study in this context, *they are not closed under projection onto a subspace*. Hence they tend to be unwieldy. However, *semialgebraic sets are closed under projection*. This follows from a more general result: the famous **Tarski-Seidenberg theorem** which is an immediate consequence of quantifier elimination, since images are described by formulas involving only existential quantifiers.

THEOREM 33.2.1 Tarski-Seidenberg Theorem [Sei74]

Let S be a semialgebraic set in \mathbb{R}^m , and let $\theta : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a semialgebraic map. Then $\theta(S)$ is semialgebraic in \mathbb{R}^n .

In fact, semialgebraic sets can be defined simply as the smallest class of subsets of \mathbb{R}^n containing real algebraic sets and closed under projection.

GLOSSARY

Connected component of a semialgebraic set: A maximal connected subset of a semialgebraic set. Semialgebraic sets have a finite number of connected components and these are also semialgebraic.

Semialgebraic decomposition of a semialgebraic set S : A finite collection \mathcal{K} of disjoint connected semialgebraic subsets of S whose union is S . The collection of connected components of a semialgebraic set forms a semialgebraic decomposition. Thus, every semialgebraic set admits a semialgebraic decomposition.

Set of sample points for S : A finite number of points meeting every nonempty connected component of S .

Sign assignment: A vector of sign values of a set of polynomials at a point p . More formally, let \mathcal{F} be a set of real multivariate polynomials in n variables. Any point $p = \langle \xi_1, \dots, \xi_n \rangle \in \mathbb{R}^n$ has a **sign assignment** with respect to \mathcal{F} as follows:

$$\text{sgn}_{\mathcal{F}}(p) = \left\langle \text{sgn}(f(\xi_1, \dots, \xi_n)) \mid f \in \mathcal{F} \right\rangle.$$

A sign assignment induces an equivalence relation: Given two points $p, q \in \mathbb{R}^n$, we say

$$p \sim_{\mathcal{F}} q, \quad \text{if and only if} \quad \text{sgn}_{\mathcal{F}}(p) = \text{sgn}_{\mathcal{F}}(q).$$

Sign class of \mathcal{F} : An equivalence class in the partition of \mathbb{R}^n defined by the equivalence relation $\sim_{\mathcal{F}}$.

Semialgebraic decomposition for \mathcal{F} : A finite collection of disjoint connected semialgebraic subsets $\{C_i\}$ such that each C_i is contained in some semialgebraic sign class of \mathcal{F} . That is, the sign of each $f \in \mathcal{F}$ is **invariant** in each C_i . The collection of connected components of the sign-invariant sets for \mathcal{F} forms a semialgebraic decomposition for \mathcal{F} .

Cell decomposition for \mathcal{F} : A semialgebraic decomposition for \mathcal{F} into finitely many disjoint semialgebraic subsets $\{C_i\}$ called **cells**, such that each cell C_i is homeomorphic to $\mathbb{R}^{\delta(i)}$, $0 \leq \delta(i) \leq n$. $\delta(i)$ is called the **dimension of the cell** C_i , and C_i is called a $\delta(i)$ -cell.

Cellular decomposition for \mathcal{F} : A cell decomposition for \mathcal{F} such that the closure $\overline{C_i}$ of each cell C_i is a union of cells C_j : $\overline{C_i} = \cup_j C_j$.

CONNECTED COMPONENTS OF SEMIALGEBRAIC SETS

A consequence of the Milnor-Thom result [Mil64, Tho65] gives a bound for the number (the zeroth **Betti number**, $B_0(S)$) of connected components of a basic semialgebraic set S : the bound is polynomial in the number m and degree d of the polynomials defining S and singly exponential in the number of variables, n . The current best bound for $B_0(S)$ is due to Pollack and Roy [PR93]: $B_0(S) = O(md)^n$.

Most recent work of Basu ([Bas01], Theorem 4) provides even more precise information about the topological complexity of basic semialgebraic sets through the **higher-order Betti numbers**. While $B_0(S)$ measures the number of connected components of the semialgebraic set S , intuitively, $B_i(S)$ ($i > 0$) measures the number of i -dimensional holes in S . The following bound on B_i is due to Basu:

THEOREM 33.2.2

Let $S \subseteq \mathbb{R}^n$ be the set defined by the conjunction of m inequalities,

$$\begin{aligned} f_i(x_1, \dots, x_n) &\geq 0, \quad f_i \in \mathbb{R}[x_1, \dots, x_n], \\ \text{degree}(f_i) &\leq d, \quad 1 \leq i \leq m, \end{aligned}$$

contained in a variety $V(Q)$ of real dimension n' , and

$$\text{degree}(Q) \leq d.$$

Then,

$$B_i(S) \leq m^{n'-i} O(d)^n.$$

A key problem in computational real algebraic geometry is to compute at least one point in each connected component of each nonempty sign assignment. An elegant solution to this problem is obtained by Collins's **cylindrical algebraic decomposition** (CAD), which is, in fact, a cell decomposition; see [Section 33.5](#) below. A related question is to provide a finitary representation for these sample points, e.g., each coordinate of the sample point may be a *real algebraic number*.

Currently, the best algorithm computing a finite set of points of bounded size that intersects *every connected component* of each nonempty sign condition is due to Basu et al. [BPR98] and has an arithmetic time-complexity of $m(m/n)^n d^{O(n)}$.

33.3 REAL ALGEBRAIC NUMBERS

Real algebraic numbers are real roots of rational univariate polynomials and provide finitary representation for some of the basic objects (e.g., sample points). Furthermore, we note that (1) real algebraic numbers have effective finitary representation, (2) field operations and polynomial evaluation on real algebraic numbers are efficiently (polynomially) computable, and (3) conversions among various representations of real algebraic numbers are efficiently (polynomially) computable. The key machinery used in describing and manipulating real algebraic numbers relies upon techniques based on the Sturm-Sylvester theorem, Thom's lemma, resultant construction, and various bounds for real root separation.

GLOSSARY

Real algebraic number: A real root α of a univariate polynomial $p(t) \in \mathbb{Z}[t]$ with integer coefficients.

Polynomial for α : A univariate polynomial p such that α is a real root of p .

Minimal polynomial of α : A univariate polynomial p of minimal degree defining α as above.

Degree of a nonzero real algebraic number: The degree of its minimal polynomial. By convention, the degree of the 0 polynomial is $-\infty$.

OPERATIONS ON REAL ALGEBRAIC NUMBERS

Note that if α and β are real algebraic numbers, then so are $-\alpha$, α^{-1} (assuming $\alpha \neq 0$), $\alpha + \beta$, and $\alpha \cdot \beta$. These facts can be constructively proved using the algebraic properties of a resultant construction.

THEOREM 33.3.1

The real algebraic numbers form a field.

A real algebraic number α can be represented by a polynomial for α and a component that identifies the root. There are essentially three types of information that may be used for this identification: *order* (where we assume the real roots are indexed from left to right), *sign* (by a vector of signs), or *interval* (an interval that contains exactly one root).

A classical technique due to Sturm and Sylvester shows how to compute the number of real roots of a univariate polynomial $p(t)$ in an interval $[a, b]$. One important use of this classical theorem is to compute a sequence of relatively small (nonoverlapping) intervals that isolate the real roots of p .

GLOSSARY

Sturm sequence of a pair of polynomials $p(t)$ and $q(t) \in \mathbb{R}[t]$:

$$\overline{\text{STURM}}(p, q) = \left\langle \hat{r}_0(t), \hat{r}_1(t), \dots, \hat{r}_s(t) \right\rangle,$$

where

$$\begin{aligned} \hat{r}_0(t) &= p(t) \\ \hat{r}_1(t) &= q(t) \\ &\vdots \\ \hat{r}_{i-1}(t) &= \hat{q}_i(t) \hat{r}_i(t) - \hat{r}_{i+1}(t), \quad \deg(\hat{r}_{i+1}) < \deg(\hat{r}_i) \\ &\vdots \\ \hat{r}_{s-1}(t) &= \hat{q}_s(t) \hat{r}_s(t). \end{aligned}$$

Number of variations in sign of a finite sequence \bar{c} of real numbers: Number of times the entries change sign when scanned sequentially from left to right; denoted $\text{Var}(\bar{c})$.

For a vector of polynomials $\bar{P} = \langle p_1(t), \dots, p_m(t) \rangle$ and a real number a :

$$\text{Var}_a(\bar{P}) = \text{Var}(\bar{P}(a)) = \text{Var}(\langle p_1(a), \dots, p_m(a) \rangle).$$

Formal derivative: $p'(t) = D(p(t))$, where $D: \mathbb{R}[t] \rightarrow \mathbb{R}[t]$ is the (formal) derivative map, taking t^n to nt^{n-1} and $a \in \mathbb{R}$ (a constant) to 0.

STURM-SYLVESTER THEOREM

THEOREM 33.3.2 *Sturm-Sylvester Theorem* [Stu35, Syl53]

Let $p(t)$ and $q(t) \in \mathbb{R}[t]$ be two real univariate polynomials. Then, for any interval $[a, b] \subseteq \mathbb{R} \cup \{\pm\infty\}$ (where $a < b$):

$$\text{Var}[\bar{P}]_a^b = c_p[q > 0]_a^b - c_p[q < 0]_a^b,$$

where

$$\begin{aligned} \bar{P} &\triangleq \overline{\text{STURM}}(p, p'q), \\ \text{Var}[\bar{P}]_a^b &\triangleq \text{Var}_a(\bar{P}) - \text{Var}_b(\bar{P}), \end{aligned}$$

and $c_p[\mathcal{P}]_a^b$ counts the number of distinct real roots (without counting multiplicity) of p in the interval (a, b) at which the predicate \mathcal{P} holds.

Note that if we take $S_p \triangleq \overline{\text{STURM}}(p, p')$ (i.e., $q = 1$) then

$$\begin{aligned} \text{Var}[S_p]_a^b &= c_p[\text{True}]_a^b - c_p[\text{False}]_a^b \\ &= \# \text{ of distinct real roots of } p \text{ in } (a, b). \end{aligned}$$

COROLLARY 33.3.3

Let $p(t)$ and $q(t)$ be two polynomials with coefficients in a real closed field K . For any interval $[a, b]$ as before, we have

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_p[q = 0]_a^b \\ c_p[q > 0]_a^b \\ c_p[q < 0]_a^b \end{bmatrix} = \begin{bmatrix} \text{Var}[\overline{\text{STURM}}(p, p')]_a^b \\ \text{Var}[\overline{\text{STURM}}(p, p'q)]_a^b \\ \text{Var}[\overline{\text{STURM}}(p, p'q^2)]_a^b \end{bmatrix}.$$

These identities as well as some related algorithmic results (the so-called BKR-algorithm) are based on results of Ben-Or et al. [BKR86] and their extensions by others. Using this identity, it is a fairly simple matter to decide the sign conditions of a single univariate polynomial q at the roots of a univariate polynomial p . It is possible to generalize this idea to decide the sign conditions of a sequence of univariate polynomials $q_0(t), q_1(t), \dots, q_n(t)$ at the roots of a single polynomial

$p(t)$ and hence give an efficient (both sequential and parallel) algorithm for the decision problem for Tarski sentences involving univariate polynomials. Further applications in the context of general decision problems are described below.

GLOSSARY

Fourier sequence of a real univariate polynomial $p(t)$ of degree n :

$$\overline{\text{FOURIER}}(p) = \langle p^{(0)}(t) = p(t), p^{(1)}(t) = p'(t), \dots, p^{(n)}(t) \rangle,$$

where $p^{(i)}$ is the i th derivative of p with respect to t .

Sign-invariant region of \mathbb{R} determined by a sign sequence \bar{s} with respect to $\overline{\text{FOURIER}}(p)$: The region $R(\bar{s})$ with the property that $\xi \in R(\bar{s})$ if and only if $\text{sgn}(p^{(i)}(\xi)) = s_i$.

THOM'S LEMMA

LEMMA 33.3.4 *Thom's Lemma* [Tho65]

Every nonempty sign-invariant region $R(\bar{s})$ (determined by a sign sequence \bar{s} with respect to $\overline{\text{FOURIER}}(p)$) must be connected, i.e., consists of a single interval.

Let $\text{sgn}_\xi(\overline{\text{FOURIER}}(p))$ be the sign sequence obtained by evaluating the polynomials of $\overline{\text{FOURIER}}(p)$ at ξ . Then as an immediate corollary of Thom's lemma, we have:

COROLLARY 33.3.5

Let ξ and ζ be two real roots of a real univariate polynomial $p(t)$ of positive degree $n > 0$. Then $\xi = \zeta$, if

$$\text{sgn}_\xi(\overline{\text{FOURIER}}(p')) = \text{sgn}_\zeta(\overline{\text{FOURIER}}(p')).$$

REPRESENTATION OF REAL ALGEBRAIC NUMBERS

Let $p(t)$ be a univariate polynomial of degree d with integer coefficients. Assume that the distinct real roots of $p(t)$ have been enumerated as follows:

$$\alpha_1 < \alpha_2 < \dots < \alpha_{j-1} < \alpha_j = \alpha < \alpha_{j+1} < \dots < \alpha_l,$$

where $l \leq d = \deg(p)$. Then we can represent any of its roots uniquely and in a finitary manner.

GLOSSARY

Order representation of an algebraic number: A pair consisting of its polynomial p and its index j in the monotone sequence enumerating the real roots of p : $\langle \alpha \rangle_o = \langle p, j \rangle$. Example: $\langle \sqrt{2} + \sqrt{3} \rangle_o = \langle x^4 - 10x^2 + 1, 4 \rangle$.

Sign representation of an algebraic number: A pair consisting of its polynomial p and a sign sequence \bar{s} representing the signs of its Fourier sequence evaluated at the root: $\langle \alpha \rangle_s = \langle p, \bar{s} = \text{sgn}_\alpha(\text{FOURIER}(p')) \rangle$. Example: $\langle \sqrt{2} + \sqrt{3} \rangle_s = \langle x^4 - 10x^2 + 1, (+1, +1, +1) \rangle$. The validity of this representation follows easily from Thom's lemma.

Interval representation of an algebraic number: A triple consisting of its polynomial p and the two endpoints of an isolating interval, (l, r) ($l, r \in \mathbb{Q}, l < r$), containing only α : $\langle \alpha \rangle_i = \langle p, l, r \rangle$. Example: $\langle \sqrt{2} + \sqrt{3} \rangle_i = \langle x^4 - 10x^2 + 1, 3, 7/2 \rangle$.

33.4 UNIVARIATE DECOMPOSITION

In the one-dimensional case, a semialgebraic set is the union of finitely many intervals whose endpoints are real algebraic numbers. For instance, given a set of univariate defining polynomials:

$$\mathcal{F} = \left\{ f_i(x) \in \mathbb{Q}[x] \mid i = 1, \dots, m \right\},$$

we may enumerate all the real roots of the f_i 's (i.e., the real roots of the single polynomial $F = \prod f_i$) as

$$-\infty < \xi_1 < \xi_2 < \dots < \xi_{i-1} < \xi_i < \xi_{i+1} < \dots < \xi_s < +\infty,$$

and consider the following finite set \mathcal{K} of elementary intervals defined by these roots:

$$\begin{aligned} & [-\infty, \xi_1], \quad [\xi_1, \xi_1], \quad (\xi_1, \xi_2), \quad \dots, \\ & (\xi_{i-1}, \xi_i), \quad [\xi_i, \xi_i], \quad (\xi_i, \xi_{i+1}), \quad \dots, \quad [\xi_s, \xi_s], \quad (\xi_s, +\infty]. \end{aligned}$$

Note that \mathcal{K} is, in fact, a cellular decomposition for \mathcal{F} . Any semialgebraic set S defined by \mathcal{F} is simply the union of a subset of elementary intervals in \mathcal{K} . Furthermore, for each interval $C \in \mathcal{K}$, we can compute a sample point α_C as follows:

$$\alpha_C = \begin{cases} \xi_1 - 1, & \text{if } C = [-\infty, \xi_1); \\ \xi_i, & \text{if } C = [\xi_i, \xi_i]; \\ (\xi_i + \xi_{i+1})/2, & \text{if } C = (\xi_i, \xi_{i+1}); \\ \xi_s + 1, & \text{if } C = (\xi_s, +\infty]. \end{cases}$$

Now, given a first-order formula involving a single variable, its validity can be checked by evaluating the associated univariate polynomials at the sample points. Using the algorithms for representing and manipulating real algebraic numbers, we see that the bit complexity of the decision algorithm is bounded by $(Lmd)^{O(1)}$. The resulting cellular decomposition has no more than $2md + 1$ cells.

Using variants of the theorem due to Ben-Or et al. [BKR86], Thom's lemma, and some results on parallel computations in linear algebra, one can show that this univariate decision problem is “well-parallelizable,” i.e., the problem is solvable by uniform circuits of bounded depth and polynomially many “gates” (simple processors).

33.5 MULTIVARIATE DECOMPOSITION

A straightforward generalization of the standard univariate decomposition to higher dimensions is provided by Collins's cylindrical algebraic decomposition [Col75]. In order to represent a semialgebraic set $S \subseteq \mathbb{R}^n$, we may assume recursively that we can construct a cell decomposition of its projection $\pi(S) \subseteq \mathbb{R}^{n-1}$ (also a semialgebraic set), and then decompose S as a union of the *sectors* and *sections* in the cylinders above each cell of the projection, $\pi(S)$. This also leads to a cell decomposition of S . One can further assign an algebraic sample point in each cell of S recursively in a straightforward manner.

If \mathcal{F} is a set of polynomials defining the semialgebraic set $S \subseteq \mathbb{R}^n$, then at no additional cost, we may in fact compute a cell decomposition for \mathcal{F} using the procedure described above. Such a decomposition leads to a *cylindrical algebraic decomposition* for \mathcal{F} .

GLOSSARY

Cylindrical algebraic decomposition (CAD): A recursively defined cell decomposition of \mathbb{R}^n for \mathcal{F} . The decomposition is a cellular decomposition if the set of defining polynomials \mathcal{F} satisfies certain nondegeneracy conditions.

In the recursive definition, the cells of n -dimensional CAD are constructed from an $(n-1)$ -dimensional CAD: Every $(n-1)$ -dimensional CAD cell C' has the property that the distinct real roots of \mathcal{F} over C' vary continuously as a function of the points of C' .

Moreover, the following quantities remain invariant over a $(n-1)$ -dimensional cell: (1) the total number of complex roots of each polynomial of \mathcal{F} ; (2) the number of distinct complex roots of each polynomial of \mathcal{F} ; and (3) the total number of common complex roots of every distinct pair of polynomials of \mathcal{F} .

These conditions can be expressed by a set $\Phi(\mathcal{F})$ of at most $O(md)^2$ polynomials in $n - 1$ variables, obtained by considering *principal subresultant coefficients* (PSC's). Thus, they correspond roughly to *resultants* and *discriminants*, and ensure that the polynomials of \mathcal{F} do not intersect or “fold” in a cylinder over an $(n-1)$ -dimensional cell. The polynomials in $\Phi(\mathcal{F})$ are each of degree no more than d^2 .

More formally, an \mathcal{F} -sign-invariant cylindrical algebraic decomposition of \mathbb{R}^n is:

- **BASE CASE:** $n = 1$. A univariate cellular decomposition of \mathbb{R}^1 as in the previous section.
- **INDUCTIVE CASE:** $n > 1$. Let \mathcal{K}' be a $\Phi(\mathcal{F})$ -sign-invariant CAD of \mathbb{R}^{n-1} . For each cell $C' \in \mathcal{K}'$, define an *auxiliary polynomial* $g_{C'}(x_1, \dots, x_{n-1}, x_n)$ as the product of those polynomials of \mathcal{F} that do not vanish over the $(n-1)$ -dimensional cell, C' . The real roots of the auxiliary polynomial $g_{C'}$ over C' give rise to a finite number (perhaps zero) of semialgebraic continuous functions, which partition the cylinder $C' \times (\mathbb{R} \cup \{\pm\infty\})$ into finitely many \mathcal{F} -sign-invariant “slices.” The auxiliary polynomials are of degree no larger than md .

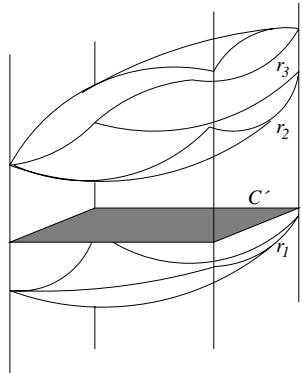


FIGURE 33.5.1

Sections and sectors “slicing” the cylinder over a lower dimensional cell.

Assume that the polynomial $g_{C'}(p', x_n)$ has l distinct real roots for each $p' \in C'$: $r_1(p'), r_2(p'), \dots, r_l(p')$, each r_i being a continuous function of p' . The following sectors and sections are cylindrical over C' (see Figure 33.5.1):

$$\begin{aligned} C_0^* &= \left\{ \langle p', x_n \rangle \mid p' \in C' \wedge x_n \in [-\infty, r_1(p')] \right\}, \\ C_1 &= \left\{ \langle p', x_n \rangle \mid p' \in C' \wedge x_n \in [r_1(p'), r_1(p')] \right\}, \\ C_1^* &= \left\{ \langle p', x_n \rangle \mid p' \in C' \wedge x_n \in (r_1(p'), r_2(p')) \right\}, \\ &\vdots \\ C_l^* &= \left\{ \langle p', x_n \rangle \mid p' \in C' \wedge x_n \in (r_l(p'), +\infty] \right\}. \end{aligned}$$

The n -dimensional CAD is thus the union of all the sections and sectors computed over the cells of the $(n-1)$ -dimensional CAD.

A straightforward recursive algorithm to compute a CAD follows from the above description.

CYLINDRICAL ALGEBRAIC DECOMPOSITION

If we assume that the dimension n is a fixed constant, then the preceding cylindrical algebraic decomposition algorithm is polynomial in $m = |\mathcal{F}|$ and $d = \deg(\mathcal{F})$. However, the algorithm can be easily seen to be doubly-exponential in n as the number of polynomials produced at the lowest dimension is $(md)^{2^{O(n)}}$, each of degree no larger than $d^{2^{O(n)}}$. The number of cells produced by the algorithm is also *doubly-exponential*. This bound can be seen to be tight by a result due to Davenport and Heintz [DH88], and is related to their lower bound for the quantifier elimination problem (Section 33.1).

CONSTRUCTING SAMPLE POINTS

Cylindrical algebraic decomposition provides a sample point in every sign-invariant connected component for \mathcal{F} . However, the total number of sample points generated is doubly-exponential, while the number of connected components of all sign

conditions is only singly-exponential. In order to avoid this high complexity (both algebraic and combinatorial) of a CAD, many recent techniques for constructing sample points use a single projection to a line instead of a sequence of cascading projections. For instance, if one chooses a height function carefully then one can easily enumerate its critical points and then associate at least two such critical points to every connected component of the semialgebraic set. From these critical points, it will be possible to create at least one sample point per connected component. Using Bézout's bound, it is seen that only a singly-exponential number of sample points is created, thus improving the complexity of the underlying algorithms.

However, in order to arrive at the preceding conclusion using critical points, one requires certain genericity conditions that can be achieved by symbolically deforming the underlying semialgebraic sets. These infinitesimal deformations can be handled by extending the underlying field to a field of *Puiseux series*. Many of the significant complexity improvements based on these techniques have been due to a careful choice of the symbolic perturbation schemes which results in keeping the number of perturbation variables small.

33.6 ALGORITHMIC APPROACHES

COLLINS'S APPROACH

The decision problem for the first-order theory of reals can be solved easily using a cylindrical algebraic decomposition. First consider the existential problem for a sentence with only existential quantifiers,

$$(\exists \mathbf{x}^{[0]}) [\psi(\mathbf{x}^{[0]})].$$

This sentence is true if and only if there is a $q \in C$, a sample point in the cell C ,

$$q = \alpha^{[0]} = \langle \alpha_1, \dots, \alpha_n \rangle \in \mathbb{R}^n,$$

such that $\psi(\alpha^{[0]})$ is true. Thus we see that the decision problem for the purely existential sentence can be solved by simply evaluating the matrix ψ over the finitely many sample points in the associated CAD. This also implies that the existential quantifiers could be replaced by finitely many disjunctions ranging over all the sample points. Note that the same arguments hold for any semialgebraic decomposition with at least one sample point per sign-invariant connected component.

In the general case, one can describe the decision procedure by means of a search process that proceeds *only on* the coordinates of the sample points in the cylindrical algebraic decomposition. This follows because a sample point in a cell acts as a representative for any point in the cell as far as the sign conditions are concerned.

Consider a Tarski sentence

$$(Q_1 \mathbf{x}^{[1]})(Q_2 \mathbf{x}^{[2]}) \cdots (Q_\omega \mathbf{x}^{[\omega]}) [\psi(\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[\omega]})],$$

with \mathcal{F} the set of polynomials appearing in the matrix ψ . Let \mathcal{K} be a cylindrical algebraic decomposition of \mathbb{R}^n for \mathcal{F} . Since the cylindrical algebraic decomposition

produces a sequence of decompositions:

$$\mathcal{K}_1 \text{ of } \mathbb{R}^1, \mathcal{K}_2 \text{ of } \mathbb{R}^2, \dots, \mathcal{K}_n \text{ of } \mathbb{R}^n,$$

such that the each cell $C_{i-1,j}$ of \mathcal{K}_i is cylindrical over some cell C_{i-1} of \mathcal{K}_{i-1} , the search progresses by first finding cells C_1 of \mathcal{K}_1 such that

$$(\mathcal{Q}_2 x_2) \cdots (\mathcal{Q}_n x_n) [\psi(\alpha_{C_1}, x_2, \dots, x_n)] = \text{True}.$$

For each C_1 , the search continues over cells C_{12} of \mathcal{K}_2 cylindrical over C_1 such that

$$(\mathcal{Q}_3 x_3) \cdots (\mathcal{Q}_n x_n) [\psi(\alpha_{C_1}, \alpha_{C_{12}}, x_3, \dots, x_n)] = \text{True},$$

etc. Finally, at the bottom level the truth properties of the matrix ψ are determined by evaluating at all the coordinates of the sample points.

This produces a tree structure, where each node at the $(i-1)$ th level corresponds to a cell $C_{i-1} \in \mathcal{K}_{i-1}$ and its children correspond to the cells $C_{i-1,j} \in \mathcal{K}_i$ that are cylindrical over C_{i-1} . The leaves of the tree correspond to the cells of the final decomposition $\mathcal{K} = \mathcal{K}_n$. Because we only have finitely many sample points, the universal quantifiers can be replaced by finitely many conjunctions and the existential quantifiers by disjunctions. Thus, we label every node at the $(i-1)$ th level “AND” (respectively, “OR”) if \mathcal{Q}_i is a universal quantifier \forall (respectively, \exists) to produce a so-called AND-OR tree. The truth of the Tarski sentence is thus determined by simply evaluating this AND-OR tree.

A quantifier elimination algorithm can be devised by a similar reasoning and a slight modification of the CAD algorithm described above.

NEW APPROACHES USING CRITICAL POINTS

In order to avoid the cascading projections inherent in Collins’s algorithm, the new approaches employ a single projection to a one-dimensional set by using critical points in a manner described above. As before, we start with a sentence with only existential quantifiers,

$$(\exists \mathbf{x}^{[0]}) [\psi(\mathbf{x}^{[0]})].$$

Let $\mathcal{F} = \{f_1, \dots, f_m\}$ be the set of polynomials appearing in the matrix ψ .

Under certain genericity conditions, it is possible to produce a set of sample points such that every sign-invariant connected component of the decomposition induced by \mathcal{F} contains at least one such point. Furthermore, these sample points are described by a set of univariate polynomial sequences, where each sequence is of the form

$$p(t), q_0(t), q_1(t), \dots, q_n(t),$$

and encodes a sample point $(\frac{q_1(\alpha)}{q_0(\alpha)}, \dots, \frac{q_n(\alpha)}{q_0(\alpha)})$. Here α is a root of p . Now the decision problem for the existential theory can be solved by deciding the sign conditions of the sequence of univariate polynomials

$$f_1(q_1/q_0, \dots, q_n/q_0), \dots, f_m(q_1/q_0, \dots, q_n/q_0),$$

at the roots of the univariate polynomial $p(t)$. Note that we have now reduced a multivariate problem to a univariate problem and can solve this by the BKR approach.

In order to keep the complexity reasonably small, one needs to ensure that the number of such sequences is small and that these polynomials are of low degree. Assuming that the polynomials in \mathcal{F} are in general position, one can achieve this and compute the polynomials p and q_i (for example, by the u -resultant method in Renegar's algorithm).

If the genericity conditions are violated, one needs to symbolically deform the polynomials and carry out the computations on these polynomials with additional perturbation parameters. The Basu-Pollack-Roy (BPR) algorithm differs from Renegar's algorithm primarily in the manner in which these perturbations are made so that their effect on the algorithmic complexity is controlled.

Next consider an existential Tarski formula of the form

$$(\exists \mathbf{x}^{[0]}) [\psi(\mathbf{y}, \mathbf{x}^{[0]})],$$

where \mathbf{y} represents the free variables. If we carry out the same computation as before over the ambient field $\mathbb{R}(\mathbf{y})$, we get a set of *parameterized* univariate polynomial sequences, each of the form

$$p(\mathbf{y}, t), q_0(\mathbf{y}, t), q_1(\mathbf{y}, t), \dots, q_n(\mathbf{y}, t).$$

For a fixed value of \mathbf{y} , say $\bar{\mathbf{y}}$, the polynomials

$$p(\bar{\mathbf{y}}, t), q_0(\bar{\mathbf{y}}, t), q_1(\bar{\mathbf{y}}, t), \dots, q_n(\bar{\mathbf{y}}, t)$$

can then be used as before to decide the truth or falsity of the sentence

$$(\exists \mathbf{x}^{[0]}) [\psi(\bar{\mathbf{y}}, \mathbf{x}^{[0]})].$$

Also, one may observe that the *parameter space* \mathbf{y} can be partitioned into semialgebraic sets so that all the necessary information can be obtained by computing at sample values $\bar{\mathbf{y}}$.

This process can be extended to ω blocks of quantifiers, by replacing each block of variables by a finite number of cases, each involving only one new variable; the last step uses a CAD method for these ω -many variables.

33.7 APPLICATIONS

Computational real algebraic geometry finds applications in robotics, vision, computer-aided design, geometric theorem proving, and other fields. Important problems in robotics include the kinematic modeling, the inverse kinematic solution, the computation of the workspace and workspace singularities, and the planning of an obstacle-avoiding motion of a robot in a cluttered environment—all arising from the algebro-geometric nature of robot kinematics. In solid modeling, graphics, and vision, almost all applications involve the description of surfaces, the generation of various auxiliary surfaces such as blending and smoothing surfaces, the classification of various algebraic surfaces, the algebraic or geometric invariants associated with a surface, the effect of various affine or projective transformations of a surface, the description of surface boundaries, and so on.

To give examples of the nature of the solutions demanded by various applications, we discuss a few representative problems from robotics, engineering, and computer science.

ROBOT MOTION PLANNING

Given the initial and desired configurations of a robot (composed of rigid subparts) and a set of obstacles, find a collision-free continuous motion of the robot from the initial configuration to the final configuration.

The algorithm proceeds in several steps. The first step translates the problem to **configuration space**, a parameter space modeled as a low-dimensional algebraic manifold (assuming that the obstacles and the robot subparts are bounded by piecewise algebraic surfaces). The second step computes the set of configurations that avoid collisions and produces a semialgebraic description of this so-called “free space” (subspaces of the configuration space). Since the initial and final configurations correspond to two points in the configuration space, we simply have to test whether they lie in the same connected component of the free space. If so, they can be connected by a piecewise algebraic path. Such a path gives rise to an obstacle-avoiding motion of the robot(s). This path planning process can be carried out using Collins’s CAD [SS83], yielding an algorithm with doubly-exponential time complexity (Theorem 40.1.1). A singly-exponential time complexity algorithm (the *roadmap algorithm*) has been devised by Canny [Can88a] (Theorem 40.1.2). The main idea of Canny’s algorithm is to determine a one-dimensional connected subset (called the “roadmap”) of each connected component of the free space. Once these roadmaps are available, they can be used to link up two points in the same connected component. The main geometric idea is to construct roadmaps starting from the critical sets of some projection function. The basic roadmap algorithm has been improved and extended by several researchers over the last decade (Heintz et al. [HRS90], Gournay and Risler [GR93], Grigor’ev and Vorobjov [Gri88, GV88], and Canny [Can88a, Can90]).

OFFSET SURFACE CONSTRUCTION IN SOLID MODELING

*Given a polynomial $f(x, y, z)$, whose zeros define an algebraic surface in three-dimensional space, compute the envelope of a family of spheres of radius r whose centers lie on the surface f . Such a surface is called a (two-sided) **offset surface** of f .*

Let $p = \langle x, y, z \rangle$ be a point on the offset surface and $q = \langle u, v, w \rangle$ be a **footprint** of p on f ; that is, q is the point at which a normal from p to f meets f . Let $\vec{t}_1 = \langle t_{1,1}, t_{1,2}, t_{1,3} \rangle$ and $\vec{t}_2 = \langle t_{2,1}, t_{2,2}, t_{2,3} \rangle$ be two linearly independent tangent vectors to f at the point q . Then, we see that the system of polynomial equations

$$\begin{aligned}(x - u)^2 + (y - v)^2 + (z - w)^2 - r^2 &= 0, \\ f(u, v, w) &= 0, \\ (x - u)t_{1,1} + (y - v)t_{1,2} + (z - w)t_{1,3} &= 0, \\ (x - u)t_{2,1} + (y - v)t_{2,2} + (z - w)t_{2,3} &= 0,\end{aligned}$$

describes a surface in the (x, y, z, u, v, w) six-dimensional space, which, when projected into the three-dimensional space with coordinates (x, y, z) , gives the offset surface in an implicit form. The offset surface is computed by simply eliminating the variables u, v, w from the preceding set of equations.

This approach (the **envelope method**) of computing the offset surface has

several problematic features: the method does not deal with self-intersection in a clean way and, sometimes, generates additional points not on the offset surface. For a discussion of these and several other related problems in solid modeling, see [Hof89] and [Chapter 56](#) of this Handbook.

GEOMETRIC THEOREM PROVING

Given a geometric statement consisting of a finite set of hypotheses and a conclusion,

$$\begin{aligned} \text{Hypotheses} &: f_1(x_1, \dots, x_n) = 0, \dots, f_r(x_1, \dots, x_n) = 0 \\ \text{Conclusion} &: g(x_1, \dots, x_n) = 0 \end{aligned}$$

decide whether the conclusion $g = 0$ is a consequence of the hypotheses $((f_1 = 0) \wedge \dots \wedge (f_r = 0))$.

Thus we need to determine whether the following universally quantified first-order sentence holds:

$$(\forall x_1, \dots, x_n) \left[((f_1 = 0) \wedge \dots \wedge (f_r = 0)) \Rightarrow g = 0 \right].$$

One way to solve the problem is by first translating it into the form: decide if the following existentially quantified first-order sentence is unsatisfiable:

$$(\exists x_1, \dots, x_n, z) \left[(f_1 = 0) \wedge \dots \wedge (f_r = 0) \wedge (gz - 1) = 0 \right].$$

When the underlying domain is assumed to be the field of real numbers, then we may simply check whether the following multivariate polynomial (in x_1, \dots, x_n, z) has no real root:

$$f_1^2 + \dots + f_r^2 + (gz - 1)^2.$$

If, on the other hand, the underlying domain is assumed to be the field of complex numbers (an algebraically closed field), then other tools from computational algebra are used (e.g., techniques based on Hilbert's Nullstellensatz). In the general setting, some techniques based on Ritt-Wu characteristic sets have proven very powerful. See [Cho88].

For another approach to geometric theorem proving, see [Section 59.4](#).

CONNECTION TO SEMIDEFINITE PROGRAMMING

Checking **global nonnegativity** of a function of several variables occupies a central role in many areas of applied mathematics, e.g., optimization problems with polynomial objectives and constraints, as in quadratic, linear and boolean programming formulations. These problems have been shown to be NP-hard in the most general setting, but do admit good approximations involving polynomial-time computable relaxations. (See Parrilo [Par00]).

Provide checkable conditions or procedure for verifying the validity of the proposition

$$F(x_1, \dots, x_n) \geq 0, \quad \forall x_1, \dots, x_n,$$

where F is a multivariate polynomial in the ring of multivariate polynomials over the reals, $\mathbb{R}[x_1, \dots, x_n]$.

An obvious necessary condition for F to be globally nonnegative is that it has even degree. On the other hand, a rather simple sufficient condition for a real-valued polynomial $F(x)$ to be globally nonnegative is the existence of a **sum-of-squares decomposition**:

$$F(x_1, \dots, x_n) = \sum_i f_i^2(x_1, \dots, x_n), \quad f_i(x_1, \dots, x_n) \in \mathbb{R}[x_1, \dots, x_n].$$

Thus one way to solve the global nonnegativity problem is by finding a sum-of-squares decomposition. Note that since there exist globally nonnegative polynomials not admitting a sum-of-squares decomposition (e.g., the Motzkin form $x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$), the procedure suggested below does not give a solution to the problem in all situations.

The procedure can be described as follows: express the given polynomial $F(x_1, \dots, x_n)$ of degree $2d$ as a quadratic form in all the monomials of degree less than or equal to d :

$$F(x_1, \dots, x_n) = z^T Q z, \quad z = [1, x_1, \dots, x_n, x_1 x_2, \dots, x_n^d],$$

where Q is a constant matrix to be determined. If the above quadratic form can be solved for a positive semidefinite Q , then $F(x_1, \dots, x_n)$ is globally nonnegative. Since the variables in z are not algebraically independent, the matrix Q is not unique, but lives in an affine subspace. Thus, we need to determine if the intersection of this affine subspace and the positive semidefinite matrix cone is nonempty. This problem can be solved by a **semidefinite programming** feasibility problem:

$$\begin{aligned} \text{trace}(zz^T Q) &= F(x_1, \dots, x_n), \\ Q &\succeq 0. \end{aligned}$$

The dimensions of the matrix inequality are $\binom{n+d}{d} \times \binom{n+d}{d}$ and is polynomial for fixed number of variables (n) or fixed degree (d). Thus our question reduces to efficiently solvable semidefinite programming (SDP) problems.

33.8 SOURCES AND RELATED MATERIAL

SURVEYS

[Mis93]: A textbook for algorithmic algebra covering Gröbner bases, characteristic sets, resultants, and real algebra. Chapter 8 gives many details of the classical results in computational real algebra.

[CJ98]: An anthology of key papers in computational real algebra and real algebraic geometry. Contains reprints of the following papers cited in this chapter: [BPR98, Col75, Ren91, Tar51].

[AB88]: A special issue of the *J. Symbolic Comput.* on computational real algebraic geometry. Contains several papers ([DH88, Gri88, GV88] cited here) addressing many key research problems in this area.

- [BR90]: A very accessible and self-contained textbook on real algebra and real algebraic geometry.
- [BCR98]: A self-contained textbook on real algebra and real algebraic geometry.
- [HRR91]: A survey of many classical and recent results in computational real algebra.
- [Cha94]: A survey of the connections among computational geometry, computational algebra, and computational real algebraic geometry.
- [Tar51]: Primary reference for Tarski's classical result on the decidability of elementary algebra.
- [Col75]: Collins's work improving the complexity of Tarski's solution for the decision problem [Tar51]. Also, introduces the concept of cylindrical algebraic decomposition (CAD).
- [Ren91]: A survey of some recent results, improving the complexity of the decision problem and quantifier elimination problem for the first-order theory of reals. This is mostly a summary of the results first given in a sequence of papers by Renegar [Ren92a,b,c].
- [Lat91]: A comprehensive textbook covering various aspects of robot motion planning problems and different solution techniques. [Chapter 5](#) includes a description of the connection between the motion planning problem and computational real algebraic geometry.
- [SS83]: A classic paper in robotics showing the connection between the robot motion planning problem and the connectivity of semialgebraic sets using CAD. Contains several improved algorithmic results in computational real algebra.
- [Can88a]: Gives a singly-exponential time algorithm for the robot motion planning problem and provides complexity improvement for many key problems in computational real algebra.
- [Hof89]: A comprehensive textbook covering various computational algebraic techniques with applications to solid modeling. Contains a very readable description of Gröbner bases algorithms.
- [Cho88]: A monograph on geometric theorem proving using Ritt-Wu characteristic sets. Includes computer-generated proofs of many classical geometric theorems.

RELATED CHAPTERS

- [Chapter 47: Algorithmic motion planning](#)
- [Chapter 48: Robotics](#)
- [Chapter 56: Solid modeling](#)
- [Chapter 59: Geometric applications of the Grassmann-Cayley algebra](#)

REFERENCES

- [AB88] D. Arnon and B. Buchberger, editors, *Algorithms in Real Algebraic Geometry*. Special Issue: *J. Symbolic Comput.*, 5(1–2), 1988.

- [Bas99] S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *J. Assoc. Comput. Mach.*, 46:537–555, 1999.
- [Bas01] S. Basu. On different bounds on different Betti numbers. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 288–292, 2001.
- [BPR96] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. Assoc. Comput. Mach.*, 43:1002–1045, 1996.
- [BPR98] S. Basu, R. Pollack, and M.-F. Roy. A new algorithm to find a point in every cell defined by a family of polynomials. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition, Texts Monographs Symbol. Comput.*, Springer-Verlag, Vienna, 1998.
- [BR90] R. Benedetti and J.-J. Risler. *Real Algebraic and Semi-Algebraic Sets*. Hermann, Paris, 1990.
- [BKR86] M. Ben-Or, D. Kozen, and J. Reif. The complexity of elementary algebra and geometry. *J. Comput. Syst. Sci.*, 32:251–264, 1986.
- [BCR98] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*. Springer-Verlag, Berlin, 1998. (Also in French, *Géométrie Algébrique Réelle*. Springer-Verlag, Berlin, 1987.)
- [Can88a] J.F. Canny. *The Complexity of Robot Motion Planning*. Ph.D. Thesis, MIT, Cambridge, 1988.
- [Can88b] J.F. Canny. Some algebraic and geometric computations in PSPACE. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 460–467, 1988.
- [Can90] J.F. Canny. Generalized characteristic polynomials. *J. Symbolic Comput.*, 9:241–250, 1990.
- [Can93] J.F. Canny. Improved algorithms for sign determination and existential quantifier elimination. *Comput. J.*, 36:409–418, 1993.
- [CJ98] B.F. Caviness and J.R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts Monographs Symbol. Comput.*, Springer-Verlag, Vienna, 1998.
- [Cha94] B. Chazelle. Computational geometry: A retrospective. In *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 75–94, 1994.
- [Cho88] S.C. Chou. *Mechanical Geometry Theorem Proving*. Reidel, Dordrecht, 1988.
- [Col75] G. Collins. Quantifier elimination for real closed fields by Cylindrical Algebraic Decomposition. *Second GI Conf. on Automata Theory Formal Lang.*, volume 33 of *Lecture Notes in Comput. Sci.*, pages 134–183. Springer-Verlag, Berlin, 1975. Also in [CJ98].
- [DH88] J.H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *J. Symbolic Comput.*, 5:29–35, 1988.
- [GR93] L. Gourney and J.-J. Risler. Construction of roadmaps in semi-algebraic sets. *Appl. Algebra Engrg. Comm. Comput.*, 4:239–252, 1993.
- [Gri88] D. Grigor'ev. The complexity of deciding Tarski algebra. *J. Symbolic Comput.*, 5:65–108, 1988.
- [GV88] D. Grigor'ev and N.N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *J. Symbolic Comput.*, 5:37–64, 1988.
- [GV92] D. Grigor'ev and N.N. Vorobjov. Counting connected components of a semialgebraic set in subexponential time. *Comput. Complexity*, 2:133–186, 1992.
- [HRR91] J. Heintz, T. Recio, and M.-F. Roy. Algorithms in real algebraic geometry and applications to computational geometry. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 137–164. Amer. Math. Soc., Providence, 1991.
- [HRS89] J. Heintz, M.-F. Roy, and P. Solernó. On the complexity of semi-algebraic sets. In *Proc. Internat. Fed. Info. Process. 89*, pages 293–298. North-Holland, San Francisco, 1989.

- [HRS90] J. Heintz, M.-F. Roy, and P. Solernó. Sur la complexité du principe de Tarski-Seidenberg. *Bull. Soc. Math. France*, 118:101–126, 1990.
- [Hof89] C.M. Hoffmann. *Geometric and Solid Modeling*. Morgan Kaufmann, San Mateo, 1989.
- [Lat91] J.-C. Latombe. *Robot Motion Planning*. Kluwer, Boston, 1991.
- [Mil64] J. Milnor. On the Betti numbers of real algebraic varieties. *Proc. Amer. Math. Soc.*, 15:275–280, 1964.
- [Mis93] B. Mishra. *Algorithmic Algebra*. In *Texts Monographs Comput. Sci.*, Springer-Verlag, New York, 1993.
- [Par00] P.A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. Thesis, California Institute of Technology, 2000.
- [PR93] R. Pollack and M.-F. Roy. On the number of cells defined by a set of polynomials. *C.R. Acad. Sci. Paris Sér. I Math.*, 316:573–577, 1993.
- [Ren91] J. Renegar. Recent progress on the complexity of the decision problem for the reals. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 287–308. Amer. Math. Soc., Providence, 1991. Also in [CJ98].
- [Ren92a] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: Part I. *J. Symbolic Comput.*, 13:255–299, 1992.
- [Ren92b] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: Part II. *J. Symbolic Comput.*, 13:301–327, 1992.
- [Ren92c] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals: Part III. *J. Symbolic Comput.*, 13:329–352, 1992.
- [SS83] J.T. Schwartz and M. Sharir. On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.
- [Sei74] A. Seidenberg. Constructions in algebra. *Trans. Amer. Math. Soc.*, 197:273–313, 1974.
- [Stu35] C. Sturm. Mémoire sur la Résolution des Équations Numériques. *Mém. Savants Etrangers*, 6:271–318, 1835.
- [Syl53] J.J. Sylvester. On a theory of the syzygetic relations of two rational integral functions, comprising an application to the theory of Sturm's functions, and that of the greatest algebraic common measure. *Philos. Trans. Roy. Soc. London*, 143:407–548, 1853.
- [Tar51] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. Univ. of California Press, Berkeley, 1951. Also in [CJ98].
- [Tho65] R. Thom. Sur l'homologie des variétés réelles. In S.S. Chern, editor, *Differential and Combinatorial Topology*, Princeton Univ. Press, pages 255–265, 1965.

34 POINT LOCATION

Jack Snoeyink

INTRODUCTION

A basic question for computer applications that employ geometric structures (e.g., for computer graphics, geographic information systems, robotics, and databases) is: “Where am I?” Given a set of disjoint geometric objects, the ***point-location problem*** asks for the object containing a query point. Instances of the problem vary in the dimension and type of objects and whether the set is static or dynamic. Solutions vary in preprocessing time, space used, and query time.

Point location has inspired several techniques for structuring geometric data, which we survey in this chapter. We begin with point location in one dimension (Section 34.1) or in one polygon (Section 34.2). In two dimensions, we look at how techniques of persistence, fractional cascading, trapezoid graphs, or hierarchical triangulations can lead to optimal methods for point location in static subdivisions (Section 34.3), at the current best methods for dynamic subdivisions (Section 34.4), and at practical methods (Section 34.5). There are fewer results on point location in higher dimensions; these we mention in (Section 34.6).

34.1 ONE-DIMENSIONAL POINT LOCATION

The simplest nontrivial instance of point location is list searching. The objects are points $x_1 \leq \dots \leq x_n$ on the real line, presented in arbitrary order, and the intervals between them, (x_i, x_{i+1}) for $1 \leq i < n$. The answer to a query q is the name of the object containing q .

The list-searching problem already illustrates several aspects of general point location problems.

GLOSSARY

Preprocessing/queries: If one assumes that many queries will ask for the same input, then resources can profitably be spent building data structures to facilitate the search. Three resources are commonly analyzed:

Query time: Computation time to answer a single query, given a point location data structure. Usually a worst-case upper bound, expressed as a function of the number of objects in the structure, n .

Preprocessing time: Time required to build a point location structure for n objects.

Space: Memory used by the point location structure for n objects.

Decomposable problem: A problem whose answer can be obtained from the

answers to the same problem on the sets of an arbitrary partition of the input [Ben79, BS80]. As initially stated, one-dimensional point location is not decomposable—taking subsets of the points gives different intervals. If, however, we choose to name each interval by its lower endpoint, then we can report (the lower endpoint of) the interval containing a query from the intervals in the subproblems—simply report the highest “lower endpoint” from the answers to queries on subsets. Most point location problems can be made decomposable and are easier to solve in such a form.

Dynamic point location: Maintaining a location data structure as points are inserted and deleted. The one-dimensional point location structures can be made dynamic without changing their asymptotic performances.

Randomized point location: Data structures whose preprocessing algorithms may make random choices in an attempt to avoid poor performance caused by pathological input data. Preprocessing and query times are reported as expectations over these random choices. Randomized algorithms make no assumptions on the input or query distributions. They often use a sample to obtain information about the input distribution, and can achieve good expected performance with simple algorithms.

Entropy bounds: If the probability of a query falling in region i is known to be p_i , then Shannon entropy $H = \sum_i -p_i \log_2(p_i)$ is a lower bound for expected query time, where here the expectation is over the query probability distribution.

LIST SEARCH AS ONE-DIMENSIONAL POINT LOCATION

Table 34.1.1 reports query time, preprocessing time, and space for five search methods. Linear search requires no additional data structure if the problem is decomposable. Binary search trees or randomized search trees [SA96, Pug90] require a total order and an ability to do comparisons. An adversary argument shows that these comparison-based query algorithms require $\Omega(\log n)$ comparisons. If the probability distribution for queries is known, then the lower bound on expected query time is H , and expected $H + 2$ can be achieved by weight-balanced trees [Meh77].

If the points are restricted to integers $[1, \dots, U]$, then van Emde Boas has shown how hashing techniques can be applied in stratified search trees to answer a query in $O(\log \log U)$ time. A useful method in practice is to partition the input range into b equal-sized buckets, and to answer a query by searching the bucket containing the query.

TABLE 34.1.1 List search as one-dimensional point location.

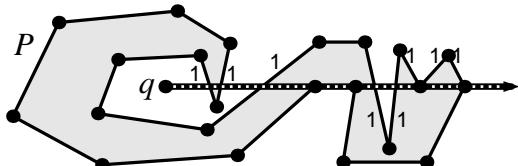
TECHNIQUE	QUERY	PREPROC	SPACE
Linear search	$O(n)$	none	data only
Binary search	$O(\log n)$	$O(n \log n)$	$O(n)$
Randomized tree	$O(\log n)$ expec	$O(n \log n)$ expec	$O(n)$
Weight-balance tree	$H+2$ expec	$O(n \log n)$	$O(n)$
van Emde Boas tree [vEKZ77]	$O(\log \log U)$	$O(n)$ expec	$O(n)$
Bucketing	$O(n)$	$O(n + b)$	$O(n + b)$

34.2 POINT-IN-POLYGON

The second simplest form of point location is to determine whether a query point q lies inside a given n -sided polygon P [Hai94]. Without preprocessing the polygon, one may use parity of the winding or crossing numbers: count intersections of a ray from q with the boundary of polygon P . Point q is inside P iff the number is odd. A query takes $O(n)$ time.

FIGURE 34.2.1

*Counting degenerate crossings:
eight crossings imply $q \notin P$.*



One must count carefully in degenerate cases when the ray passes through a vertex or edge of P . When the ray is horizontal, as in Figure 34.2.1, then edges of P can be considered to contain their lower but not their upper endpoints. Edges inside the ray can be ignored. This is consistent with the count obtained by perturbing the ray infinitesimally upward. Stewart [Ste91] further considered instances in which vertex and edge positions may be imprecise.

To obtain sublinear query times, preprocess the polygon P using the more general techniques of the next sections.

34.3 PLANAR POINT LOCATION: STATIC

Theoretical research has produced a number of planar point location methods that are optimal for comparison-based models: $O(n \log n)$ time to preprocess a planar subdivision with n vertices for $O(\log n)$ time queries using $O(n)$ space. Preprocessing time reduces to linear if the input is given in an appropriate format, and some preprocessing schemes have been parallelized (see [Chapter 42](#)).

We focus on the data structuring techniques used to reach optimality: persistence, fractional cascading, trapezoid graphs, and hierarchical triangulations.

In a planar subdivision, point location can be made decomposable by storing with each edge the name of the face immediately above. If one knows for each subproblem the edge below a query, then one can determine the edge directly below and report the containing face, even for an arbitrary partition into subproblems.

GLOSSARY

Planar subdivision: A partitioning of a region of the plane into point *vertices*, line segment *edges*, and polygonal *faces*.

Size of a planar subdivision: The number of vertices, usually denoted by n .

Euler's relation bounds the numbers of edges $e \leq 3n - 6$ and faces $f \leq 2n - 4$; often the constants are suppressed by saying that the number of vertices, edges, and faces are all $O(n)$.

Monotone subdivision: A planar subdivision whose faces are x -monotone polygons: i.e., the intersection of any face with any vertical line is connected.

Triangulation or trapezoidation: Planar subdivisions whose faces are triangles or whose faces are trapezoids whose parallel sides are all parallel.

Dual graph: A planar subdivision can be viewed as a graph with vertices joined by edges. The dual graph has a node for each face and an arc joining two faces if they share a common edge.

TABLE 34.3.1 A selection of the best static planar point location results for subdivision with n edges. Expectations are over decisions made by the algorithm; averages are over a query distribution with entropy H .

TECHNIQUE	QUERY	PREPROC	SPACE
Slab + persistence [ST86]	$O(\log n)$	$O(n \log n)$	$O(n)$
Separating chain + fractional cascade [EGS86]	$O(\log n)$	$O(n \log n)$	$O(n)$
Optimal query [SA00] + struct. sharing	$\log_2 n + \sqrt{\log_2 n} + \Theta(1)$ $\log_2 n + \sqrt{\log_2 n} + O(\log_2^{1/4} n)$	$O(2^{2\sqrt{\log n}})$ $\exp. O(n \log n)$	$O(2^{2\sqrt{\log n}})$ $\exp. O(n)$
Randomized [Mul90]	expected $O(\log n)$	$\exp. O(n \log n)$	$\exp. O(n)$
Weighted randomized [AMM01b]	average $(5 \ln 2)H + O(1)$	$\exp. O(n \log n)$	$\exp. O(n)$
Optimal entropy [AMM01a]	average $H + o(H)$	$\exp. O(n \log n)$	$O(n \log^* n)$

PLANAR SEPARATOR THEOREM

The first optimal point location scheme is based on Lipton and Tarjan's **planar separator theorem** [LT80] that every planar graph of n nodes has a set of $O(\sqrt{n})$ nodes that partition it into roughly equal pieces. When applied to the dual graph of a planar subdivision, the nodes are a small set of faces that partition the remainder of the faces: simple quadratic-space methods can be used to determine which set of the partition needs to be searched recursively.

The fact that embedded graphs have small separators continues to be important in theoretical work. For example, Goodrich [Goo95] gave a linear-time construction of a family of planar separators in his parallel triangulation algorithm.

SLABS AND PERSISTENCE

By drawing a vertical line through every vertex, as shown in Figure 34.3.1(a), we obtain vertical **slabs** in which point location is almost one-dimensional. Two binary searches suffice to answer a query: one on x -coordinates for the slab containing q , and one on edges that cross that slab. Query time is $O(\log n)$, but space may be quadratic if all edges are stored with the slabs that they cross.

The location structures for adjacent slabs are similar. We could sweep from left to right to construct balanced binary search trees on edges for all slabs: As we sweep over the right endpoint of an edge, we remove the corresponding tree node. As we sweep over the left endpoint of an edge, we add a node. This takes $O(n \log n)$ total time and a linear number of node updates. To store all slabs in linear space, Sarnak and Tarjan [ST86] add to this the idea of *persistence*.

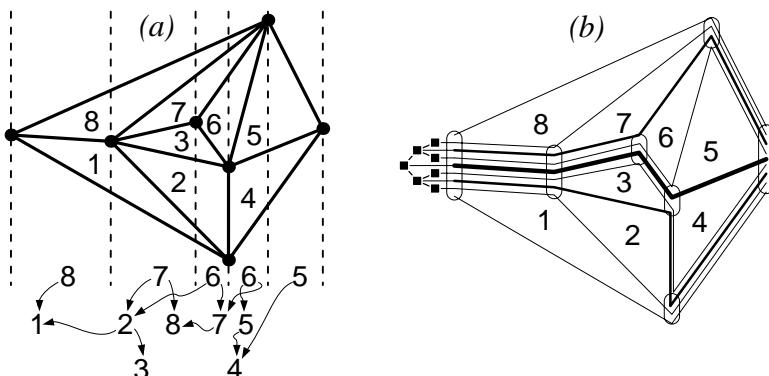
Rather than modifying a node to update the tree, copy the $O(\log n)$ nodes on the path from the root to this node, then modify the copies. This ***node-copying persistence*** preserves the former tree and gives access to a new tree (through the new root) that represents the adjacent slab. The total space for n trees is $O(n \log n)$. Figure 34.3.1(a) provides an illustration. The initial tree contains 8 and 1. (Recall that edges are named by the face immediately above.) Then 2, 3, and 7 are added, 8 is copied during rebalancing, but node 1 is not changed. When 6 is added, 7 is copied in the rebalancing, but the two subtrees holding 1, 2, 3, and 8 are not changed.

Limited node copying reduces the space to linear. Give each node spare left and right pointers and left and right time-stamps. Build a balanced tree for the initial slab. When a pointer is to be modified, use a spare and time-stamp it, if there is a spare available. Future searches can use the time-stamp to determine whether to follow the pointer or the spare. Otherwise, copy the node and modify its ancestor to point to the copy. If the slab location structures are maintained with $O(1)$ rotations per update, then the amortized cost of copying is also $O(1)$ per update.

Preprocessing takes $O(n \log n)$ time to sort by x coordinates and build either persistent data structure. To compare constants with other methods, the data structure has about 12 entries per edge because of extra pointers and copying. Searches take about $4 \log_2 N$ comparisons, where N is the number of edges that can intersect a vertical line; this is because there are two comparisons per node and “ $O(1)$ rotation” tree-balancing routines are balanced only to within a factor of two.

FIGURE 34.3.1

Optimal static methods: (a) Slab (persistent); (b) separating chain (fractional cascading).



SEPARATING CHAINS AND FRACTIONAL CASCADING

If a subdivision is monotone, then its faces can be totally ordered consistent with aboveness; in other words, we can number faces $1, \dots, f$ so that any vertical line encounters lower numbers below higher numbers. The *separating chain* between the faces $< k$ and those $\geq k$ is a monotone chain of edges [LP77]. Figure 34.3.1(b) shows all separating chains for a subdivision; the middle chain, $k = 5$, is shown darkest.

A balanced binary tree of separating chains can be used for point location: if query point q is above chain i and below chain $i + 1$, then q is in face i . To preserve linear space we need to avoid the duplication of edges in chains that can be seen in Figure 34.3.1(b).

Note that the separating chains that contain an edge are defined by consecutive integers; we can store the first and last with each edge. Then form a binary tree in which each subtree stores the separating chains from some interval—at each node, store the edges of the median chain that have not been stored higher in the tree, and recursively store the intervals below and above the median in the left and right subtrees respectively. The root, for example, stores all edges of the middle chain. Since no edge is stored twice, this data structure takes $O(n)$ space.

As we search the tree for a query point q , we keep track of the edges found so far that are immediately above and below q . (Initially, no edges have been found.) Now, the root of the subtree to search is associated with a separating chain. If that chain does not contain one of the edges that we know is above or below q , then we search the x -coordinates of edges stored at the node and find the one on the vertical line through q . We then compare against the separating chain and recursively search the left or right subtree. Thus, this separating chain method [LP77] inspects $O(\log n)$ tree nodes at a cost of $O(\log n)$ each, giving $O(\log^2 n)$ query time.

To reduce the query time, we can use fractional cascading [CG86, EGS86] for efficient search in multiple lists. As we traverse our search tree, at every node we search a list by x -coordinates. We can make all searches after the first take constant time, if we increase the total size of these lists by 50%. Pass every fourth x -coordinate from a child list to its parent, and establish connections so that knowing one's position in the parent list gives one's position in the child to within four nodes.

Preprocessing takes $O(n)$ time on a monotone subdivision; arbitrary planar subdivisions can be made monotone by plane sweep in $O(n \log n)$ time. One can trade off space and query time in fractional cascading, but typical constants are 8 entries per edge for a query time of $4 \log_2 n$.

TRAPEZOID GRAPH METHODS

Preparata's [Pre81] trapezoid method is a simple, practical method that achieves $O(\log n)$ query time at the cost of $O(n \log n)$ space. Its underlying search structure, the *trapezoid graph*, is the basis for important variations: randomized point location in optimal expected time and space, a recursive application giving exact worst-case optimal query time, and average-time point location achieving the entropy bound.

A trapezoid graph is a directed, acyclic graph (DAG) in which each nonleaf node ν is associated with a trapezoid τ_ν whose parallel sides are vertical and whose

To locate the triangle containing a query point q , start by finding the triangle in the coarsest triangulation, at right in Figure 34.3.3. Knowing the hole (shaded) that this triangle came from, one need only replace the missing vertex and check the incident triangles to locate q in the previous, finer triangulation.

Given a triangulation, preprocessing takes $O(n)$ time, but the hidden constants on time and space are large. For example, choosing the independent set by greedily taking vertices in order of increasing degree up to 10 guarantees 1/6th of the vertices [SvK97], which leads to a data structure with $12n$ triangles in which a query could take $35 \log_2 n$ comparisons.

OPEN PROBLEMS

1. Develop a data structure for point location in a planar subdivision that simultaneously achieves linear space and $H + o(H)$ query time.
2. Splay trees [ST85] achieve $O(m + mH)$ query time for m queries (if each element is queried at least once). Can one develop a “self-adjusting” data structure for 2D point location with similar query time?

34.4 PLANAR POINT LOCATION: DYNAMIC

In dynamic planar point location, the subdivision can be updated by adding or deleting vertices and edges. Unlike the static case, algorithms that match the performance of one-dimensional point location have not yet been found. Again, we focus on the data structures used by the best methods, summarized in [Table 34.4.1](#).

GLOSSARY

Updates: A dynamic planar subdivision is most commonly updated by inserting or deleting a vertex or edge. Update time usually refers to the worst-case time for a single insertion or deletion.

Chain insertion/deletion: Some methods support insertion or deletion of a chain of k vertices and edges, so that this is faster than doing k insertions or k deletions.

Vertex expansion/contraction: Updating a planar subdivision by splitting a vertex into two vertices joined by an edge, or the inverse: contracting an edge and merging the two endpoints into one. This operation, supported by the “primal/dual spanning tree” (discussed below), is important for point location in 3D subdivisions.

Amortized update time: When times are reported as amortized, then an individual operation may be expensive, but the total time for k operations, starting from an empty data structure, will take at most k times the amortized bound.

I/O efficient algorithm: An algorithm whose asymptotic number of I/O operations is minimal. Model parameters are problem size N , disk block size B and memory size M , with typically $B \leq \sqrt{M}$. Sorting requires $O((N/B) \log_B N)$ time.

TABLE 34.4.1 Dynamic point location results.

TECHNIQUE	QUERY	UPDATE	SPACE	UPDATES SUPPORTED
Trapezoid method [CT92]	$O(\log n)$	$O(\log^2 n)$	$O(n \log n)$	ins/del vertex & edge
Interval tree [CJ92] with frac casc [BJM94]	$O(\log^2 n)$ $O(\log n \log \log n)$	$O(\log n)$ $O(\log^2 n)$	$O(n)$ $O(n)$	ins/del edge & chain amort del, ins faster
Pr/dual span tree [GT91] amortized	$O(\log^2 n)$ $O(\log n \log \log n)$	$O(\log n)$ $O(1)$	$O(n)$ $O(n)$	$\begin{cases} \text{ins/del edge & chain,} \\ \text{expand/contract vertex} \end{cases}$
Separating chain [PT89] I/O-efficient [AV00]	$O(\log^2 n)$ $O(\log_B^2 N)$	$O(\log^2 n)$ $O(\log_B^2 N)$	$O(n)$ $O(N/B)$	ins/del edge & edge measures I/O blocks read

TRAPEZOID METHOD

Preparata's trapezoid method [Pre81], which stores a binary tree on subdivision edges as described in Section 34.3, can be made dynamic. It preserves its optimal $O(\log n)$ query time, as well as retaining its suboptimal $O(n \log n)$ space.

To allow updates in $O(\log^2 n)$ time, Chiang and Tamassia [CT92, CPT96] store the tree on subdivision edges in a *link-cut* tree [ST83], which supports in $O(\log n)$ time the operation of linking two trees by adding an arc, and the inverse, cutting an arc to make two trees.

DYNAMIC INTERVAL TREE

An *interval tree* storing segments can be defined recursively: the root stores segments that cross a given vertical line ℓ ; segments to the left (right) are stored in an interval tree that is the left (right) child of the root. To locate a query point q , one must search down the interval tree, answering the following subproblem at $O(\log n)$ nodes: Given a set of segments S that intersect a common line ℓ , which segment is immediately below q ?

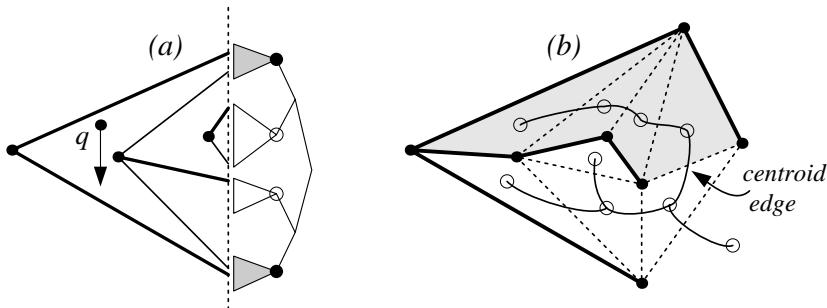
Cheng and Janardan [CJ92] solve this subproblem by a priority-tree search, which allows them to use the interval tree for dynamic point location. In an interval tree node, store the segments in a binary search tree ordered along ℓ , and store in each subtree a pointer to the “priority segment” with endpoint farthest left of ℓ . (Priority must also be stored on the right.) At each level of the search tree, only two candidate subtrees may contain the segment below q —the ones whose priority segments are immediately above and below q . Figure 34.4.1(a) illustrates a case in which the search continues in the two shaded subtrees.

Performing this search in each node of the interval tree leads to $O(\log^2 n)$ query time using $O(n)$ space. Constants are moderate, with only 4 or 5 entries per edge and 6 comparisons per search step. Updates take $O(\log n)$ time with larger constants; they must maintain tree balance and segment priorities.

Baumgarten et al. [BJM94] use fractional cascading on blocks of $O(\log^2 n)$ segments in each interval tree node to speed up queries to $O(\log n \log \log n)$, at the cost of slowing insertions to $O(\log n \log \log n)$ amortized, and deletions to $O(\log^2 n)$. This is a surprising development because fractional cascading requires a global order that is difficult to establish in interval tree techniques.

FIGURE 34.4.1

Dynamic methods: (a) Priority search (interval tree); (b) primal/dual spanning tree.



PRIMAL/DUAL SPANNING TREE

A monotone subdivision has a *monotone spanning tree* in which all root-to-leaf paths are monotone. Each edge not in the tree closes a cycle and defines a monotone polygon.

In any planar graph whose faces are simple polygons, the duals of edges not in the spanning tree form a dual spanning tree of faces, as in Figure 34.4.1(b). Goodrich and Tamassia [GT91] use a centroid decomposition of the dual tree to guide comparisons with monotone polygons in the primal tree. The centroid edge, which breaks the dual tree into two nearly-equal pieces, is indicated in Figure 34.4.1(b). The primal edge creates the shaded monotone polygon; if the query is inside then we recursively explore the corresponding piece of the dual tree. Using link-cut trees, the centroid decomposition can be maintained in logarithmic time per update, giving a dynamic point-location structure with $O(\log^2 n)$ query time.

In the static setting, fractional cascading can turn this into an optimal point location method. Dynamic fractional cascading can be used to reduce the dynamic query time and to obtain $O(1)$ amortized update time.

The dual nature of the structure supports insertion and deletion of dual edges, which correspond to expansion and contraction of vertices. These are needed to support static 3D point location via persistence. Furthermore, a k -vertex monotone chain can be inserted/deleted in $O(\log n + k)$ time.

SEPARATING CHAINS

The separating chain method was the first to be made fully dynamic [PT89]. Although both its asymptotics and its constant factors are larger than other methods, it has been made I/O-efficient [AV00]. This is an impressive theoretical accomplishment, but simpler algorithms that assume that the input is somewhat evenly distributed in the plane will be more practical.

OPEN PROBLEMS

1. Improve dynamic planar point location to simultaneously attain $O(n)$ space

and $O(\log n)$ query and update time, or establish a lower bound.

2. Can persistent data structures be made dynamic? The fact that data are copied seems to work against maintaining a data structure under insertions and deletions.
-

34.5 PLANAR POINT LOCATION: COMMON PRACTICE

Programming complexity and nonnegligible asymptotic constants mean that optimal point location techniques are used less than might be expected. See [TV01] for a study of geometric algorithm engineering that uses point location schemes as its example.

PICK HARDWARE

Graphic workstations employ special “pick hardware” that draws objects on the screen and returns a list of objects that intersect a query pixel. The hardware imposes a minimum time of about 1/30th of a second on a pick operation, but hundreds of thousands of polygons may be considered in this time.

BUCKETING AND SPATIAL INDEX STRUCTURES

Because data in practical applications tend to be evenly distributed, bucketing techniques are far more effective [AEI⁺85, EKA84] than worst-case analysis would predict. For problems in two and three dimensions, a uniform grid will often trim data to a manageable size.

Adaptive data structures for more general spatial indexing, such as k -d trees, quadtrees, BANG files, R-trees, and their relatives [Sam90], can be used as filters for point location—these techniques are common in databases and geographic information systems.

SUBDIVISION WALKING

Applications that store planar subdivisions with their adjacency relations, such as geographic information systems, can walk through the regions of the subdivision from a known position p to the query q .

To walk a subdivision with $O(n)$ edges, compute the intersections of \overline{pq} with the current region and determine if q is inside. If not, let q' denote the intersection point closest to q . Advance to the region incident to q' that contains a point in the interior of $\overline{q'q}$ and repeat. In the worst case, this walk takes $O(n)$ time. The application literature typically claims $O(\sqrt{n})$ time, which is the average number of intersections with a line under the assumption that vertices and edges of the subdivision are evenly distributed. When combined with bucketing or hierarchical data structures (for example, maintaining a regular grid or quadtree with known positions and starting from the closest to answer a query), walking is an effective, practical location method.

For triangulations, the algorithm walking \overline{pq} is easy to implement. Guibas and Stolfi's [GS85] incremental Delaunay triangulation uses an even simpler walk from edge to edge, but this depends on an acyclicity theorem (Sections 20.4 and 22.1) that does not hold for arbitrary triangulations. A robust walk should remember its starting point and handle vertices on the traversed segment as if they had been perturbed consistently.

There have been several recent analyses of ***Jump & Walk*** schemes in triangulations [DPT02, DLM99, DMZ98]. Devroye et al. [DLM99] show expected query times of $O(n^{1/4})$ for a scheme that keeps $n^{1/4}$ points with known locations, and walks from the nearest to find a query. In their experiments, the combination of a 2D search tree with walking performed the best. Devillers' hierarchical Delaunay [Dev02] uses an idea that applies to other triangulations as well: maintain a hierarchy of triangulations using small samples (e.g., 3% [Dev98]) and then walk from a vertex located in one level to find a vertex in the next level. This is implemented in the CGAL library [BDP⁺02].

34.6 LOCATION IN HIGHER DIMENSIONS

In higher dimensions, known point location methods do not achieve both linear space and logarithmic query time. Linear space can be attained by relatively straightforward linear search, such as the point-in-polygon test.

Logarithmic time, or $O(d \log n)$ time, can be obtained by projection [DL76]: project the $d - 2$ faces of a subdivision to an arrangement in $d - 1$ dimensions and recursively build a point location structure for the arrangement in the projection. Knowing the cell in the projection gives a list of the possible faces that project to that cell, so an additional logarithmic search can return the answer. The worst-case space required is $O(n^{2^d})$.

Because point location is decomposable, batching can trade space for time: preprocessing n/k groups of k facets into structures with $S(k)$ space and $Q(k)$ time gives, in total, $O(nS(k)/k)$ space and $O(nQ(k)/k)$ query time.

Clever ways of batching can lead to better structures. Randomized methods can often reduce the dependence on dimension from doubly- to singly-exponential, since random samples can be good approximations to a set of geometric objects. They can also be used with objects that are implicitly defined.

We should mention that convex polyhedra can be preprocessed using the Dobkin-Kirkpatrick hierarchy (Section 34.3) so that the point-in-convex-polyhedron test does take $O(n)$ space and $O(\log n)$ query time.

THREE-DIMENSIONAL POINT LOCATION

Dynamic location structures can be used for static spatial point location in one higher dimension by employing persistence. If one swept a plane through a subdivision of three-space into polyhedra, one could see the intersection as a dynamic planar subdivision in which vertices (intersections of the sweep plane with edges) move along linear trajectories. Whenever the sweep plane passes through a vertex in space, vertices in the plane may join and split.

Goodrich and Tamassia's primal/dual method supports the necessary opera-

tions to maintain a point location structure for the sweeping plane. Using node-copying to make the structures persistent gives an $O(n \log n)$ space structure that can answer queries in $O(\log^2 n)$ time. Preprocessing takes $O(n \log n)$ time.

Devillers et al. [DPT02] tested several approaches to subdivision walking for Delaunay tetrahedralization, and established the practical effectiveness of the hierarchical Delaunay in three dimensions as well.

RECTILINEAR SUBDIVISIONS

Restricting attention to rectilinear (orthogonal) subdivisions permits better results via data structures for orthogonal range search. The *skewer tree*, a multidimensional interval tree, gives static point location among n rectangular prisms with $O(n)$ space and $O(\log^{d-1} n)$ query time after $O(n \log n)$ preprocessing [EHH86].

In dimensions two and three, stratified trees and perfect hashing [DKM⁺94] can be used to obtain $O((\log \log U)^{d-1})$ query time in a fixed universe $[1, \dots, U]$, or $O(\log n)$ query time in general. Iacono and Langerman [IL00] use “justified hyperrectangles” to obtain $O(\log \log U)$ query times in every dimension d , but the space and preprocessing time, which are $O(fn \log \log U)$ and $O(fn \log U \log \log U)$, respectively, depend on a *fatness parameter* f that equals the average ratio of the d th power of smallest dimension to volume of all hyperrectangles in the subdivision.

POINT LOCATION AMONG ALGEBRAIC VARIETIES

Chazelle and Sharir [CS90] consider point location in a general setting, among n algebraic varieties of constant maximum degree b in d -dimensional Euclidean space. They augment Collins’s cylindrical algebraic decomposition to obtain an $O(n^{2^{d-1}})$ -space, $O(\log n)$ -query time structure after $O(n^{2^{d+6}})$ preprocessing. Hidden constants depend on the degrees of projections and intersections, which can be b^{4^d} .

This method provides a general technique to obtain subquadratic solutions to optimization problems that minimize a function $\{F(a, b) \mid a \in A, b \in B\}$, where $F(a, b)$ has a constant-size algebraic description. For a fixed b , F is algebraic in a . Thus, small batches of points from B can be preprocessed in subquadratic time, and each a can be tested against each batch, again in subquadratic time.

OPEN PROBLEMS

1. Find an optimal method for static (or dynamic) point location in a 3D subdivision with n vertices and $O(n)$ faces: $O(n)$ space and $O(\log n)$ query time.
2. In a subdivision of a d -dimensional rectangular prism into n prisms, is there an optimal $O(\log n)$ -query, $O(n)$ -space point location method? The constants hidden by the big- O may depend on d . Under a pointer model of computation, this is already open for $d = 3$.

RANDOMIZED POINT LOCATION

TABLE 34.6.1 Randomized point location in arrangements.

TECHNIQUE	OBJECTS	QUERY	PREPROC	SPACE
Random sample [Cla87]	hyperplanes	$O(c^d \log n)$ exp	$O(n^{d+1+\epsilon})$ exp	$O(n^{d+\epsilon})$
Derandomized [CF94]	hyperplanes	$O(c^d \log n)$	$O(n^{2d+1})$	$O(n^d)$
Random sample [MS91]	dyn hpl $d \leq 4$	$O(\log n)$ exp	$O(n^{d+\epsilon})$ exp	$O(n^{d+\epsilon})$
Epsilon nets [Mei93]	hyperplanes	$O(d^5 \log n)$ exp	$O(n^{d+1+\epsilon})$ exp	$O(n^{d+\epsilon})$

The techniques of Chapter 40 can lead to good point location methods when a random sample of a set of objects can be used to approximate the whole. Arrangements of hyperplanes in dimension d are a good example. A random sample of hyperplanes divides space into cells intersected by few hyperplanes; recursively sampling in each cell gives a point location structure for the arrangement. Table 34.6.1 lists the performance of some randomized point location methods for hyperplanes. Query time can be traded for space by choosing larger random samples.

The randomized incremental construction algorithms of Chapter 40 are simple because they naturally build randomized point location structures along with the objects that they aim to construct [Mul93, Sei93]. These have good “tail bounds” and work well as insertion-only location structures.

Randomized point location structures can be made fully dynamic by lazy deletion and randomized rebuild techniques [dBDS95, MS91]; they maintain good expected performance if random elements are chosen for insertion and deletion. That is, the sequence of insertions and deletions may be specified, but the elements are to be chosen independently of their roles in the data structure.

IMPLICIT POINT LOCATION

In some applications of point location, the objects are not given explicitly. A planar motion planning problem may ask whether a start and a goal point are in the same cell of an arrangement of constraint segments or curves, without having explicit representations of all cells.

Consider a simple example: an arrangement of n lines, which defines nearly n^2 bounded cells. Without storing all cells, we can determine whether two points p and q are in the same cell by preprocessing \sqrt{n} subarrangements of \sqrt{n} lines ($O(n\sqrt{n})$ cells in all) and making sure that p and q are together in each subarrangement. If the lines are put into batches by slope, then within the same asymptotic time, an algorithm can return the pair of lines defining the lowest vertex as a unique cell name.

Implicit location methods are often seen as special cases of range queries (Chapter 36) or vertical ray shooting [Aga91]. Table 34.6.2 lists results on implicit location among line segments, which depend upon tools discussed in Chapters 36, 37, and 40, specifically random sampling, ϵ -net theory, and spanning trees with low stabbing number.

TABLE 34.6.2 Implicit point location results for arrangements of n line segments.

TECHNIQUE	QUERY	PREPROC	SPACE
Span tree lsn [Aga92]	$O(\sqrt{n} \log^2 n)$	$O(n^{3/2} \log^\omega n)$	$O(n \log^2 n)$
Batch sp tree [AvK94]	$O\left((n/\sqrt{s}) \log^2(n/\sqrt{s}) + \log n\right)$	$O\left((sn(\log(n/\sqrt{s}) + 1)^{2/3}\right)$	$n\sqrt{\log n} \leq s \leq n^2$

34.7 SOURCES AND RELATED MATERIAL

SURVEYS

Further references may be found in these surveys.

[Pre90]: A survey of planar point-location algorithms.

[Hai94, Wei94]: Point-in-polygon algorithms in *Graphics Gems IV*, with code.

RELATED CHAPTERS

- Chapter 24: Arrangements
- Chapter 25: Triangulations
- Chapter 26: Polygons
- Chapter 36: Range searching
- Chapter 37: Ray shooting and lines in space
- Chapter 40: Randomized algorithms
- Chapter 42: Parallel algorithms in geometry
- Chapter 49: Computer graphics

REFERENCES

- [AEI⁺85] Ta. Asano, M. Edahiro, H. Imai, M. Iri, and K. Murota. Practical use of bucketing techniques in computational geometry. In G.T. Toussaint, editor, *Computational Geometry*, pages 153–195. Elsevier North-Holland, Amsterdam, 1985.
- [Aga91] P.K. Agarwal. Geometric partitioning and its applications. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: Papers from the DIMACS Special Year*. Amer. Math. Soc., Providence, 1991.
- [Aga92] P.K. Agarwal. Ray shooting and other applications of spanning trees with low stabbing number. *SIAM J. Comput.*, 21:540–570, 1992.
- [AMM01a] S. Arya, T. Malamatos, and D.M. Mount. Entropy-preserving cuttings and space-efficient planar point location. In *Proc. 12th ACM-SIAM Sympos. Disc. Alg.*, pages 256–261, 2001.
- [AMM01b] S. Arya, T. Malamatos, and D.M. Mount. A simple entropy-based algorithm for planar point location. In *Proc. 12th ACM-SIAM Sympos. Disc. Alg.*, pages 262–268, 2001.

- [AV00] L. Arge and J. Vahrenhold. I/O-efficient dynamic planar point location. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 191–200, 2000.
- [AvK94] P.K. Agarwal and M. van Kreveld. Implicit point location in arrangements of line segments, with an application to motion planning. *Internat. J. Comput. Geom. Appl.*, 4:369–383, 1994.
- [BDP⁺02] J.-D. Boissonnat, O. Devillers, S. Pion, M. Teillaud, and M. Yvinec. Triangulations in cgal. *Comp. Geom. Theory Appl.*, 22(1–3):5–19, 2002.
- [Ben79] J.L. Bentley. Decomposable searching problems. *Inform. Process. Lett.*, 8:244–251, 1979.
- [BJM94] H. Baumgarten, H. Jung, and K. Mehlhorn. Dynamic point location in general subdivisions. *J. Algorithms*, 17:342–380, 1994.
- [BS80] J.L. Bentley and J.B. Saxe. Decomposable searching problems I: Static-to-dynamic transformations. *J. Algorithms*, 1:301–358, 1980.
- [CF94] B. Chazelle and J. Friedman. Point location among hyperplanes and unidirectional ray-shooting. *Comput. Geom. Theory Appl.*, 4:53–62, 1994.
- [CG86] B. Chazelle and L.J. Guibas. Fractional cascading: I. A data structuring technique. *Algorithmica*, 1:133–162, 1986.
- [CJ92] S.W. Cheng and R. Janardan. New results on dynamic planar point location. *SIAM J. Comput.*, 21:972–999, 1992.
- [Cla87] K.L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.
- [CPT96] Y.-J. Chiang, F.P. Preparata, and R. Tamassia. A unified approach to dynamic point location, ray shooting, and shortest paths in planar maps. *SIAM J. Comput.*, 25:207–233, 1996.
- [CS90] B. Chazelle and M. Sharir. An algorithm for generalized point location and its application. *J. Symbolic Comput.*, 10:281–309, 1990.
- [CT92] Y.-J. Chiang and R. Tamassia. Dynamization of the trapezoid method for planar point location in monotone subdivisions. *Internat. J. Comput. Geom. Appl.*, 2:311–333, 1992.
- [dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.
- [Dev98] O. Devillers. Improved incremental randomized Delaunay triangulation. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 106–115, 1998.
- [Dev02] O. Devillers. The Delaunay hierarchy. *Internat. J. Found. Comput. Sci.*, 13:163–180, 2002.
- [DKM⁺94] M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf der Heide, H. Rohnert, and R.E. Tarjan. Dynamic perfect hashing: upper and lower bounds. *SIAM J. Comput.*, 23:738–761, 1994.
- [DL76] D.P. Dobkin and R.J. Lipton. Multidimensional searching problems. *SIAM J. Comput.*, 5:181–186, 1976.
- [DLM99] L. Devroye, C. Lemaire, and J.-M. Moreau. Fast Delaunay point location with search structures. In *Proc. 11th Canad. Conf. Comput. Geom.*, pages 136–141, 1999.
- [DMZ98] L. Devroye, E.P. Mücke, and B. Zhu. A note on point location in Delaunay triangulations of random points. *Algorithmica*, 22:477–482, 1998.
- [DPT02] O. Devillers, S. Pion, and M. Teillaud. Walking in a triangulation. *Internat. J. Found. Comput. Sci.*, 13:181–199, 2002.

- [EGS86] H. Edelsbrunner, L.J. Guibas, and J. Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Comput.*, 15:317–340, 1986.
- [EHH86] H. Edelsbrunner, G. Haring, and D. Hilbert. Rectangular point location in d dimensions with applications. *Comput. J.*, 29:76–82, 1986.
- [EKA84] M. Edahiro, I. Kokubo, and Ta. Asano. A new point-location algorithm and its practical efficiency—Comparison with existing algorithms. *ACM Trans. Graph.*, 3:86–109, 1984.
- [Goo95] M.T. Goodrich. Planar separators and parallel polygon triangulation. *J. Comput. Syst. Sci.*, 51:374–389, 1995.
- [GS85] L.J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graph.*, 4:74–123, 1985.
- [GT91] M.T. Goodrich and R. Tamassia. Dynamic trees and dynamic point location. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 523–533, 1991.
- [Hai94] E. Haines. Point in polygon strategies. In P. Heckbert, editor, *Graphics Gems IV*, pages 24–46. Academic Press, Boston, 1994.
- [IL00] J. Iacono and S. Langerman. Dynamic point location in fat hyperrectangles with integer coordinates. In *Proc. 12th Canad. Conf. Comput. Geom.*, pages 181–186, 2000.
- [Kir83] D.G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. Comput.*, 12:28–35, 1983.
- [LP77] D.T. Lee and F.P. Preparata. Location of a point in a planar subdivision and its applications. *SIAM J. Comput.*, 6:594–606, 1977.
- [LT80] R.J. Lipton and R.E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9:615–627, 1980.
- [Meh77] K. Mehlhorn. Best possible bounds on the weighted path length of optimum binary search trees. *SIAM J. Comput.*, 6:235–239, 1977.
- [Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Inform. Comput.*, 106:286–303, 1993.
- [MS91] K. Mulmuley and S. Sen. Dynamic point location in arrangements of hyperplanes. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 132–141, 1991.
- [Mul90] K. Mulmuley. A fast planar partition algorithm, I. *J. Symbolic Comput.*, 10(3–4):253–280, 1990.
- [Mul93] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1993.
- [Pre81] F.P. Preparata. A new approach to planar point location. *SIAM J. Comput.*, 10:473–482, 1981.
- [Pre90] F.P. Preparata. Planar point location revisited. *Internat. J. Found. Comput. Sci.*, 1:71–86, 1990.
- [PT89] F.P. Preparata and R. Tamassia. Fully dynamic point location in a monotone subdivision. *SIAM J. Comput.*, 18:811–830, 1989.
- [Pug90] W. Pugh. Skip lists: a probabilistic alternative to balanced trees. *Commun. ACM*, 33:668–676, 1990.
- [SA96] R. Seidel and C.R. Aragon. Randomized search trees. *Algorithmica*, 16:464–497, 1996.
- [SA00] R. Seidel and U. Adamy. On the exact worst case query complexity of planar point location. *J. Alg.*, 37:189–217, 2000.

- [Sam90] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, 1990.
- [Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1:51–64, 1991.
- [Sei93] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 37–68. Springer-Verlag, Berlin, 1993.
- [ST83] D.D. Sleator and R.E. Tarjan. A data structure for dynamic trees. *J. Comput. Syst. Sci.*, 26:362–381, 1983.
- [ST85] D.D. Sleator and R.E. Tarjan. Self-adjusting binary search trees. *J. Assoc. Comput. Mach.*, 32:652–686, 1985.
- [ST86] N. Sarnak and R.E. Tarjan. Planar point location using persistent search trees. *Commun. ACM*, 29:669–679, 1986.
- [Ste91] A.J. Stewart. Robust point location in approximate polygons. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 179–182, 1991.
- [SvK97] J. Snoeyink and M. van Kreveld. Linear-time reconstruction of Delaunay triangulations with applications. In *Proc. Annu. European Sympos. Algorithms*, volume 1284 of *Lecture Notes Comput. Sci.*, pages 459–471. Springer-Verlag, Berlin, 1997.
- [TV01] R. Tamassia and L. Vismara. A case study in algorithm engineering for geometric computing. *Internat. J. Comput. Geom. Appl.*, 11:15–70, 2001.
- [vEKZ77] P. van Emde Boas, R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Math. Syst. Theory*, 10:99–127, 1977.
- [Wei94] K. Weiler. An incremental angle point in polygon test. In P. Heckbert, editor, *Graphics Gems IV*, pages 16–23. Academic Press, Boston, 1994.

35 COLLISION AND PROXIMITY QUERIES

Ming C. Lin and Dinesh Manocha

INTRODUCTION

In a geometric context, a collision or proximity query reports information about the relative configuration or placement of two objects. Some of the common examples of such queries include checking whether two objects overlap in space, or whether their boundaries intersect, or computing the minimum Euclidean separation distance between their boundaries. Hundreds of papers have been published on different aspects of these queries in computational geometry and related areas such as robotics, computer graphics, virtual environments, and computer-aided design. These queries arise in different applications including robot motion planning, dynamic simulation, haptic rendering, virtual prototyping, interactive walkthroughs, computer gaming, and molecular modeling. For example, a large-scale virtual environment, e.g., a walkthrough, creates a model of the environment with virtual objects. Such an environment is used to give the user a sense of presence in a synthetic world and it should make the images of both the user and the surrounding objects feel solid. The objects should not pass through each other, and objects should move as expected when pushed, pulled, or grasped; see Fig. 35.0.1. Such actions require fast and accurate collision detection between the geometric representations of both real and virtual objects. Another example is rapid prototyping, where digital representations of mechanical parts, tools, and machines, need to be tested for interconnectivity, functionality, and reliability. In Fig. 35.0.2, the motion of the pistons within the combustion chamber wall is simulated to check for tolerances and verify the design.

This chapter provides an overview of different queries and the underlying algorithms. It includes algorithms for collision detection and distance queries among convex polytopes (Section 35.1), nonconvex polygonal models (Section 35.2), penetration depth queries (Section 35.3), curved objects (Section 35.4), dynamic queries (Section 35.5), and large environments composed of multiple objects (Section 35.6). Finally, it briefly describes different software packages available to perform some of the queries (Section 35.7).

PROBLEM CLASSIFICATION

Collision Detection: Checks whether two objects overlap in space or their boundaries share at least one common point.

Separation Distance: Length of the shortest line segment joining two sets of points, A and B :

$$\text{dist}(A, B) = \min_{a \in A} \min_{b \in B} |a - b|.$$

FIGURE 35.0.1

A hand reaching toward a chair on a virtual porch, at top. The corresponding image of the user in the real world is shown on the bottom. Darkened finger tips indicate contacts between the user's hand and the virtual chair.



Hausdorff distance: Maximum deviation of one set from the other:

$$\text{haus}(A, B) = \max_{a \in A} \min_{b \in B} |a - b|.$$

Spanning Distance: Maximum distance between the points of two sets:

$$\text{span}(A, B) = \max_{a \in A} \max_{b \in B} |a - b|.$$

Penetration Depth: Minimum distance needed to translate one set to make it disjoint from the other:

$$\text{pen}(A, B) = \text{minimum } \|\mathbf{v}\| \text{ such that } \min_{a \in A} \min_{b \in B} |\mathbf{a} - \mathbf{b} + \mathbf{v}| > 0.$$

There are two forms of collision detection query: **Boolean** and **enumerative**. The Boolean distance query computes whether the two sets have at least one point in common. The enumerative form yields some representation of the intersection set.

There are at least three forms of the distance queries: exact, approximate, and Boolean. The exact form asks for the exact distance between the objects. The approximate form yields an answer within some given error tolerance of the true measure—the tolerance could be specified as a relative or absolute error. The Boolean form reports whether the exact measure is greater or less than a given value. Furthermore, the norm by which distance is defined may be varied. The Euclidean norm is the most common, but in principle other norms are possible, such as the L_1 and L_∞ norms.

Each of these queries can be augmented by adding the element of time. If the trajectories of two objects are known, then the next time can be determined at which a particular Boolean query (collision, separation distance, or penetration) will become TRUE or FALSE. In fact, this “time-to-next-event” query can have exact,

FIGURE 35.0.2

In this virtual prototyping application, the motion of the pistons is simulated to check for tolerances by performing distance queries.

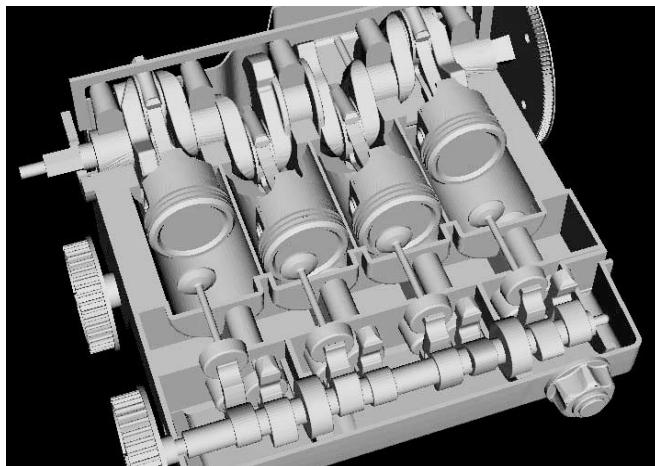


TABLE 35.0.1 Classification of Proximity Queries.

CRITERIA	TYPES
Report	Boolean, exact, approximate, enumerative
Measure	Separation, span, Hausdorff, penetration, collision
Multiplicity	2-body, n -body
Temporality	Static, dynamic
Representation	Polyhedra, convex objects, implicit, parametric, NURBS, quadrics, set-theoretic combinations
Dimension	2,3,d

approximate, and Boolean forms. These queries are called *dynamic queries*, whereas the ones that do not use motion information are called *static queries*. In the case where the motion of an object can not be represented as a closed-form function of time, the underlying application often performs static queries at specific time steps in the application.

These measures, as defined above, apply only to pairs of sets. However, some applications work with many objects, and need to find the proximity information among all or a subset of the pairs. Thus, most of the query types listed above have associated N -body variants.

Finally, the primitives can be represented in different forms. They may be convex polytopes, general polygonal models, curved models represented using parametric or implicit surfaces, set-theoretic combination of objects, etc. Different set of algorithms are known for each representation. A classification of proximity queries based on these criteria is shown in [Table 35.1.1](#).

35.1 CONVEX POLYTOPES

In this section, we give a brief survey of algorithms for collision detection and separation-distance computation between a pair of convex polytopes. A number of algorithms with good asymptotic performance have been proposed. The algorithm with the current best runtime for Boolean collision queries takes $O(\log^2 n)$ time, where n is the number of features [DK90]. It precomputes the Dobkin-Kirkpatrick (DK) hierarchy for each polytope and uses it to perform the query. In practice, three classes of algorithms are commonly used for convex polytopes: linear programming, Minkowski sums, and tracking closest features based on Voronoi diagrams.

LINEAR PROGRAMMING

The problem of checking whether two convex polytopes intersect or not can be posed as a linear programming (LP) problem. In particular, two convex polytopes do not overlap if and only if there exists a separation plane between them. The coefficients of the separation plane equation are treated as unknowns. Linear constraints result by requiring that all vertices of the first polytope lie in one halfspace of this plane and those of the other polytope lie in the other halfspace. The linear programming algorithms are used to check whether there is any feasible solution to the given set of constraints. Given the fixed dimension of the problem, some of the well-known linear programming algorithms (e.g., [Sei90]; cf. Chapter 45) can be used to perform the Boolean collision query in expected linear-time. By caching the last pair of witness points to compute the new separating planes, Chung and Wang [CW96] proposed an iterative method that can quickly update the separating axis or the separating vector in nearly “constant time” in dynamic applications with high motion coherence.

MINKOWSKI SUMS AND CONVEX OPTIMIZATION

Collision and distance queries can be performed based on the Minkowski sum of two objects. It has been shown [CC86] that the minimum separation distance between two objects is the same as the minimum distance from the origin of the *Minkowski sums* of A and $-B$ to the surface of the sums. The Minkowski sum is also referred to as the *translational C-space obstacle* (TCSO). While the Minkowski sum of two convex polytopes can have $O(n^2)$ features [DHKS93], a fast algorithm for separation-distance computation based on convex optimization that exhibits linear-time performance in practice has been proposed by Gilbert et al. [GJK88], also known as the GJK algorithm. It uses pairs of vertices from each object that define simplices within each polytope and a corresponding simplex in the TCSO. Initially the simplex is set randomly and the algorithm refines it using local optimization, until it computes the closest point on the TCSO from the origin of the Minkowski sums. The algorithm assumes that the origin is not inside the TCSO.

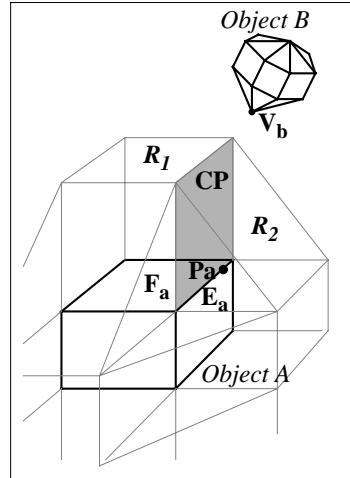


FIGURE 35.1.1

A walk across external Voronoi region of Object A. Vertex V_b of Object B lies in the Voronoi region of E_a .

TRACKING CLOSEST FEATURES USING GEOMETRIC LOCALITY AND MOTION COHERENCE

Lin and Canny [LC91] proposed a distance-computation algorithm between non-overlapping convex polytopes. Often referred to as the LC algorithm, it tracks the closest features between the polytopes. This is the first approach that explicitly takes advantages of motion coherence and geometric locality. The features may correspond to a vertex, face, or an edge on each polytope. It precomputes the external Voronoi region for each polytope. At each time step, it starts with a pair of features and checks whether they are the closest features, based on whether they lie in each other's Voronoi region. If not, it performs a local walk on the boundary of each polytope until it finds the closest features. See Figure 35.1.1. In applications with high motion coherence, the local walk typically takes nearly “constant time” in practice. Typically the number of neighbors for each feature of a polytope is constant and the extent of “local walk” is proportional to the amount of the relative motion undergone by the polytopes.

Mirtich [Mir98] further optimized this algorithm by proposing a more robust variation that avoids some geometric degeneracies during the local walk, without sacrificing the accuracy or correctness of the original algorithm.

Guibas et al. [GHZ99] proposed an approach that exploits both coherence of motion using LC and hierarchical representations by Dobkin and Kirkpatrick [DK90] to reduce the runtime dependency on the amount of the local walks.

Ehmann and Lin [EL00] modified the LC algorithm and used an error-bounded level-of-detail (LOD) hierarchy to perform different types of proximity queries, using the progressive refinement framework (cf. Chapter 54). The implementation of this technique, “multi-level Voronoi Marching,” outperforms the existing libraries for collision detection between convex polytopes. It also uses an initialization technique based on directional lookup using hashing, resembling that of [DZ93].

By taking the similar philosophy as LC, Cameron [Cam97] presented an extension to the basic GJK algorithm by exploiting motion coherence and geometric locality in terms of connectivity between neighboring features. The algorithm tracks

the *witness points*, a pair of points from the two objects that realize the minimum separation distance between them. Rather than starting from a random simplex in the TCSO, the algorithm starts with the witness points from the previous iteration and performs hill climbing to compute a new set of witness points for the current configuration. The running time of this algorithm is a function of the number of refinement steps that the algorithm performs.

TABLE 35.1.1 Algorithms for convex polytopes.

METHOD	FEATURES
DK	$O(\log^2 n)$ query time, collision query only
LP	Linear running time, collision query
GJK	Linear-time behavior in practice, collision and separation-distance queries
LC	Expected constant-time in coherent environments, collision and separation-distance queries

KINETIC DATA STRUCTURES

Recently a new class of algorithms using “kinetic data structures” (or KDS for short) have been proposed for collision detection between moving convex polygons and polyhedra [BEG⁺99, EGSZ99, KSS02] (cf. [Chapter 50](#)). These algorithms are based on the formal framework of KDS to keep track of closest features of polytopes during their motion and exploits motion coherence and geometric locality. The performance of KDS-based algorithms is separation sensitive, and may depend on the amount of the minimum distance between the objects during their motion, relative to their size. The type of motion includes straight-line linear motion, translation along an algebraic trajectory, or algebraic rigid motion (including both rotation and translation).

35.2 GENERAL POLYGONAL MODELS

Algorithms for collision and separation-distance queries between general polygon models can be classified based whether they assume closed polyhedral models, or are represented as a collection of polygons. The latter, also referred to as “polygon soups,” make no assumption related to the connectivity among different faces or whether they represent a closed set.

Some of the most common algorithms for collision detection and separation-distance computation use spatial partitioning or bounding volume hierarchies (BVHs). The spatial subdivisions are a recursive partitioning of the embedding space, whereas bounding volume hierarchies are based on a recursive partitioning of the primitives of an object. These algorithms are based on the divide-and-conquer paradigm. Examples of spatial partitioning hierarchies include k-D trees

and octrees [Sam89], R-trees and their variants [HKM95], cone trees, BSPs [NAT90] and their extensions to multi-space partitions [WG91]. The BVHs use bounding volumes (BVs) to bound or contain sets of geometric primitives, such as triangles, polygons, curved surfaces, etc. In a BVH, BVs are stored at the internal nodes of a tree structure. The root BV contains all the primitives of a model, and children BVs each contain separate partitions of the primitives enclosed by the parent. Leaf node BVs typically contain one primitive. In some variations, one may place several primitives at a leaf node, or use several volumes to contain a single primitive. BVHs are used to perform collision and separation-distance queries. These include sphere-trees [Hub95, Qui94], AABB-trees [BKSS90, HKM95, PML97], OBB-trees [GLM96, BCG⁺96, Got00], spherical shell-trees [KPLM98, KGL⁺98], k -DOP-trees [HKM96, KHM⁺98], SSV-trees [LGLM99], multiresolution hierarchies [OL03], and convex hull-trees [EL01], as shown in Table 35.2.1.

TABLE 35.2.1 Types of bounding volume hierarchies.

NAME	TYPE OF BOUNDING VOLUME
Sphere-tree	Sphere
AABB-tree	Axis-aligned bounding box (AABB)
OBB-Tree	Oriented bounding box (OBB)
Spherical shell-tree	Spherical shell
k -DOP-tree	Discretely oriented polytope defined by k vectors (k -DOP)
SSV-Tree	Swept-sphere volume (SSV)
Convex hull-tree	Convex polytope

COLLISION DETECTION

Collision queries are performed by traversing the BVHs. Two models are compared by recursively traversing their BVHs in tandem. Each recursive step tests whether BVs A and B , one from each hierarchy, overlap. If they do not, the recursion branch is terminated. But if A and B overlap, the enclosed primitives may overlap and the algorithm is applied recursively to their children. If A and B are both leaf nodes, the primitives within them are compared directly.

SEPARATION-DISTANCE COMPUTATION

The structure of the separation-distance query is very similar to the collision query. As the query proceeds, the smallest distance found from comparing primitives is maintained in a variable δ . At the start of the query, δ is initialized to ∞ , or to the distance between an arbitrary pair of primitives. Each recursive call with BVs A and B must determine if some primitive within A and some primitive within B are closer than, and therefore will modify, δ . The call returns trivially if BVs A and B are farther than the current δ , as this precludes any primitives within them being closer than δ . Otherwise the algorithm is applied recursively to its children. For

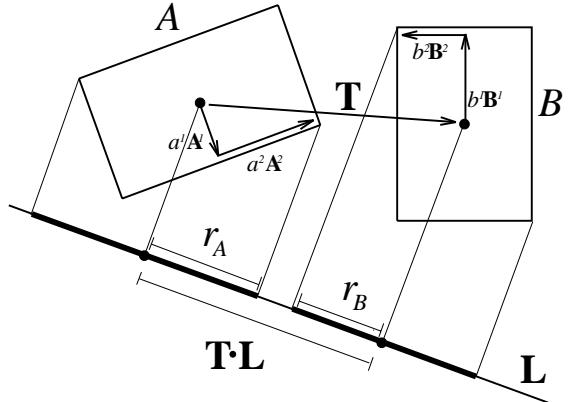


FIGURE 35.2.1

L is a separating axis for OBBs A and B because projection onto L renders them disjoint intervals.

leaf nodes it computes the exact distance between the primitives, and if the new computed distance is less than δ , it updates δ .

To perform an approximate distance query, the distance between BVs A and B is used as a lower limit to the exact distances between their primitives. If this bound prevents δ from being reduced by more than the acceptable tolerance, that recursion branch is terminated.

QUERIES ON BOUNDING VOLUMES

Algorithms for collision detection and distance computation need to perform the underlying queries on the BVHs, including whether two BVs overlap, or computing the separation distance between them. The performance of the overall proximity query algorithm is governed largely by the performance of the subalgorithms used for proximity queries on a pair of BVs.

A number of specialized and highly optimized algorithms have been proposed to perform these queries on different BVs. It is relatively simple to check whether two spheres overlap. Two AABBs can be checked for overlap by comparing their dimensions along the three axes. The separation distance between them can be computed based on the separation along each axis. The overlap test can be easily extended to k -DOPs, where their projections are checked along the k fixed axis [KHM⁺98].

An efficient algorithm to test two OBBs for overlap based on the separating axis theorem (SAT) has been presented in [GLM96, Got00]. It computes the projection of each OBB along 15 axes in 3D. The 15 axes are computed from the face normals of the OBBs (6 face normals) and by taking the cross-products of the edges of the OBBs (9 cross-products). It is shown that two OBBs overlap if and only if their projection along each of these axes overlap. Furthermore, an efficient algorithm that performs overlap tests along each axis has been described. In practice, it can take anywhere from 80 to 240 arithmetic operations to check whether two OBBs overlap. The computation is robust and works well in practice [GLM96]. Figure 35.2.1 shows one of the separating axis tests for two rectangles in 2D.

Algorithms based on different *swept-sphere volumes* (SSVs) have been presented in [LGLM99]. Three types of SSVs are suggested: point swept-sphere (PSS), line swept-sphere (LSS), and a rectangular swept-sphere (RSS). Each BV is formu-

lated by taking the Minkowski sum of the underlying primitive—a point, line, or a rectangle in 3D, respectively—with a sphere. Algorithms to perform collision or distance queries between these BVs can be formulated as computing the distance between the underlying primitives. Larsen et al. [LGLM99] have presented an efficient and robust algorithm to compute distance between two rectangles in 3D (as well rectangles degenerating to lines and points). Moreover, they used priority directed search and primitive caching to lower the number of bounding volume tests for separation-distance computations.

In terms of higher-order bounding volumes, fast overlap tests based on spherical shells have been presented in [KPLM98, KGL⁺98]. Each spherical shell corresponds to a portion of the volume between two concentric spheres. The overlap test between two spherical shells takes into account their structure and reduces to checking whether there is a point contained in a circle that lies in the positive halfplane defined by two lines. The two lines and the circles belong to the same plane.

PERFORMANCE OF BOUNDING VOLUME HIERARCHIES

The performance of BVHs on proximity queries is governed by a number of design parameters, including techniques to build the trees, the maximum number of children per node, and the choice of BV type. An additional design choice is the descent rule. This is the policy for generating recursive calls when a comparison of two BVs does not prune the recursion branch. For instance, if BVs A and B failed to prune, one may recursively compare A with each of the children of B , B with each of the children of A , or each of the children of A with each of the children of B . This choice does not affect the correctness of the algorithm, but may impact the performance. Some of the commonly used algorithms assume that the BVHs are binary trees and each primitive is a single triangle or a polygon. The cost of performing the proximity query is given as [GLM96, LGLM99]:

$$T = N_{bv} \times C_{bv} + N_p \times C_p,$$

where T is the total cost function for proximity queries, N_{bv} is the number of bounding volume pair operations, and C_{bv} is the total cost of a BV pair operation, including the cost of transforming each BV for use in a given configuration of the models, and other per BV-operation overhead. N_p is the number of primitive pairs tested for proximity, and C_p is the cost of testing a pair of primitives for proximity (e.g., overlaps or distance computation).

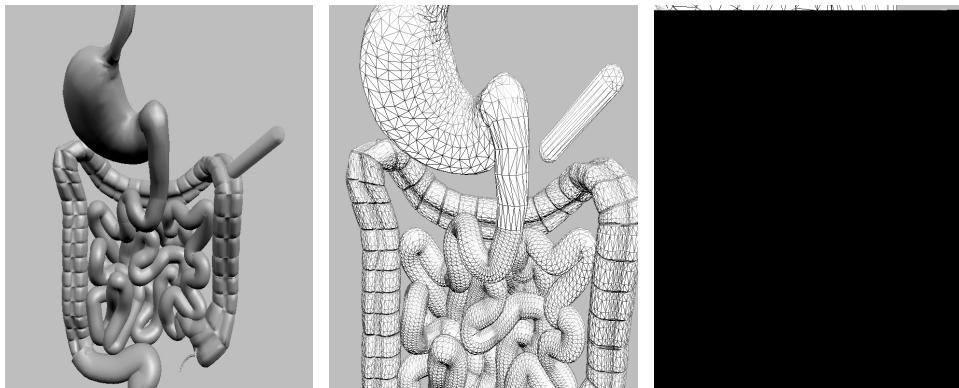
Typically, for tight-fitting bounding volumes, e.g., oriented bounding boxes (OBBs), N_{bv} and N_p are relatively small, whereas C_{bv} is relatively high. In contrast, C_{bv} is low while N_{bv} and N_p may be larger for simple BV types like spheres and axis-aligned bounding boxes (AABBs). Due to these opposing trends, no single BV yields optimum performance for proximity queries in all situations.

35.3 PENETRATION-DEPTH COMPUTATION

In this section, we briefly review ***penetration depth*** (PD) computation algorithms between convex polytopes and general polyhedral models. The PD of two inter-

FIGURE 35.3.1

Penetration depth is applied to virtual exploration of a digestive system using haptic interaction to feel and examine different parts of the model. The distance computation and penetration depth computation algorithms are used for disjoint (D) and penetrating (P) situations, respectively, to compute the forces at the contact areas.



penetrating objects A and B is defined as the minimum translation distance that one object undergoes to make the interiors of A and B disjoint. It can be also defined in terms of the TCSO. When two objects are overlapping, the origin of the Minkowski sum of A and $-B$ is contained inside the TCSO. The penetration depth corresponds to the minimum distance from the origin to the surface of TCSO [Cam97]. PD computation is often used in motion planning [HKL⁺98], contact resolution for dynamic simulation [MZ90, ST96] and force computation in haptic rendering [KOLM02]. Fig. 35.3.1 shows a haptic rendering application of penetration-depth and separation-distance computation. For example, computation of dynamic response in penalty-based methods often needs to perform PD queries for imposing the nonpenetration constraint for rigid body simulation. In addition, many applications, such as motion planning and dynamic simulation, require a continuous distance measure when two (nonconvex) objects collide for a well-posed computation.

Several algorithms for PD computation involve computing Minkowski sums and the closest point on its surface from the origin. The worst-case complexity of the overall PD algorithm is dominated by computing Minkowski sums, which can be $\Omega(n^2)$ for convex polytopes and $\Omega(n^6)$ for general (or nonconvex) polyhedral models [DHKS93]. Given the complexity of Minkowski sums, many approximation algorithms have been proposed in the literature for fast PD estimation.

CONVEX POLYTOPES

Dobkin et al. [DHKS93] proposed a hierarchical algorithm to compute the directional PD using Dobkin and Kirkpatrick polyhedral hierarchy. For any direction d , it computes the directional penetration depth in $O(\log n \log m)$ time for polytopes with m and n vertices. Agarwal et al. [AGHP⁺00] designed a randomized approach to compute the PD values [AGHP⁺00], achieving $O(m^{\frac{3}{4}+\epsilon}n^{\frac{3}{4}+\epsilon} + m^{1+\epsilon} + n^{1+\epsilon})$ expected time for any positive constant ϵ . Cameron [Cam97] presented an exten-

sion to the GJK algorithm [GJK88] to compute upper and lower bounds on the PD between convex polytopes. Bergen further elaborated this idea in an expanding polytope algorithm [Ber01]. The algorithm iteratively improves the result of the PD computation by expanding a polyhedral approximation of the Minkowski sums of two polytopes. Kim et al. [KLM02] presented an incremental algorithm that marches toward a “locally optimal” solution by walking on the surface of the Minkowski sum. The surface of the TCSO is implicitly computed by constructing a local Gauss map and performing a local walk on the polytopes.

POLYHEDRAL MODELS

Algorithms for penetration-depth estimation between general polygonal models are based on discretization of the object space containing the objects, or use of digital geometric algorithms that perform computations on a finite resolution grid. Fisher and Lin [FL01] presented a PD estimation algorithm based on the distance-field computation using the fast marching level-set method. It is applicable to all polyhedral objects as well as deformable models, and it can also check for self-penetration. Hoff et al. [HZLM01, HZLM02] proposed an approach based on performing discretized computations on graphics rasterization hardware. It uses multi-pass rendering techniques for different proximity queries between general rigid and deformable models, including penetration depth estimation. Kim et al. [KLM02] presented a fast approximation algorithm for general polyhedral models using a combination of object-space as well as discretized computations. Given the global nature of the PD problem, it decomposes the boundary of each polyhedron into convex pieces, computes the pairwise Minkowski sums of the resulting convex polytopes and uses graphics rasterization hardware to perform the closest-point query up to a given discretized resolution. The results obtained are refined using a local walking algorithm. To further speed up this computation and improve the estimate, the algorithm uses a hierarchical refinement technique that takes advantage of geometry culling, model simplification, accelerated ray-shooting, and local refinement with greedy walking. The overall approach combines discretized closest-point queries with geometry culling and refinement at each level of the hierarchy. Its accuracy can vary as a function of the discretization error.

OTHER METRICS

Other metrics to characterize the intersection between two objects include the *growth distance* defined by Gilbert and Ong [GO94]. This is a consistent distance measure regardless of whether the objects are disjoint or overlapping; it is differs from the PD between two interpenetrating convex objects.

35.4 SPLINE AND ALGEBRAIC OBJECTS

Most of the algorithms highlighted above are limited to polygonal objects. In many applications of geometric and solid modeling, curved objects whose boundaries are described using rational splines or algebraic equations are used (cf. [Chapter 53](#)).

Algorithms to perform different proximity queries on these objects may be classified by subdivision methods, tracing methods, and analytic methods. See [Pra86, Hof89, Man92] for surveys. Next, we briefly enumerate these methods.

SUBDIVISION METHODS

All subdivision methods for parametric surfaces work by recursively subdividing the domain of the two surface patches in tandem, and examining the spatial relationship between patches [LR80]. Depending on various criteria, the domains are further subdivided and recursively examined, or the given recursion branch is terminated. In all cases, whether it is the intersection curve or the distance function, the solution is known only to some finite precision.

TRACING METHODS

The tracing method begins with a given point known to be on the intersection curve [BFJP87, MC91, KM97]. Then the intersection curve is traced in sufficiently small steps until the edge of the patch is found, or until the curve loops back to itself. In practice, it is easy to check for intersections with a patch boundary, but difficult to know when the tracing point has returned to its starting position. Frequently this is posed as an initial-value differential equations problem [KPW90], or as solving a system of algebraic equations [MC91, KM97, LM97]. At the intersection point on the surfaces, the intersection curve must be mutually orthogonal to the normals of the surfaces. Consequently, the vector field which the tracing point must follow is given by the cross product of the normals.

ANALYTIC METHODS

Analytic methods usually involve implicitizing one of the parametric surfaces—obtaining an implicit representation of the model [SAG84, MC92]. The parametric surface is a mapping from (u, v) -space to (x, y, z) -space, and the implicit surface is a mapping from (x, y, z) -space to \mathbb{R} . Substituting the parametric functions $f_x(u, v), f_y(u, v), f_z(u, v)$ for x, y, z of the implicit function leads to a scalar function in u and v . The locus of roots of this scalar function map out curves in the (u, v) plane which are the preimages of the intersection curve [KPP90, MC91, KM97, Sar83]. Based on its representation as an algebraic plane curve, efficient algorithms have been proposed by a number of researchers [AB88, KM97, KCMh99].

35.5 DYNAMIC QUERIES

In this section we give a brief overview of algorithms used to perform *dynamic queries*. Unlike static queries, which check for collisions or perform separation-distance queries at discrete instances, these algorithms use continuous techniques based on the object motion to compute the time of first collision.

Many algorithms assume that the motion of the objects can be expressed as a closed-form function of time. Cameron [Cam90] has presented algorithms that

pose the problem as interference computation in a 4-dimensional space. Given a parametric representation of each object's boundary as well as its motion, Herzen et al. [HBZ90] presented a collision detection algorithm that subdivides the domain of the surface, including the time dimension. They use Lipschitz conditions, based on bounds on the various derivatives of the mapping, to compute bounds on the extent of the resulting function. The bounds are used to check two objects for overlap. Snyder et al. [Sea93] improved the runtime performance of this algorithm by introducing more conditions that prune the search space for collisions and combined it with interval arithmetic [Moo79].

Other continuous techniques use the object motion to estimate the time of first contact. For prespecified trajectories consisting of a sequence of individual translations and rotations about an arbitrary axis, Boyse [Boy79] presented an algorithm for detecting and analyzing collisions between a moving and a stationary objects. Canny [Can86] described an algorithm for computing the exact points of collision for objects that are simultaneously translating and rotating. It can deal with any path in the space that can be expressed as a polynomial function of time. Given bounds on the maximum velocity and acceleration of the objects are known, Lin [Lin93] presented a scheduling scheme that maintains a priority queue and sorts the object based on approximate time to collision. The approximation is computed from the separation distance as well as from bounds on velocity and acceleration. Redon et al. [RKC00] proposed an algorithm that replaces the unknown motion between two discrete instances by an arbitrary rigid motion. It reduces the problem of computing the time of collision to computing a root of a univariate cubic polynomial.

35.6 LARGE ENVIRONMENTS

Large environments are composed of multiple moving objects. Different methods have been proposed to overcome the bottleneck of $O(n^2)$ pairwise tests in an environment composed of n objects. The problem of performing proximity queries in large environments is typically divided into two parts [Hub95, CLMP95]: the *broad phase*, in which we identify the pair of objects on which we need to perform different proximity queries, and the *narrow phase*, in which we perform the exact pairwise queries. An architecture for multi-body collision detection algorithm is shown in [Figure 35.6.1](#). In this section, we present a brief overview of algorithms used in the broad phase.

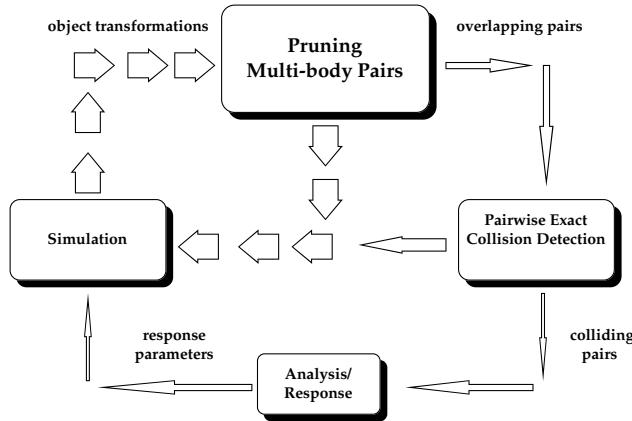
The simplest algorithms for large environments are based on spatial subdivisions. The space is divided into cells of equal volume, and at each instance the objects are assigned to one or more cells. Collisions are checked between all object pairs belonging to each cell. In fact, Overmars presented an efficient algorithm based on hash table to efficiently perform point location queries in fat subdivisions [Ove92] (see also [Chapter 34](#)). This approach works well for sparse environments in which the objects are uniformly distributed through the space. Another approach operates directly on 4D volumes swept out by object motion over time [Cam90]. Efficient algorithms for maintenance and self-collision testing for kinematic chains composed of multiple links have been presented in [LSHL02].

Several algorithms compute an axis-aligned bounding box (AABB) for each

FIGURE 35.6.1

Typically, the object's motion is constrained by collisions with other objects in the simulated environment. Depending on the outcome of the proximity queries, the resulting simulation computes an appropriate response.

Architecture for Multi-body Collision Detection



object, based on their extremal points along each direction. Given n bounding boxes, they check which boxes overlap in space. A number of efficient algorithms are known for the static version of this problem. In 2D, the problem reduces to checking 2D intervals for overlap using interval trees and can be performed in $O(n \log n + s)$ where s is the total number of intersecting rectangles [Ede83]. In 3D, algorithms of complexity $O(n \log^2 n + s)$ complexity are known, where s is the number of overlapping pairwise bounding boxes [HSS83, HD82]. Algorithms for N -body proximity queries in dynamic environments are based on the *sweep and prune* approach [CLMP95]. This incrementally computes the AABBs for each object and checks them for overlap by computing the projection of the bounding boxes along each dimension, and sorting the interval endpoints using insertion sort or bubble sort [MD76, Bar92, CLMP95]. In environments where the objects make relatively small movements between successive frames, the lists can be sorted in expected linear time, leading to expected-time $O(n + m)$, where m is the number of overlapping intervals along any dimension. These algorithms are limited to environments where objects undergo rigid motion. Govindaraju et al. [GRLM03] have presented a general algorithm for large environments composed of rigid as well as nonrigid motion. This algorithm uses graphics hardware to prune the number of objects that are in close proximity and eventually checks for overlapping triangles between the objects. In practice, it works well in large environments composed of nonrigid and breakable objects. However, its accuracy is governed by the resolution of the rasterization hardware.

OUT-OF-CORE ALGORITHMS

In many applications, it may not be possible to load a massive geometric model composed of millions of primitives in the main memory for interactive proximity queries. In addition, algorithms based on spatial partitioning or bounding volume hierarchies also add additional memory overhead. Thus, it is important to develop proximity-query algorithms that use a relatively small or bounded memory footprint.

Wilson et al. [WLML99] presented an out-of-core algorithm to perform collision and separation-distance queries on large environments. It uses *overlap graphs* to exploit locality of computation. For a large model, the algorithm automatically encodes the proximity information between objects and represents it using an overlap graph. The overlap graph is computed off-line and preprocessed using graph partitioning, object decomposition, and refinement algorithms. At run time it traverses localized subgraphs and orders the computations to check the corresponding geometry for proximity tests, as well as pre-fetch geometry and associated hierarchical data structures. To perform interactive proximity queries in dynamic environments, the algorithm uses the BVHs, modifies the localized subgraph(s) on the fly, and takes advantage of spatial and temporal coherence.

35.7 PROXIMITY QUERY PACKAGES

Many systems and libraries have been developed for performing different proximity queries. These include:

I-COLLIDE: I-COLLIDE is an interactive and exact collision-detection system for environments composed of convex polyhedra or union of convex pieces. The system is based on the LC incremental distance computation algorithm [LC91] and an algorithm to check for collision between multiple moving objects [CLMP95]. It takes advantage of temporal coherence. http://gamma.cs.unc.edu/I_COLLIDE.

RAPID: RAPID is a robust and accurate interference detection library for a pair of unstructured polygonal models. It is applicable to polygon soups—models which contain no adjacency information and obey no topological constraints. It is based on OBBTrees and uses a fast overlap test based on Separating Axis Theorem to check whether two OBBs overlap [GLM96]. <http://gamma.cs.unc.edu/OBB/OBBT.html>

V-COLLIDE: V-COLLIDE is a collision detection library for large dynamic environments [HLC⁺97], and unites the N -body processing algorithm of I-COLLIDE with the pair processing algorithm of RAPID. Consequently, it is designed to operate on large numbers of static or moving polygonal objects, and the models may be unstructured. http://gamma.cs.unc.edu/V_COLLIDE

Enhanced GJK Algorithm: It is a library for distance computation based on the enhanced GJK algorithm [GJK88] developed by Cameron [Cam97]. It takes advantage of temporal coherence between successive frames. <http://www.comlab.ox.ac.uk/oucl/users/stephen.cameron/distances.html>

SOLID: SOLID is a library for interference detection of multiple 3D polygonal objects undergoing rigid motion. The shapes used by SOLID are polygon soups. The library exploits frame coherence by maintaining a set of pairs of prox-

mate objects using incremental sweep and pruning on hierarchies of axis-aligned bounding boxes. Though slower for close proximity scenarios, its performance is comparable to that of V-COLLIDE in other cases. <http://www.win.tue.nl/cs/tt/gino/solid>

PQP: PQP, a Proximity Query Package, supports collision detection, separation-distance computation or tolerance verification. It uses OBBTree for collision queries and a hierarchy of swept-sphere volumes to perform distance queries [LGLM99]. It assumes that each object is a collection of triangles and can handle polygon soup models. <http://gamma.cs.unc.edu/SSV>

SWIFT: SWIFT a library for collision detection, distance computation, and contact determination between 3D polygonal objects undergoing rigid motion. It assumes that the input primitives are convex polytopes or a union of convex pieces. The underlying algorithm is based on a variation of LC [EL00]. The resulting system is faster, more robust, and more memory efficient than I-COLLIDE. <http://gamma.cs.unc.edu/SWIFT>

SWIFT++: SWIFT++ a library for collision detection, approximate and exact distance computation, and contact determination between closed and bounded polyhedral models. It decomposes the boundary of each polyhedra into convex patches and precomputes a hierarchy of convex polytopes [EL01]. It uses the SWIFT library to perform the underlying computations between the bounding volumes. <http://gamma.cs.unc.edu/SWIFT++>

QuickCD: QuickCD is a general-purpose collision detection library, capable of performing exact collision detection on complex models. The input model is a collection of triangles, with assumptions on the structure or topologies of the model. It precomputes a hierarchy of k -DOPs for each object and uses them to perform fast collision queries [KHM⁺98]. <http://www.ams.sunysb.edu/~jklosow/quickcd/QuickCD.html>

OPCODE: OPCODE is a collision detection library between general polygonal models. It uses a hierarchy of AABBs. It is memory efficient in comparison to RAPID, SOLID, or QuickCD. <http://www.codercorner.com/Opcode.htm>

DEEP: DEEP estimates the penetration depth and the associated penetration direction between two overlapping convex polytopes. It uses an incremental algorithm the computes a “locally optimal solution” by walking on the surface of the Minkowski sum of two polytopes [KLM02]. <http://gamma.cs.unc.edu/DEEP>

PIVOT: PIVOT computes generalized proximity information between arbitrary objects using graphics hardware. It uses multipass rendering techniques and accelerated distance computation, and provides an approximate solution for different proximity queries. These include collision detection, distance computation, local penetration depth, contact region and normals, etc. [HZLM01, HZLM02]. It involves no preprocessing and can handle deformable models. <http://gamma.cs.unc.edu/PIVOT>

RELATED CHAPTERS

- [Chapter 23. Voronoi diagrams and Delaunay triangulations](#)
- [Chapter 34. Point location](#)

- [Chapter 38. Geometric intersection](#)
 - [Chapter 47. Algorithmic motion planning](#)
 - [Chapter 50: Modeling motion](#)
 - [Chapter 54. Surface simplification and 3D geometry compression](#)
 - [Chapter 64. Software](#)
-

REFERENCES

- [AB88] S.S. Abhyankar and C.L. Bajaj. Computations with algebraic curves. In *Lecture Notes Comput. Sci.*, volume 358, pages 279–284. Springer-Verlag, Berlin, 1988.
- [AGHP⁺00] P.K. Agarwal, L.J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir. Penetration depth of two convex polytopes in 3d. *Nordic J. Computing*, 7:227–240, 2000.
- [Bar92] D. Baraff. *Dynamic simulation of non-penetrating rigid body simulation*. Ph.D. thesis, Cornell Univ., Ithaca, 1992.
- [BCG⁺96] G. Barequet, B. Chazelle, L.J. Guibas, J.S.B. Mitchell, and A. Tal. Boxtree: A hierarchical representation of surfaces in 3D. In *Proc. Eurographics '96*, 1996.
- [BEG⁺99] J. Basch, J. Erickson, L.J. Guibas, J. Hershberger, and L. Zhang. Kinetic collision detection between two simple polygons. In *Proc. 10th Sympos. Discrete Algorithms*, pages 102–111, 1999.
- [Ber01] G. Bergen. Proximity queries and penetration depth computation on 3d game objects. *Game Developers Conf.*, 2001.
- [BFJP87] R. Barnhill, G. Farin, M. Jordan, and B. Piper. Surface/surface intersection. *Comput. Aided Geom. Design*, 4:3–16, 1987.
- [BKSS90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The r*-tree: An efficient and robust access method for points and rectangles. *Proc. SIGMOD Conf. Management Data*, pages 322–331, 1990.
- [Boy79] J.W. Boyse. Interference detection among solids and surfaces. *Commun. ACM*, 22:3–9, 1979.
- [Cam90] S. Cameron. Collision detection by four-dimensional intersection testing. *Proc. Internat. Conf. Robot. Autom.*, pages 291–302, 1990.
- [Cam97] S. Cameron. Enhancing GJK: Computing minimum and penetration distance between convex polyhedra. *Proc. Internat. Conf. Robot. Autom.*, pages 3112–3117, 1997.
- [Can86] J.F. Canny. Collision detection for moving polyhedra. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:200–209, 1986.
- [CC86] S. Cameron and R.K. Culley. Determining the minimum translational distance between two convex polyhedra. *Proc. Internat. Conf. Robot. Autom.*, pages 591–596, 1986.
- [CLMP95] J. Cohen, M.C. Lin, D. Manocha, and M. Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *Proc. ACM Interactive 3D Graphics Conf.*, pages 189–196, 1995.
- [CW96] K. Chung and W. Wang. Quick collision detection of polytopes in virtual environments. In *Proc. ACM Sympos. Virtual Reality Soft. Tech.*, 1996.

- [DHKS93] D.P. Dobkin, J. Hershberger, D.G. Kirkpatrick, and S. Suri. Computing the intersection-depth of polyhedra. *Algorithmica*, 9:518–533, 1993.
- [DK90] D.P. Dobkin and D.G. Kirkpatrick. Determining the separation of preprocessed polyhedra—A unified approach. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes Comput. Sci.*, pages 400–413. Springer-Verlag, Berlin, 1990.
- [DZ93] P. Dworkin and D. Zeltzer. A new model for efficient dynamics simulation. *Proc. EG Workshop Comput. Animat. Simul.*, pages 175–184, 1993.
- [Ede83] H. Edelsbrunner. A new approach to rectangle intersections, Part I. *Internat. J. Comput. Math.*, 13:209–219, 1983.
- [EGSZ99] J. Erickson, L.J. Guibas, J. Stolfi, and L. Zhang. Separation sensitive collision detection for convex objects. *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 327–336, 1999.
- [EL00] S. Ehmann and M.C. Lin. Accelerated proximity queries between convex polyhedra using multi-level Voronoi marching. *Proc. IEEE/RSJ Internat. Conf. Intell. Robots Sys.*, pages 2101–2106, 2000.
- [EL01] S. Ehmann and M.C. Lin. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Comput. Graph. Forum*, 20(3), pages 500–510, 2001.
- [FL01] S. Fisher and M.C. Lin. Deformed distance fields for simulation of non-penetrating flexible bodies. *Proc. EG Workshop Comput. Animat. Simul.*, pages 99–111, 2001.
- [GHZ99] L.J. Guibas, D. Hsu, and L. Zhang. *H-Walk*: Hierarchical distance computation for moving convex bodies. *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 265–273, 1999.
- [GJK88] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between objects in three-dimensional space. *IEEE J. Robot. Autom.*, RA-4:193–203, 1988.
- [GLM96] S. Gottschalk, M.C. Lin, and D. Manocha. OBB-Tree: A hierarchical structure for rapid interference detection. In *Proc. ACM Conf. SIGGRAPH 96*, pages 171–180, 1996.
- [GO94] E.G. Gilbert and C.J. Ong. New distances for the separation and penetration of objects. In *Proc. Internat. Conf. Robot. Autom.*, pages 579–586, 1994.
- [Got00] S. Gottschalk. *Collision Queries using Oriented Bounding Boxes*. Ph.D. thesis, Univ. North Carolina, Chapel Hill, Dept. Computer Science, 2000.
- [GRLM03] N. Govindraju, S. Redon, M.C. Lin and D. Manocha. CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware. In *Proc. ACM SIGGRAPH/Eurographics Workshop Graphics Hardware*, pages 25–32, 2003.
- [HBZ90] B.V. Herzen, A.H. Barr, and H.R. Zatz. Geometric collisions for time-dependent parametric surfaces. *Comput. Graph.*, 24:39–48, 1990.
- [HD82] H. Six and D. Wood. Counting and reporting intersections of d -ranges. *IEEE Trans. Comput.*, C-31:181–187, 1982.
- [HKL⁺98] D. Hsu, L.E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. *Proc. 3rd Workshop Algorithmic Found. Robot.*, pages 141–154, 1998.

- [HKM95] M. Held, J.T. Klosowski, and J.S.B. Mitchell. Evaluation of collision detection methods for virtual reality fly-throughs. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 205–210, 1995.
- [HKM96] M. Held, J.T. Klosowski, and J.S.B. Mitchell. Real-time collision detection for motion simulation within complex environments. In *ACM SIGGRAPH 96 Visual Proc.*, page 151, 1996.
- [HLC⁺97] T. Hudson, M.C. Lin, J. Cohen, S. Gottschalk, and D. Manocha. V-collide: Accelerated collision detection for vrml. In *Proc. VRML Conf.*, pages 119–125, 1997.
- [Hof89] C. Hoffmann. *Geometric and Solid Modeling*. Morgan Kaufmann, San Mateo, 1989.
- [HSS83] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. Efficient detection of intersections among spheres. *Internat. J. Robot. Res.*, 2:77–80, 1983.
- [Hub95] P.M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graphics*, 15:179–210, 1995.
- [HZLM01] K.E. Hoff III, A. Zaferakis, M.C. Lin, and D. Manocha. Fast and simple geometric proximity queries using graphics hardware. *Proc. ACM Sympos. Interactive 3D Graphics*, pages 145–148, 2001.
- [HZLM02] K.E. Hoff III, A. Zaferakis, M.C. Lin, and D. Manocha. Fast 3D geometric proximity queries between rigid and deformable models using graphics hardware acceleration. Tech. Rep. TR02-004, Dept. of Comput. Sci., Univ. North Carolina, Chapel Hill, 2002.
- [KCMh99] J. Keyser, T. Culver, D. Manocha, and S. Krishnan. MAPC: A library for efficient and exact manipulation of algebraic points and curves. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1999.
- [KGL⁺98] S. Krishnan, M. Gopi, M.C. Lin, D. Manocha, and A. Pattekar. Rapid and accurate contact determination between spline models using shelltrees. *Comput. Graph. Forum*, 17:C315–C326, 1998.
- [KHM⁺98] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k -DOPs. *IEEE Trans. Visualization Comput. Graph.*, 4:21–37, 1998.
- [KLM02] Y. Kim, M.C. Lin, and D. Manocha. Deep: An incremental algorithm for penetration depth computation between convex polytopes. *Proc. IEEE Conf. Robot. Autom.*, pages 921–926, 2002.
- [KM97] S. Krishnan and D. Manocha. An efficient surface intersection algorithm based on the lower dimensional formulation. *ACM Trans. Graph.*, 16:74–106, 1997.
- [KOLM02] Y. Kim, M. Otaduy, M.C. Lin, and D. Manocha. 6-DOF haptic display using localized contact computations. *Proc. Haptics Sympos.*, pages 209–216, 2002.
- [KPLM98] S. Krishnan, A. Pattekar, M.C. Lin, and D. Manocha. Spherical shell: A higher order bounding volume for fast proximity queries. In *Proc. 3rd Internat. Workshop Algorithmic Found. Robot.*, pages 122–136, 1998.
- [KPP90] G.A. Kriegis, P.V. Prakash, and N.M. Patrikalakis. Method for intersecting algebraic surfaces with rational polynomial patches. *Comput. Aided Design*, 22:645–654, 1990.
- [KPW90] G.A. Kriegis, N.M. Patrikalakis, and F.E. Wolter. Topological and differential equation methods for surface intersections. *Comput. Aided Design*, 24:41–55, 1990.
- [KSS02] D.G. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. *Internat. J. Comput. Geom. Appl.*, 12:3–27, 2002.

- [LSHL02] I. Lotan, F. Schwarzer, D. Halperin and J.-C. Latombe. Efficient maintenance and self-collision testing for kinematic chains. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 43–52, 2002.
- [LC91] M.C. Lin and J.F. Canny. Efficient algorithms for incremental distance computation. In *IEEE Conf. Robot. Autom.*, pages 1008–1014, 1991.
- [LGLM99] E. Larsen, S. Gottschalk, M.C. Lin, and D. Manocha. Fast proximity queries with swept sphere volumes. Tech. Rep. TR99-018, Dept. of Comput. Sci., Univ. North Carolina, Chapel Hill, 1999.
- [Lin93] M.C. Lin. *Efficient Collision Detection for Animation and Robotics*. Ph.D. thesis, Dept. Elec. Eng. Comput. Sci., Univ. California, Berkeley, 1993.
- [LM97] M.C. Lin and D. Manocha. Efficient contact determination between geometric models. *Internat. J. Comput. Geom. Appl.*, 7:123–151, 1997.
- [LR80] J.M. Lane and R.F. Riesenfeld. A theoretical development for the computer generation and display of piecewise polynomial surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2:150–159, 1980.
- [Man92] D. Manocha. *Algebraic and Numeric Techniques for Modeling and Robotics*. Ph.D. thesis, Dept. Elec. Eng. Comput. Sci., Univ. California, Berkeley, 1992.
- [MC91] D. Manocha and J.F. Canny. A new approach for surface intersection. *Internat. Comput. Geom. Appl.*, 1:491–516, 1991. Special issue on Solid Modeling.
- [MC92] D. Manocha and J.F. Canny. Algorithms for implicitizing rational parametric surfaces. *Comput. Aided Geom. Design*, 9:25–50, 1992.
- [MD76] M.I. Shamos and D. Hoey. Geometric intersection problems. *Proc. 17th Annu. IEEE Symp. Found. Comput. Sci.*, pages 208–215, 1976.
- [Mir98] B. Mirtich. V-Clip: Fast and robust polyhedral collision detection. *ACM Trans. Graph.*, 17:177–208, 1998.
- [Moo79] R.E. Moore. *Methods and Applications of Interval Analysis*. SIAM Studies in Applied Mathematics 2. SIAM, Philadelphia, 1979.
- [MZ90] M. McKenna and D. Zeltzer. Dynamic simulation of autonomous legged locomotion. *Proc. ACM Conf. SIGGRAPH 90*, pages 29–38, 1990.
- [NAT90] B. Naylor, J. Amanatides, and W. Thibault. Merging bsp trees yield polyhedral modeling results. In *Proc. ACM Conf. SIGGRAPH 90*, pages 115–124, 1990.
- [OL03] M. Otaduy and M.C. Lin. CLODs: Dual hierarchies for multiresolution collision detection. In *Proc. Eurographics Sympos. Geom. Processing*, pages 94–101, 2003.
- [Ove92] M.H. Overmars. Point location in fat subdivisions. *Inform. Proc. Lett.*, 44:261–265, 1992.
- [PML97] M. Ponamgi, D. Manocha, and M.C. Lin. Incremental algorithms for collision detection between solid models. *IEEE Trans. Visualization Comput. Graph.*, 3:51–67, 1997.
- [Pra86] M. Pratt. Surface/surface intersection problems. In J.A. Gregory, editor, *The Mathematics of Surfaces II*, pages 117–142, Clarendon Press, Oxford, 1986.
- [Qui94] S. Quinlan. Efficient distance computation between non-convex objects. In *Proc. Internat. Conf. Robot. Autom.*, pages 3324–3329, 1994.
- [RKC00] S. Redon, A. Kheddar, and S. Coquillart. An algebraic solution to the problem of collision detection for rigid polyhedral objects. *Proc. IEEE Conf. Robot. Autom.*, 2000.

- [SAG84] T.W. Sederberg, D.C. Anderson, and R.N. Goldman. Implicit representation of parametric curves and surfaces. *Comput. Vision Graph. Image Process.*, 28:72–84, 1984.
- [Sam89] H. Samet. *Spatial Data Structures: Quadtree, Octrees and Other Hierarchical Methods*. Addison-Wesley, 1989.
- [Sar83] R.F. Sarraga. Algebraic methods for intersection. *Comput. Vision Graph. Image Process.*, 22:222–238, 1983.
- [Sea93] J. Snyder, A.R. Woodbury, K. Fleischer, B. Currin, A.H. Barr. Interval methods for multi-point collisions between time dependent curved surfaces. In *Proc. ACM Conf. SIGGRAPH 93*, pages 321–334, 1993.
- [Sei90] R. Seidel. Linear programming and convex hulls made easy. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 211–215, Berkeley, California, 1990.
- [ST96] D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Internat. J. Numer. Methods Eng.*, 39:2673–2691, 1996.
- [WG91] W. Bouma and G. Vaněček. Collision detection and analysis in a physically based simulation. *Proc. EG Workshop Comput. Animat. Simul.*, pages 191–203, 1991.
- [WLML99] A. Wilson, E. Larsen, D. Manocha, and M.C. Lin. Partitioning and handling massive models for interactive collision detection. *Comput. Graph. Forum*, 18:319–329, 1999.

36 RANGE SEARCHING

Pankaj K. Agarwal

INTRODUCTION

Range searching is one of the central problems in computational geometry, because it arises in many applications and a variety of geometric problems can be formulated as range-searching problems. A typical range-searching problem has the following form. Let S be a set of n points in \mathbb{R}^d , and let \mathcal{R} be a family of subsets of \mathbb{R}^d ; elements of \mathcal{R} are called **ranges**. We wish to preprocess S into a data structure, so that for a query range γ , the points in $S \cap \gamma$ can be reported or counted efficiently. Typical examples of ranges include rectangles, halfspaces, simplices, and balls. If we are only interested in answering a single query, it can be done in linear time, using linear space, by simply checking each point of S whether it lies in the query range. However, most of the applications call for querying the same set S several times (perhaps with periodic insertions and deletions), in which case we would like to answer a query faster by preprocessing S into a data structure.

Range counting and range reporting are just two instances of range-searching queries. Other examples include *emptiness queries*, in which one wants to determine whether $S \cap \gamma = \emptyset$, and *optimization queries*, in which one wants to choose a point with certain property (e.g., a point in γ with the largest x_1 -coordinate). In order to encompass all different types of range-searching queries, a general range-searching problem can be defined as follows.

Let $(\mathbf{S}, +)$ be a commutative semigroup. For each point $p \in S$, we assign a weight $w(p) \in \mathbf{S}$. For any subset $S' \subseteq S$, let $w(S') = \sum_{p \in S'} w(p)$, where addition is taken over the semigroup. For a query range $\gamma \in \mathcal{R}$, we wish to compute $w(S \cap \gamma)$. For example, counting queries can be answered by choosing the semigroup to be $(\mathbb{Z}, +)$, where $+$ denotes standard integer addition, and setting $w(p) = 1$ for every $p \in S$; emptiness queries by choosing the semigroup to be $(\{0, 1\}, \vee)$ and setting $w(p) = 1$; reporting queries by choosing the semigroup to be $(2^S, \cup)$ and setting $w(p) = \{p\}$; and optimization queries by choosing the semigroup to be (\mathbb{R}, \max) and choosing $w(p)$ to be, for example, the x_1 -coordinate of p .

We can, in fact, define a more general (decomposable) *geometric searching* problem. Let S be a set of *objects* in \mathbb{R}^d (e.g., points, hyperplanes, balls, or simplices), $(\mathbf{S}, +)$ a commutative semigroup, $w : S \rightarrow \mathbf{S}$ a weight function, \mathcal{R} a set of ranges, and $\diamond \subseteq S \times \mathcal{R}$ a “spatial” relation between objects and ranges. Then for a range $\gamma \in \mathcal{R}$, we want to compute $\sum_{p \diamond \gamma} w(p)$. Range searching is a special case of this general searching problem in which S is a set of points in \mathbb{R}^d and $\diamond = \in$. Another widely studied searching problem is *intersection searching*, where $p \diamond \gamma$ if p intersects γ . As we will see below, range-searching data structures are useful for many other geometric searching problems.

The performance of a data structure is measured by the time spent in answering a query, called the *query time*, by the *size* of the data structure, and by the time constructed in the data structure, called the *preprocessing time*. Since the data structure is constructed only once, its query time and size are generally more

important than its preprocessing time. If a data structure supports insertion and deletion operations, its *update time* is also relevant. We should remark that the query time of a range-reporting query on any reasonable machine depends on the output size, so the query time for a range-reporting query consists of two parts — *search time*, which depends only on n and d , and *reporting time*, which depends on n , d , and the output size. Throughout this chapter we will use k to denote the output size.

We assume that d is a small fixed constant, and that big- O and big-Omega notation hide constants depending on d . The dependence on d of the performance of almost all the data structures mentioned in this survey is exponential, which makes them unsuitable in practice for large values of d .

The size of any range-searching data structure is at least linear, since it has to store each point (or its weight) at least once, and the query time in any reasonable model of computation such as pointer machines, RAMs, or algebraic decision trees is $\Omega(\log n)$ even when $d = 1$. Therefore, we would like to develop a linear-size data structure with logarithmic query time. Although near-linear-size data structures are known for orthogonal range searching in any fixed dimension that can answer a query in polylogarithmic time, no similar bounds are known for range searching with more complex ranges such as simplices or disks. In such cases, we seek a tradeoff between the query time and the size of the data structure — How fast can a query be answered using $O(n \text{polylog}(n))$ space, how much space is required to answer a query in $O(\text{polylog}(n))$ time, and what kind of tradeoff between the size and the query time can be achieved?

This chapter is organized as follows. In Section 36.1 we describe various models of computation that are used for range searching. In Section 36.2 we review the orthogonal range-searching data structures, and in Section 36.3 we review simplex range-searching data structures. Section 36.4 surveys other variants and extensions of range searching, including multilevel data structures and kinetic range searching. In Section 36.5, we study intersection-searching problems, which can be regarded as a generalization of range searching. Finally, Section 36.6 explores several optimization queries.

36.1 MODELS OF COMPUTATION

Most geometric algorithms and data structures are implicitly described in the familiar *random access machine* (RAM) model, or the ***real RAM*** model. In the traditional RAM model, memory cells can contain arbitrary $(\log n)$ -bit integers, which can be added, multiplied, subtracted, divided (computing $\lfloor x/y \rfloor$), compared, and used as pointers to other memory cells in constant time. In a real RAM, we also allow memory cells to store arbitrary real numbers (such as coordinates of points). We allow constant-time arithmetic on and comparisons between real numbers, but we do not allow conversion between integers and reals. In the case of range searching over a semigroup other than the integers, we also allow memory cells to contain arbitrary values from the semigroup, but only the semigroup-addition operations can be performed on them.

Many range-searching data structures are described in the more restrictive ***pointer-machine*** model. The main difference between RAM and pointer-machine models is that on a pointer machine, a memory cell can be accessed only through

a series of pointers, while in the RAM model, any memory cell can be accessed in constant time. In the basic pointer-machine model, a data structure is a directed graph with outdegree 2; each node is associated with a label, which is an integer between 0 and n . Nonzero labels are indices of the points in S , and the nodes with label 0 store auxiliary information. The query algorithm traverses a portion of the graph and for each point in the query range it identifies at least one node that stores the index of that point. Chazelle [Cha88b] defines several generalizations of the pointer-machine model that are more appropriate for answering counting and semigroup queries. In Chazelle's generalized pointer-machine models, nodes are labeled with arbitrary $O(\log n)$ -bit integers. In addition to traversing edges in the graph, the query algorithm is also allowed to perform various arithmetic operations on these integers. An *elementary* pointer machine (called EPM) can perform addition and comparisons between integers; an *arithmetic* pointer machine (called APM) can perform subtraction, multiplication, integer division, and shifting ($x \mapsto 2^x$).

If the input is too large to fit into main memory, then the data structure must be stored in secondary memory—on disk, for example—and portions of it must be moved into main memory when needed to answer a query. In this case the bottleneck in query and preprocessing time is the time spent in transferring data between main and secondary memory. A commonly used model is the standard *two-level* memory model, in which one assumes that data is stored in secondary memory in blocks of size B , where B is a parameter. Each access to secondary memory transfers one block (i.e., B words), and we count this as one input/output (I/O) operation. The size of a data structure is the number of blocks required to store it in secondary memory, and the query (resp. preprocessing) time is defined as the number of I/O operations required to answer a query (resp. to construct the structure). Under this model, the size of any data structure is at least n/B , and the range-reporting query time is at least $\log_B n + k/B$. There have been various extensions of this model, including the so-called *cache-oblivious* model in which one does not know the value of B and the goal is to minimize I/O as well as the total work performed.

Most lower bounds, and a few upper bounds, are described in the so-called **semigroup arithmetic model**, which was originally introduced by Fredman [Fre81a] and refined by Yao [Yao85]. In this model, a data structure can be regarded informally as a set of precomputed partial sums in the underlying semigroup. The size of the data structure is the number of sums stored, and the query time is the minimum number of semigroup operations required (on the precomputed sums) to compute the answer to a query. The query time ignores the cost of various auxiliary operations, including the cost of determining which of the precomputed sums should be added to answer a query. Unlike the pointer-machine model, the semigroup model allows immediate access, at no cost, to any precomputed sum.

The informal model we have just described is much too powerful. For example, in this semigroup model, the optimal data structure for range-counting queries consists of the $n + 1$ integers $0, 1, \dots, n$. To answer a counting query, we simply return the correct answer; since no additions are required, we can answer queries in zero “time,” using a “data structure” of only linear size! We need the notion of a **faithful** semigroup to circumvent this problem. A commutative semigroup $(\mathbf{S}, +)$ is faithful if for each $n > 0$, for any sets of indices $I, J \subseteq \{1, \dots, n\}$ where $I \neq J$, and for every sequence of positive integers α_i, β_j ($i \in I, j \in J$), there are semigroup

values $s_1, s_2, \dots, s_n \in \mathbf{S}$ such that $\sum_{i \in I} \alpha_i s_i \neq \sum_{j \in J} \beta_j s_j$. For example, $(\mathbb{Z}, +)$, (\mathbb{R}, \min) , (\mathbb{N}, \gcd) , and $(\{0, 1\}, \vee)$ are faithful, but $(\{0, 1\}, + \bmod 2)$ is not faithful.

Let $S = \{p_1, p_2, \dots, p_n\}$ be a set of objects, \mathbf{S} a faithful semigroup, \mathcal{R} a set of ranges, and \diamond a relation between objects and ranges. (Recall that in the standard range-searching problem, the objects in S are points, and \diamond is containment.) Let x_1, x_2, \dots, x_n be a set of n variables over \mathbf{S} , each corresponding to a point in S . A *generator* $g(x_1, \dots, x_n)$ is a linear form $\sum_{i=1}^n \alpha_i x_i$, where α_i 's are nonnegative integers, not all zero. (In practice, the coefficients α_i are either 0 or 1.) A *storage scheme* for $(S, \mathbf{S}, \mathcal{R}, \diamond)$ is a collection of generators $\{g_1, g_2, \dots, g_s\}$ with the following property: For any query range $\gamma \in \mathcal{R}$, there is a set of indices $I_\gamma \subseteq \{1, 2, \dots, s\}$ and a set of labeled nonnegative integers $\{\beta_i \mid i \in I_\gamma\}$ such that the linear forms

$$\sum_{p_i \diamond \gamma} x_i \quad \text{and} \quad \sum_{i \in I_\gamma} \beta_i g_i$$

are identically equal. In other words, the equation

$$\sum_{p_i \diamond \gamma} w(p_i) = \sum_{i \in I_\gamma} \beta_i g_i(w(p_1), w(p_2), \dots, w(p_n))$$

holds for *any* weight function $w : S \rightarrow \mathbf{S}$. (Again, in practice, $\beta_i = 1$ for all $i \in I_\gamma$.) The size of the smallest such set I_γ is the query time for γ ; the time to actually choose the indices I_γ is ignored. The space used by the storage scheme is measured by the number of generators. There is no notion of preprocessing time in this model.

The semigroup model is formulated slightly differently for off-line range-searching problems. Here we are given a set of weighted points S and a finite set of query ranges \mathcal{R} , and we want to compute the total weight of the points in each query range. This is equivalent to computing the product Aw , where A is the incidence matrix of the points and ranges, and w is the vector of weights. In the off-line semigroup model, introduced by Chazelle [Cha97, Cha01], an algorithm can be described as a circuit with one input for every point and one output for every query range, where every gate performs a binary semigroup addition. The running time of the algorithm is the total number of gates.

A serious weakness of the semigroup model is that it does not allow subtractions even if the weights of points belong to a group. Therefore, we will also consider the *group model*, in which both additions and subtractions are allowed [Cha98].

Almost all geometric range-searching data structures are constructed by subdividing space into several regions with nice properties and recursively constructing a data structure for each region. Range queries are answered with such a data structure by performing a depth-first search through the resulting recursive space partition. The *partition-graph* model, introduced by Erickson [Eri96a, Eri96b], formalizes this divide-and-conquer approach, at least for simplex range searching data structures. The partition graph model can be used to study the complexity of emptiness queries, unlike the semigroup arithmetic and pointer machine models, in which such queries are trivial.

We conclude this section by noting that most of the range-searching data structures discussed in this paper (halfspace range-reporting data structures being a notable exception) are based on the following general scheme. Given a point set S , the structure precomputes a family $\mathcal{F} = \mathcal{F}(S)$ of *canonical subsets* of S and store the weight $w(C) = \sum_{p \in C} w(p)$ of each canonical subset $C \in \mathcal{F}$. For a query range γ , the query procedure determines a partition $\mathcal{C}_\gamma = \mathcal{C}(S, \gamma) \subseteq \mathcal{F}$ of $S \cap \gamma$ and adds

the weights of the subsets in \mathcal{C}_γ to compute $w(S \cap \gamma)$. We will refer to such a data structure as a **decomposition scheme**.

There is a close connection between the decomposition schemes and the storage schemes of the semigroup arithmetic model described earlier. Each canonical subset $C = \{p_i \mid i \in I\} \in \mathcal{F}$, where $I \subseteq \{1, 2, \dots, n\}$, corresponds to the generator $\sum_{i \in I} x_i$. How exactly the weights of canonical subsets are stored and how \mathcal{C}_γ is computed depends on the model of computation and on the specific range-searching problem. In the semigroup (or group) arithmetic model, the query time depends only on the number of canonical subsets in \mathcal{C}_γ , regardless of how they are computed, so the weights of canonical subsets can be stored in an arbitrary manner. In more realistic models of computation, however, some additional structure must be imposed on the decomposition scheme in order to efficiently compute \mathcal{C}_γ . In a *hierarchical* decomposition scheme, the weights are stored in a tree T . Each node v of T is associated with a canonical subset $C_v \in \mathcal{F}$, and the children of v are associated with subsets of C_v . Besides the weight of C_v , some auxiliary information is also stored at v , which is used to determine whether $C_v \in \mathcal{C}_\gamma$ for a query range γ . If the weight of each canonical subset can be stored in $O(1)$ memory cells and if we can determine in $O(1)$ time whether $C_w \in \mathcal{C}_\gamma$ where w is a descendent of a given node v , we call the hierarchical decomposition scheme *efficient*. The total size of an efficient decomposition scheme is simply $O(|\mathcal{F}|)$. For range-reporting queries, in which the “weight” of a canonical subset is the set itself, the size of the data structure is reduced to $O(|\mathcal{F}|)$ by storing the canonical subsets implicitly. Finally, let $r > 1$ be a parameter, and set $\mathcal{F}_i = \{C \in \mathcal{F} \mid r^{i-1} \leq |C| \leq r^i\}$. We call a hierarchical decomposition scheme *r-convergent* if there exist constants $\alpha \geq 1$ and $\beta > 0$ so that the degree of every node in T is $O(r^\alpha)$ and for all $i \geq 1$, $|\mathcal{F}_i| = O((n/r^i)^\alpha)$ and, for all query ranges γ , $|\mathcal{C}_\gamma \cap \mathcal{F}_i| = O((n/r^i)^\beta)$, i.e., the number of canonical subsets in the data structure and in any query output decreases exponentially with their size. We will see below in [Section 36.4](#) that *r*-convergent hierarchical decomposition schemes can be cascaded together to construct multilevel structures that answer complex geometric queries.

To compute $\sum_{p_i \in \gamma} w(p_i)$ for a query range γ using a hierarchical decomposition scheme T , a query procedure performs a depth-first search on T , starting from its root. At each node v , using the auxiliary information stored at v , the procedure determines whether γ contains C_v , whether γ intersects C_v but does not contain C_v , or whether γ is disjoint from C_v . If γ contains C_v , then C_v is added to \mathcal{C}_γ (rather, the weight of C_v is added to a running counter). Otherwise, if γ intersects C_v , the query procedure identifies a subset of children of v , say $\{w_1, \dots, w_a\}$, so that the canonical subsets $C_{w_i} \cap \gamma$, for $1 \leq i \leq a$, form a partition of $C_v \cap \gamma$. Then the procedure searches each w_i recursively. The total query time is $O(\log n + |\mathcal{C}_\gamma|)$, provided constant time is spent at each node visited.

36.2 ORTHOGONAL RANGE SEARCHING

In d -dimensional orthogonal range searching, the ranges are d -rectangles, each of the form $\prod_{i=1}^d [a_i, b_i]$ where $a_i, b_i \in \mathbb{R}$. This is an abstraction of **multikey searching**. For example, the points of S may correspond to employees of a company, each coordinate corresponding to a key such as age, salary, experience, etc. Queries of the form, e.g., “report all employees between the ages of 30 and 40 who earn more

than \$30,000 and who have worked for more than 5 years,” can be formulated as orthogonal range-reporting queries. Because of its numerous applications, orthogonal range searching has been studied extensively. In this section we review recent data structures and lower bounds.

UPPER BOUNDS

Most orthogonal range-searching data structures are based on *range trees*, introduced by Bentley [Ben80]. For a set S of n points in \mathbb{R}^2 , the range tree T of S is a minimum-height binary tree with n leaves whose i th leftmost leaf stores the point of S with the i th smallest x -coordinate. Each interior node v of T is associated with a canonical subset $C_v \subseteq S$ containing the points stored at leaves in the subtree rooted at v . Let a_v (resp. b_v) be the smallest (resp. largest) x -coordinate of any point in C_v . The interior node v stores the values a_v and b_v and the set C_v in an array sorted by the y -coordinates of its points. The size of T is $O(n \log n)$, and it can be constructed in time $O(n \log n)$. The range-reporting query for a rectangle $q = [a_1, b_1] \times [a_2, b_2]$ can be answered by traversing T as follows. Suppose we are at a node v . If v is a leaf, then we report the point stored at v if it lies inside q . If v is an interior node and the interval $[a_v, b_v]$ does not intersect $[a_1, b_1]$, there is nothing to do. If $[a_v, b_v] \subseteq [a_1, b_1]$, we report all the points of C_v whose y -coordinates lie in the interval $[a_2, b_2]$, by performing a binary search. Otherwise, we recursively visit both children of v . The query time of this procedure is $O(\log^2 n + k)$, which can be improved to $O(\log n + k)$, using *fractional-cascading* (Section 34.3).

The size of the data structure can be reduced to $O(n \log n / \log \log n)$, without affecting the asymptotic query time, by constructing a range tree with $O(\log n)$ fanout and storing additional auxiliary structures at each node [Cha86]. If the query rectangles are “3-sided rectangles” of the form $[a_1, b_1] \times [a_2, \infty]$, then one can use a *priority search tree* of size $O(n)$ to answer a planar range-reporting query in time $O(\log n + k)$ [McC85]; see [AE99] for a few other special cases in which the storage can be reduced to linear. All these structures can be implemented in the elementary pointer-machine model and can be dynamized using the standard partial-rebuilding technique [Ove83]. If the preprocessing time of the data structure is $P(n)$, then a point can be inserted into or deleted from the data structure in $O((P(n)/n) \log n)$ amortized time. The update time can be made worst-case using the known deamortization techniques [DR91]. If we have a data structure for answering d -dimensional range-reporting queries, one can construct a $(d+1)$ -dimensional range-reporting structure in the EPM model, using multilevel range trees (see Section 36.4), by paying a $\log n$ factor in storage, preprocessing time, and query time.

If we use the RAM model, a set S of n points in \mathbb{R}^2 can be preprocessed into a data structure of size $O(n \log^\epsilon n)$ so that all k points lying inside a query rectangle can be reported in $O(\log n + k)$ time. Mortensen [Mor03] has developed a data structure of size $O(n \log n / \log \log n)$ that can answer a range query in $O(\log n + k)$ time and can insert or delete a point in $O(\log n)$ time. If the points lie on a $n \times n$ grid in the plane, then a query can be answered in $O(\log \log n + k)$ time using $O(n \log^\epsilon n)$ storage or in time $O((\log \log n)^2 + k \log \log n)$ using $O(n \log \log n)$ storage. For points in \mathbb{R}^3 , a query can be answered in $O(\log n + k)$ time using $O(n \log^{1+\epsilon} n)$ storage. As for the data structures in the pointer-machine model, the range reporting data structures in the RAM model can be extended to higher dimensions by paying

a $\log n$ factor in storage and query time for each dimension. Alternatively, a d -dimensional data structure can be extended to a $(d+1)$ -dimensional data structure by paying a $\log^{1+\epsilon} n$ factor in storage and a $\log n / \log \log n$ factor in the query time.

TABLE 36.2.1 Upper bounds known on orthogonal range reporting.

d	MODEL	$S(n)$	$Q(n)$
$d = 2$	RAM	n	$\log n + k \log^\epsilon(2n/k)$
	RAM	$n \log \log n$	$\log n + k \log \log(4n/k)$
	RAM	$n \log^\epsilon n$	$\log n + k$
	APM	n	$k \log(2n/k)$
	EPM	n	$k \log^2(2n/k)$
	EPM	$\frac{n \log n}{\log \log n}$	$\log n + k$
$d = 3$	RAM	$n \log^{1+\epsilon} n$	$\log n + k$
	EPM	$n \log^3 n$	$\log n + k$

The two-dimensional range tree described earlier can be used to answer a range counting query in $O(\log n)$ time using $O(n \log n)$ storage. However, if we use the RAM model in which we assume that each word stores $\log n$ bits, the size can be reduced to $O(n)$ by compressing the auxiliary information stored at each node [Cha88b].

TABLE 36.2.2 Upper bounds known on orthogonal semigroup range searching.

MODEL	$S(n)$	$Q(n)$
arithmetic	m	$\frac{n \log n}{\log(2m/n)}$
RAM	n	$\log^{2+\epsilon} n$
RAM	$n \log \log n$	$\log^2 n \log \log n$
RAM	$n \log^\epsilon n$	$\log^2 n$
APM	n	$\log^3 n$
EPM	n	$\log^4 n$

LOWER BOUNDS

Fredman [Fre80, Fre81a] was the first to prove nontrivial lower bounds on orthogonal range searching, but he considered the framework in which the points could be inserted and deleted dynamically. He showed that a mixed sequence of n insertions, deletions, and queries takes $\Omega(n \log^d n)$ time. These bounds were extended by Willard [Wil89] to a group model, under some restrictions. Chazelle proved lower bounds for the static version of orthogonal range searching, which almost match the best upper bounds known [Cha90b]. The following theorem summarizes his main result.

THEOREM 36.2.1 Chazelle [Cha90b]

Let (\mathbf{S}, \oplus) be a faithful semigroup, let d be a constant, and let n and m be parameters. Then there exists a set S of n weighted points in \mathbb{R}^d , with weights from \mathbf{S} , such that the worst-case query time, under the semigroup model, for an orthogonal range-searching data structure that uses m units of storage is $\Omega((\log n / \log(2m/n))^{d-1})$.

Theorem 36.1.1 holds even if the queries are quadrants instead of rectangles. In fact, this lower bound applies to answering the dominance query for a randomly chosen query point; in this sense the above theorem gives a lower bound on the average-case complexity of the query time. It should be pointed out that Theorem 36.1.1 assumes the weights of points in S to be a part of the input. That is, the data structure is not tailored to a special set of weights, and it should work for any set of weights. It is conceivable that a faster algorithm can be developed for answering orthogonal range-counting queries, exploiting the fact that the weight of each point is 1 in this case. None of the known algorithms are able to exploit this fact, however.

A rather surprising result of Chazelle [Cha90a] shows that the size of any data structure on a pointer machine that answers a d -dimensional range-reporting query in $O(\log^c n + k)$ time, for any constant c , is $\Omega(n(\log n / \log \log n)^{d-1})$. Notice that this lower bound is greater than the known upper bound for answering two-dimensional reporting queries on the RAM model.

These lower bounds do not hold for off-line orthogonal range searching, where given a set of n weighted points in \mathbb{R}^d and a set of n rectangles, one wants to compute the weight of points in each rectangle. Chazelle [Cha97] proved that the off-line version takes $\Omega(n(\log n / \log \log n)^{d-1})$ time in the semigroup model and $\Omega(n \log \log n)$ time in the group model. For $d = \Omega(\log n)$ (resp. $d = \Omega(\log n / \log \log n)$), the lower bound for the off-line range-searching problem in the group model can be improved to $\Omega(n \log n)$ (resp. $\Omega(n \log n / \log \log n)$) [CL01]. The close connection between the lower bounds on range searching and the “discrepancy” of set systems is discussed in [Chapter 44](#).

SECONDARY MEMORY STRUCTURES

I/O-efficient orthogonal range-searching structures have received much attention recently because of massive data sets in spatial databases. The main idea underlying these structures is to construct high-degree trees instead of binary trees. For example, variants of B-trees are used to answer one-dimensional range-searching queries [Sam90]. Arge et al. [ASV99] developed an external priority search tree so that a 3-sided-rectangle-reporting query can be answered in $O(\log_B \nu + \kappa)$ I/Os using $O(\nu)$ storage, where $\nu = n/B$ and $\kappa = k/B$. The main ingredient of their algorithm is a data structure that can store B^2 points using $O(B)$ blocks and can report all points lying inside a 3-sided rectangle in $O(1 + \kappa)$ I/Os. Combining their external priority search tree with Chazelle’s data structure for range reporting [Cha86], they construct an external range tree that uses $O(\nu \log_B \nu / \log \log_B \nu)$ blocks and answers a two-dimensional rectangle reporting query in time $O(\log_B n + \kappa)$. By extending the ideas proposed in [Cha90a], it can be shown that any secondary-memory data structure that answers a range-reporting query using $O(\log_B^c \nu + \kappa)$ I/Os requires $\Omega(\nu \log_B \nu / \log \log_B n)$ storage. Govindrajan et al. [GAA03] have shown that a two-

dimensional range counting query can be answered in $O(\log_B \nu)$ I/Os using $O(\nu)$ blocks of storage, assuming that each word can store $\log n$ bits.

TABLE 36.2.3 Secondary-memory structures for orthogonal range searching. Here $\beta(n) = \log \log \log_B \nu$.

d	RANGE	$Q(n)$	$S(n)$
$d = 1$	interval	$\log_B \nu + \kappa$	ν
$d = 2$	3-sided rect rectangle	$\log_B \nu + \kappa$	ν
		$\log_B \nu + \kappa$	$\nu \log_B \nu / \log \log_B \nu$
$d = 3$	octant box	$\beta(\nu, B) \log_B \nu + \kappa$	$\nu \log \nu$
		$\beta(\nu, B) \log_B \nu + \kappa$	$\nu \log^4 \nu$

LINEAR-SIZE DATA STRUCTURES

None of the data structures described above are used in practice, even in two dimensions, because of the polylogarithmic overhead in their size. For a data structure to be used in real applications, its size should be at most cn , where c is a very small constant, the time to answer a typical query should be small—the lower bounds mentioned earlier imply that we cannot hope for small worst-case bounds—and it should support insertions and deletions of points. Keeping these goals in mind, a plethora of data structures have been proposed.

The most widely used data structures for answering one-dimensional range queries are B-trees and their variants. Since a B-tree requires a linear order on the input elements, several techniques such as lexicographic ordering, bit interleaving, and space-filling curves have been used to define a linear ordering on points in higher dimensions in order to store them in a B-tree. A more efficient approach to answer high-dimensional range queries is to construct a recursive partition of space, typically into rectangles, and to construct a tree induced by this partition. The simplest example of this type of data structure is the **quadtree** in the plane. A quadtree is a 4-way tree, each of whose nodes is associated with a square R_v . R_v is partitioned into four equal-size squares, each of which is associated with one of the children of v . The squares are partitioned until at most one point is left inside a square. A range-search query can be answered by traversing the quadtree in a top-down fashion. Because of their simplicity, quadtrees are one of the most widely used data structures for a variety of problems. One disadvantage of quadtrees is that arbitrarily many levels of partitioning may be required to separate tightly clustered points. Finkel and Bentley [FB74] described a variant of the quad tree for range searching, called a *point quadtree*, in which each node is associated with a rectangle and the rectangle is partitioned into four rectangles by choosing a point in the interior and drawing horizontal and vertical lines through that point. Typically the point is chosen so that the height of the tree is $O(\log n)$. In order to minimize the number of disk accesses, one can partition the square into many squares (instead of four) by drawing either a uniform or a nonuniform grid. The **grid file** data structure, introduced by Nievergelt et al. [NHS84], is based on this idea.

Quadtrees and their variants construct a grid on a rectangle containing all the

input points. One can instead partition the enclosing rectangle into two rectangles by drawing a horizontal or a vertical line and partitioning each of the two rectangles independently. This is the idea behind the ***k-d-tree*** data structure of Bentley [Ben75]. In particular, a *k-d-tree* is a binary tree, each of whose nodes v is associated with a rectangle R_v . If R_v does not contain any point in its interior, v is a leaf. Otherwise, R_v is partitioned into two rectangles by drawing a horizontal or vertical line so that each rectangle contains at most half of the points; splitting lines are alternately horizontal and vertical. In order to minimize the number of disk accesses, Robinson [Rob81] generalized a *k-d-tree* to a ***kd-B-tree***, in which one constructs a B-tree instead of a binary tree on the recursive partition of the enclosing rectangle, so all leaves of the tree are at the same level and each node has between $B/2$ and B children. The rectangles associated with the children are obtained by splitting R_v recursively, as in a *k-d-tree*. A simple top-down approach to construct a *kd-B-tree* requires $O(\nu \log_2 \nu)$ I/Os, but the preprocessing cost can be reduced to $O(\nu \log_B \nu)$ I/Os using a more sophisticated approach [AAPV01].

If points are dynamically inserted into a *k-d-tree* or *kd-B-tree*, then some of the nodes may have to be split, an expensive operation because splitting a node may require reconstructing the entire subtree rooted at that node. A few variants of *k-d-trees* have been proposed that can update the structure in $O(\text{polylog} n)$ time and can answer a query in $O(\sqrt{n} + k)$ time. On the practical side, many variants of *kd-B-trees* have also been proposed to minimize the number of splits, to optimize the space, and to improve the query time, most notably ***buddy trees*** [SRF87] and ***hB-trees*** [LS90, ELS97]. A buddy tree is a combination of quad- and *kd-B-trees* in the sense that rectangles are split into sub-rectangles only at some specific locations, which simplifies the split procedure. If points are in degenerate position, then it may not be possible to split a square into two halves by a line. Lomet and Salzberg [LS90] circumvent this problem by introducing a new data structure, called an ***hB-tree***, in which the region associated with a node is allowed to be $R_1 \setminus R_2$ where R_1 and R_2 are rectangles. A more refined version of this data structure, known as an ***hB^{II}-tree***, is presented in [ELS97].

All the data structures described in this section for d -dimensional range searching construct a recursive partition of \mathbb{R}^d . There are other data structures that construct a hierarchical cover of \mathbb{R}^d , most popular of which is the ***R-tree***, originally introduced by Guttman [Gut84]. An R-tree is a B-tree, each of whose nodes stores a set of rectangles. Each leaf stores a subset of input points, and each input point is stored at exactly one leaf. For each node v , let R_v be the smallest rectangle containing all the rectangles stored at v ; R_v is stored at the parent of v (along with the pointer to v). R_v induces the subspace corresponding to the subtree rooted at v , in the sense that for any query rectangle intersecting R_v , the subtree rooted at v is searched. Rectangles stored at a node are allowed to overlap. Although allowing rectangles to overlap helps reduce the size of the data structure, answering a query becomes more expensive. Guttman suggests a few heuristics to construct an R-tree so that the overlap is minimized. Several better heuristics for improving the performance minimizing the overlap have been proposed, including R*- and Hilbert-R-trees. An R-tree also may be constructed on a set of rectangles. Agarwal et al. [AdBG⁺02] showed how to construct an R-tree on a set of n rectangles in \mathbb{R}^d so that all k rectangles intersecting a query rectangle can be reported in $O(n^{1-1/d} + k)$ time.

PARTIAL-SUM QUERIES

Partial-sum queries require preprocessing a d -dimensional array A with n entries, in an additive semigroup, into a data structure, so that for a d -dimensional rectangle $\gamma = [a_1, b_1] \times \dots \times [a_d, b_d]$, the sum

$$\sigma(A, \gamma) = \sum_{(k_1, k_2, \dots, k_d) \in \gamma} A[k_1, k_2, \dots, k_d]$$

can be computed efficiently. In the off-line version, given A and m rectangles $\gamma_1, \gamma_2, \dots, \gamma_m$, we wish to compute $\sigma(A, \gamma_i)$ for each i . This is just a special case of orthogonal range searching, where the points lie on a regular d -dimensional lattice.

Partial-sum queries are widely used for on-line analytical processing (OLAP) of commercial databases. OLAP allows companies to analyze aggregate databases built from their data warehouses. A popular data model for OLAP applications is the multidimensional database, known as **data cube** [GBLP96], which represents the data as d -dimensional array. Thus, an aggregate query can be formulated as a partial-sum query. Driven by this application, several heuristics have been proposed to answer partial-sum queries on data cubes [HBA97, HAMS97] and the references therein.

Yao [Yao82] showed that, for $d = 1$, a partial-sum query can be answered in $O(\alpha(n))$ time using $O(n)$ space, where $\alpha(n)$ is the inverse Ackermann function. If the additive operator is *max* or *min*, then a partial-sum query can be answered in $O(1)$ time under the RAM model using a Cartesian tree, developed by Vuillemin [Vui80].

For $d > 1$, Chazelle and Rosenberg [CR89] developed a data structure of size $O(n \log^{d-1} n)$ that can answer a partial-sum query in time $O(\alpha(n) \log^{d-2} n)$. They also showed that the off-line version that answers m given partial-sum queries on n points takes $\Omega(n + m\alpha(m, n))$ time for any fixed $d \geq 1$. If points are allowed to insert into S , the query time is $\Omega(\log n / \log \log n)$ [Fre79, Yao85] for the one-dimensional case; the bounds were extended by Chazelle [Cha90b] to $\Omega((\log n / \log \log n)^d)$, for any fixed dimension d .

36.3 SIMPLEX RANGE SEARCHING

Unlike orthogonal range searching, no simplex range-searching data structure is known that can answer a query in polylogarithmic time using near-linear storage. In fact, the lower bounds stated below indicate that there is little hope of obtaining such a data structure, since the query time of a linear-size data structure, under the semigroup model, is roughly at least $n^{1-1/d}$ (thus only saving a factor of $n^{1/d}$ over the naive approach). Because the size and query time of any data structure have to be at least linear and logarithmic, respectively, we consider these two ends of the spectrum: (i) how fast a simplex range query can be answered using a linear-size data structure; and (ii) how large the size of a data structure should be in order to answer a query in logarithmic time. Combining these two extreme cases leads to a space/query-time tradeoff.

GLOSSARY

Arrangements: The arrangement of a set H of hyperplanes in \mathbb{R}^d is the subdivision of \mathbb{R}^d into cells of dimension k , for $0 \leq k \leq d$, each cell of dimension $k < d$ being a maximal connected set contained in the intersection of a fixed subset of H and not intersecting any other hyperplane of H . See [Chapter 24](#).

1/r-cutting: Let H be a set of n hyperplanes in \mathbb{R}^d and let $1 \leq r \leq n$ be a parameter. A $(1/r)$ -cutting of H is a set of (relatively open) disjoint simplices covering \mathbb{R}^d so that each simplex intersects at most n/r hyperplanes of H .

Duality: The dual of a point $(a_1, \dots, a_d) \in \mathbb{R}^d$ is the hyperplane $x_d = -a_1x_1 - \dots - a_{d-1}x_{d-1} + a_d$, and the dual of a hyperplane $x_d = b_1x_1 + \dots + b_d$ is the point $(b_1, \dots, b_{d-1}, b_d)$.

LINEAR-SIZE DATA STRUCTURES

Most of the linear-size data structures for simplex range searching are based on **partition trees**, originally introduced by Willard [Wil82] for a set of points in the plane. Roughly speaking, a partition tree is a hierarchical decomposition scheme (in the sense described in Section 36.1) that recursively partitions the points into canonical subsets and encloses each canonical subset by a simple convex region (e.g. simplex), so that any hyperplane intersects only a fraction of the regions associated with the “children” of a canonical subset. A query is answered as described in Section 36.1. The query time depends on the maximum number of children regions of a node that a hyperplane can intersect. The partition tree proposed by Willard partitions each canonical subsets into four children, each contained in a wedge so that any line intersects at most three of them. As a result, the time spent in reporting all k points lying inside a triangle is $O(n^{0.792} + k)$. A number of partition trees with improved query time were introduced later, but a major breakthrough in simplex range searching was made by Haussler and Welzl [HW87]. They formulated range searching in an abstract setting and, using elegant probabilistic methods, gave a randomized algorithm to construct a linear-size partition tree with $O(n^\alpha)$ query time, where $\alpha = 1 - \frac{1}{d(d-1)+1} + \epsilon$ for any $\epsilon > 0$. The best linear-size data structure known for simplex range searching, which almost matches the lower bounds mentioned below, is by Matoušek [Mat93]. He showed that a simplex range-counting (resp. range-reporting) query in \mathbb{R}^d can be answered in time $O(n^{1-1/d})$ (resp. $O(n^{1-1/d} + k)$). His algorithm is based on a stronger version of the following theorem.

THEOREM 36.3.1 Matoušek [Mat92]

Let S be a set of n points in \mathbb{R}^d , and let $1 < r \leq n/2$ be a given parameter. Then there exists a family of pairs $\Pi = \{(S_1, \Delta_1), \dots, (S_m, \Delta_m)\}$ such that $S_i \subseteq S$ lies inside simplex Δ_i , $n/r \leq |S_i| \leq 2n/r$, $S_i \cap S_j = \emptyset$ for $i \neq j$, and every hyperplane crosses at most $cr^{1-1/d}$ simplices of Π ; here c is a constant. If r is a constant, then Π can be constructed in $O(n)$ time.

Using this theorem, a partition tree T can be constructed as follows. Each interior node v of T is associated with a subset $S_v \subseteq S$ and a simplex Δ_v containing S_v ; the root of T is associated with S and \mathbb{R}^d . Choose r to be a sufficiently large

constant. If $|S| \leq 4r$, T consists of a single node, and it stores all points of S . Otherwise, we construct a family of pairs $\Pi = \{(S_1, \Delta_1), \dots, (S_m, \Delta_m)\}$ using Theorem 36.3.1. We recursively construct a partition tree T_i for each S_i and attach T_i as the i th subtree of u . The root of T_i also stores Δ_i . The total size of the data structure is linear, and it can be constructed in time $O(n \log n)$. Since any hyperplane intersects at most $cr^{1-1/d}$ simplices of Π , the query time of simplex range reporting is $O(n^{1-1/d} \cdot n^{\log_r c} + k)$; the $\log_r c$ factor can be reduced to any arbitrarily small positive constant ϵ by choosing r sufficiently large. Although the query time can be improved to $O(n^{1-1/d} \log^c n + k)$ by choosing r to be n^ϵ , a stronger version of Theorem 36.3.1, which was proved in [Mat93], and some other sophisticated techniques are needed to obtain $O(n^{1-1/d} + k)$ query time.

If the points in S lie on a b -dimensional algebraic surface of constant degree, a simplex range-counting query can be answered in time $O(n^{1-\gamma})$ using linear space, where $\gamma = 1/\lfloor(d+b)/2\rfloor$. Better bounds can be obtained for halfspace range reporting, using *filtering search*; see Table 36.3.1. A halfspace range-reporting query in the I/O model can be answered in $O(\log_B \nu + \kappa)$ I/Os using $O(\nu)$ (resp. $O(\nu \log_B \nu)$) blocks of storage for $d = 2$ (resp. $d = 3$) [AAE⁺00].

TABLE 36.3.1 Near-linear-size data structures for halfspace range searching.

d	$S(n)$	$Q(n)$	NOTES
$d = 2$	n	$\log n + k$	reporting
	n	$\log n$	emptiness
$d = 3$	$n \log \log n$	$\log n + k$	reporting
	n	$\log^2 n + k$	reporting
	n	$\log n$	emptiness
$d > 3$	$n \log \log n$	$n^{1-1/\lfloor d/2 \rfloor} \log^c n + k$	reporting
	n	$n^{1-1/d} 2^{O(\log^* n)}$	emptiness
even d	n	$n^{1-1/\lfloor d/2 \rfloor} \log^c n + k$	reporting

DATA STRUCTURES WITH LOGARITHMIC QUERY TIME

For the sake of simplicity, we first consider the halfspace range-counting problem. Using a standard duality transform, this problem can be reduced to the following: *Given a set H of n hyperplanes, determine the number of hyperplanes of H lying above a query point.* Since the same subset of hyperplanes lies above all points in a single cell of $\mathcal{A}(H)$, the arrangement of H , we can answer a halfspace range-counting query by locating the cell of $\mathcal{A}(H)$ that contains the point dual to the hyperplane bounding the query halfspace. The following theorem of Chazelle [Cha93] yields an $O((n/\log n)^d)$ -size data structure, with $O(\log n)$ query time, for halfspace range counting.

THEOREM 36.3.2 Chazelle [Cha93]

Let H be a set of n hyperplanes and $r \leq n$ a parameter. Set $s = \lceil \log_2 r \rceil$. There exist k cuttings Ξ_1, \dots, Ξ_s so that Ξ_i is a $(1/2^i)$ -cutting of size $O(2^{id})$, each simplex of Ξ_i is contained in a simplex of Ξ_{i-1} , and each simplex of Ξ_{i-1} contains a constant number of simplices of Ξ_i . Moreover, Ξ_1, \dots, Ξ_s can be computed in time $O(nr^{d-1})$.

The above approach can be extended to the simplex range-counting problem as well. That is, store the solution of every combinatorially distinct simplex (two simplices are combinatorially distinct if they do not contain the same subset of S). Since there are $\Theta(n^{d(d+1)})$ combinatorially distinct simplices, such an approach will require $\Omega(n^{d(d+1)})$ storage. Chazelle et al. [CSW92] showed that the size can be reduced to $O(n^{d+\epsilon})$, for any $\epsilon > 0$, using a multilevel data structure, with each level composed of a halfspace range-counting data structure. The space bound can be reduced to $O(n^d)$ by increasing the query time to $O(\log^{d+1} n)$ [Mat93]. Halfspace range-reporting queries can be answered in $O(\log n + k)$ time, using $O(n^{\lfloor d/2 \rfloor} \text{polylog} n)$ space.

A space/query-time tradeoff for simplex range searching can be attained by combining the linear-size and logarithmic query-time data structures. The known results on this tradeoff are summarized in Table 36.3.2. $Q(m, n)$ is the query time on n points using m units of storage.

TABLE 36.3.2 Space/query-time tradeoff.

RANGE	MODE	$Q(m, n)$
Simplex	reporting	$\frac{n}{m^{1/d}} \log^{d+1} \frac{m}{n} + k$
Simplex	counting	$\frac{n}{m^{1/d}} \log^{d+1} \frac{m}{n}$
Halfspace	reporting	$\frac{n}{m^{1/\lfloor d/2 \rfloor}} \log^c n + k$
Halfspace	emptiness	$\frac{n}{m^{1/\lfloor d/2 \rfloor}} \log^c n$
Halfspace	counting	$\frac{n}{m^{1/d}} \log \frac{m}{n}$

LOWER BOUNDS

Fredman [Fre81b] showed that a sequence of n insertions, deletions, and halfplane queries on a set of points in the plane requires $\Omega(n^{4/3})$ time, in the semigroup model. His technique, however, does not extend to static data structures. In a series of papers, Chazelle has proved nontrivial lower bounds on the complexity of on-line simplex range searching, using various elegant mathematical techniques. The following theorem is perhaps the most interesting result on lower bounds.

THEOREM 36.3.3 Chazelle [Cha89]

Let (S, \oplus) be a faithful semigroup, let n, m be positive integers such that $n \leq m \leq$

n^d , and let S be a random set of weighted points in $[0, 1]^d$ with weights from \mathbf{S} . If only m words of storage is available, then with high probability, the worst-case query time for a simplex range query in S is $\Omega(n/\sqrt{m})$ for $d = 2$, or $\Omega(n/(m^{1/d} \log n))$ for $d \geq 3$, in the semigroup model.

It should be pointed out that this theorem holds even if the query ranges are wedges or strips, but it does not hold if the ranges are hyperplanes. Chazelle and Rosenberg [CR96] proved a lower bound of $\Omega(n^{1-\epsilon}/m + k)$ for simplex range reporting under the pointer-machine model. These lower bounds do not hold for halfspace range searching. A somewhat weaker lower bound for halfspace queries was proved by Brönnimann et al. [BCP93].

As we saw earlier, faster data structures are known for halfspace emptiness queries. A series of papers by Erickson established the first nontrivial lower bounds for on-line and off-line emptiness query problems, in the partition-graph model of computation. He first considered this model for Hopcroft's problem—Given a set of n points and m lines, does any point lie on a line?—for which he obtained a lower bound of $\Omega(n \log m + n^{2/3}m^{2/3} + m \log n)$ [Eri96b], almost matching the best known upper bound $O(n \log m + n^{2/3}m^{2/3}2^{O(\log^*(n+m))}) + m \log n$, due to Matoušek [Mat93]. He later established lower bounds on a tradeoff between space and query time, or preprocessing and query time, for on-line hyperplane emptiness queries [Eri00]. For d -dimensional hyperplane queries, $\Omega(n^d/\text{polylog}n)$ preprocessing time is required to achieve polylogarithmic query time, and the best possible query time is $\Omega(n^{1/d}/\text{polylog}n)$ if only $O(n\text{polylog}n)$ preprocessing time is allowed. More generally, in two dimensions, if the preprocessing time is p , the query time is $\Omega(n/\sqrt{p})$.

Table 36.3.3 summarizes the best lower bounds known for on-line simplex queries. Lower bounds for emptiness problems apply to counting and reporting problems as well.

TABLE 36.3.3 Lower bounds for on-line simplex range searching using $O(m)$ space.

Range	Problem	Model	Query Time
Simplex	Semigroup	Semigroup ($d = 2$)	n/\sqrt{m}
	Semigroup	Semigroup ($d > 2$)	$n/(m^{1/d} \log n)$
	Reporting	Pointer machine	$n^{1-\epsilon}/m^{1/d} + k$
Hyperplane	Semigroup	Semigroup	$(n/m^{1/d})^{2/(d+1)}$
	Emptiness	Partition graph	$(n/\log n)^{\frac{d^2+1}{d^2+d}} \cdot (1/m^{1/d})$
Halfspace	Semigroup	Semigroup	$(n/\log n)^{\frac{d^2+1}{d^2+d}} \cdot (1/m^{1/d})$
	Emptiness	Partition graph	$(n/\log n)^{\frac{\delta^2+1}{\delta^2+\delta}} \cdot (1/m^{1/\delta})$, where $d \geq \delta(\delta + 3)/2$

OPEN PROBLEMS

- Bridge the gap between the known upper and lower bounds in the group model. Even in the semigroup model there is a small gap between the known bounds.

-
2. Can a halfspace range-reporting query be answered in $O(n^{1-1/\lfloor d/2 \rfloor} + k)$ time using linear space if d is odd?
-

36.4 VARIANTS AND EXTENSIONS

In this section we review a few extensions of range-searching data structures: multilevel structures, semialgebraic range searching, and kinetic range searching.

GLOSSARY

Semialgebraic set: A subset of \mathbb{R}^d obtained as a finite Boolean combination of sets of the form $\{f \geq 0\}$, where f is a d -variate polynomial (see [Chapter 29](#)).

Tarski cells: A simply connected real semialgebraic set defined by a constant number of polynomials, each of constant degree.

MULTI-LEVEL STRUCTURES

A powerful property of data structures based on decomposition schemes (described in Section 36.1) is that they can be cascaded together to answer more complex queries, at the increase of a logarithmic factor per level in their performance. The real power of the cascading property was first observed by Dobkin and Edelsbrunner [DE87], who used this property to answer several complex geometric queries. Since their result, several papers have exploited and extended this property to solve numerous geometric-searching problems. We briefly sketch the general cascading scheme.

Let S be a set of weighted objects. Recall that a geometric-searching problem \mathcal{P} , with underlying relation \diamond , requires computing $\sum_{p \diamond \gamma} w(p)$ for a query range γ . Let \mathcal{P}^1 and \mathcal{P}^2 be two geometric-searching problems, and let \diamond^1 and \diamond^2 be the corresponding relations. Then we define $\mathcal{P}^1 \circ \mathcal{P}^2$ to be the conjunction of \mathcal{P}^1 and \mathcal{P}^2 , whose relation is $\diamond^1 \cap \diamond^2$. That is, for a query range γ , we want to compute $\sum_{p \diamond^1 \gamma, p \diamond^2 \gamma} w(p)$. Suppose we have hierarchical decomposition schemes \mathcal{D}^1 and \mathcal{D}^2 for problems \mathcal{P}^1 and \mathcal{P}^2 . Let $\mathcal{F}^1 = \mathcal{F}^1(S)$ be the set of canonical subsets constructed by \mathcal{D}^1 , and for a range γ , let $\mathcal{C}_\gamma^1 = \mathcal{C}^1(S, \gamma)$ be the corresponding partition of $\{p \in S \mid p \diamond^1 \gamma\}$ into canonical subsets. For each canonical subset $C \in \mathcal{F}^1$, let $\mathcal{F}^2(C)$ be the collection of canonical subsets of C constructed by \mathcal{D}^2 , and let $\mathcal{C}^2(C, \gamma)$ be the corresponding partition of $\{p \in C \mid p \diamond^2 \gamma\}$ into level-two canonical subsets. The decomposition scheme $\mathcal{D}^1 \circ \mathcal{D}^2$ for the problem $\mathcal{P}^1 \circ \mathcal{P}^2$ consists of the canonical subsets $\mathcal{F} = \bigcup_{C \in \mathcal{F}^1} \mathcal{F}^2(C)$. For a query range γ , the query output is $\mathcal{C}_\gamma = \bigcup_{C \in \mathcal{C}_\gamma^1} \mathcal{C}^2(C, \gamma)$. We can cascade any number of decomposition schemes in this manner.

If we view \mathcal{D}^1 and \mathcal{D}^2 as tree data structures, then cascading the two decomposition schemes can be regarded as constructing a two-level tree, as follows. We first construct the tree induced by \mathcal{D}^1 on S . Each node v of \mathcal{D}^1 is associated with a canonical subset C_v . We construct a second-level tree \mathcal{D}_v^2 on C_v and store \mathcal{D}_v^2 at v as its secondary structure. A query is answered by first identifying the nodes that

correspond to the canonical subsets $C_v \in \mathcal{C}_\gamma^1$ and then searching the corresponding secondary trees to compute the second-level canonical subsets $\mathcal{C}^2(C_v, \gamma)$.

Suppose the size and query time of each decomposition scheme are at most $S(n)$ and $Q(n)$, respectively, and \mathcal{D}^1 is efficient and r -convergent (cf. Section 36.1), for some constant $r > 1$. Then the size and query time of the decomposition scheme \mathcal{D} are $O(S(n) \log_r n)$ and $O(Q(n) \log_r n)$, respectively. If \mathcal{D}^2 is also efficient and r -convergent, then \mathcal{D} is efficient and r -convergent. In some cases, the logarithmic overhead in the query time or the space can be avoided.

The real power of multilevel data structures stems from the fact that there are no restrictions on the relations \diamondsuit^1 and \diamondsuit^2 . Hence, any query that can be represented as a conjunction of a constant number of “primitive” queries, each of which admits an efficient, r -convergent decomposition scheme, can be answered by cascading individual decomposition schemes. We will describe a few multilevel data structures in this and the following sections.

SEMIALGEBRAIC RANGE SEARCHING

So far we have assumed that the ranges were bounded by hyperplanes, but in many applications one has to deal with ranges bounded by nonlinear functions. For example, a query of the form, “for a given point p and a real number r , find all points of S lying within distance r from p ,” is a range-searching problem in which the ranges are balls.

As shown below, ball range searching in \mathbb{R}^d can be formulated as an instance of the halfspace range searching in \mathbb{R}^{d+1} . So a ball range-reporting (resp. range-counting) query in \mathbb{R}^d can be answered in time $O(n/m^{1/\lceil d/2 \rceil} \log^c n + k)$ (resp. $O(n/m^{1/(d+1)} \log(m/n))$), using $O(m)$ space; somewhat better performance can be obtained using a more direct approach (Table 36.4.1). However, relatively little is known about range-searching data structures for more general ranges.

A natural class of nonlinear ranges is the family of Tarski cells. It suffices to consider the ranges bounded by a single polynomial because the ranges bounded by multiple polynomials can be handled using multilevel data structures. We assume that the ranges are of the form

$$\Gamma_f(a) = \{x \in \mathbb{R}^d \mid f(x, a) \geq 0\},$$

where f is a $(d+p)$ -variate polynomial specifying the type of ranges (disks, cylinders, cones, etc.), and a is a p -tuple specifying a specific range of the given type (e.g., a specific disk). We will refer to the range-searching problem in which the ranges are from the set Γ_f as **Γ_f -range searching**.

One approach to answering Γ_f -range queries is **linearization**. We represent the polynomial $f(x, a)$ in the form

$$f(x, a) = \psi_0(a) + \psi_1(a)\varphi_1(x) + \cdots + \psi_\lambda(a)\varphi_\lambda(x)$$

where $\varphi_1, \dots, \varphi_\lambda, \psi_0, \dots, \psi_\lambda$ are real functions. A point $x \in \mathbb{R}^d$ is mapped to the point

$$\varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_\lambda(x)) \in \mathbb{R}^\lambda.$$

Then a range $\gamma_f(a) = \{x \in \mathbb{R}^d \mid f(x, a) \geq 0\}$ is mapped to a halfspace

$$\varphi^\#(a) : \{y \in \mathbb{R}^\lambda \mid \psi_0(a) + \psi_1(a)y_1 + \cdots + \psi_\lambda(a)y_\lambda \geq 0\};$$

TABLE 36.4.1 Semialgebraic range counting; λ is the dimension of linearization.

d	RANGE	$S(n)$	$Q(n)$	NOTES
$d = 2$	disk	$n \log n$	$\sqrt{n \log n}$	
$d \leq 4$	Tarski cell	n	$n^{1-1/d+\epsilon}$	partition tree
$d \geq 4$	Tarski cell	n	$n^{1-\frac{1}{2d-4}+\epsilon}$	partition tree
	Tarski cell	n	$n^{1-\frac{1}{\lambda}+\epsilon}$	linearization
	disk	n	$n^{1-\frac{1}{d}+\epsilon}$	linearization

λ is called the *dimension* of linearization. For example, a set of spheres in \mathbb{R}^d admit a linearization of dimension $d + 1$, using the well-known lifting transform. Agarwal and Matoušek [AM94] have described an algorithm for computing a linearization of the smallest dimension under certain assumptions on φ_i 's and ψ_i 's. If f admits a linearization of dimension λ , a Γ_f -range query can be answered using a λ -dimensional halfspace range-searching data structure. Agarwal and Matoušek [AM94] have also proposed another approach to answer Γ_f -range queries, by extending Theorem 36.3.1 to Tarski cells and by constructing partition trees using this extension. Table 36.4.1 summarizes the known results on Γ_f -range-counting queries. The bounds mentioned in the third row of the table rely on the result by Koltun [Kol01] on the vertical decomposition of arrangements of surfaces.

KINETIC RANGE SEARCHING

Let $S = \{p_1, \dots, p_n\}$ be a set of n points in \mathbb{R}^2 , each moving continuously. Let $p_i(t)$ denote the position of p_i at time t , and let $S(t) = \{p_1(t), \dots, p_n(t)\}$. We assume that each point p_i is moving with fixed velocity, i.e., $p_i(t) = a_i + b_i t$ for $a_i, b_i \in \mathbb{R}^2$, and the trajectory of a point p_i is a line \bar{p}_i . Let L denote the set of lines corresponding to the trajectories of points in S .

We consider the following two range-reporting queries:

- Q1.** Given an axis-aligned rectangle R in the xy -plane and a time value t_q , report all points of S that lie inside R at time t_q , i.e., report $S(t_q) \cap R$; t_q is called the *time stamp* of the query.
- Q2.** Given a rectangle R and two time values $t_1 \leq t_2$, report all points of S that lie inside R at any time between t_1 and t_2 , i.e., report $\bigcup_{t=t_1}^{t_2} (S(t) \cap R)$.

Two general approaches have been proposed to preprocess moving points for range searching. The first approach, which is known as the *time-oblivious* approach, regards time as a new dimension and stores the trajectories \bar{p}_i of input points p_i . One can either preprocess the trajectories themselves using various techniques, or one can work in a parametric space, map each trajectory to a point in this space, and build a data structure on these points. An advantage of the time-oblivious scheme is that the data structure is updated only if the trajectory of a point changes or if a point is inserted into or deleted from the index. Since this approach preprocesses either curves in \mathbb{R}^2 or points in higher dimensions, the query time tends to be large. For example, if S is a set of points moving in \mathbb{R}^1 , then the trajectory of each point

is a line in \mathbb{R}^2 and a Q1 query corresponds to reporting all lines of L that intersect a query segment σ parallel to the x -axis. As we will see below, L can be preprocessed into a data structure of linear size so that all lines intersecting σ can be reported in $O(n^{1/2+\epsilon} + k)$ time. A similar structure can answer Q2 queries within the same asymptotic time bound. The lower bounds on simplex range searching suggest that one cannot hope to answer a query in $O(\log n + k)$ time using this approach. If S is a set of points moving in \mathbb{R}^2 , then a Q1 query asks for reporting all lines of L that intersect a query rectangle R parallel to the xy -plane (in the xyt -space). A line ℓ in \mathbb{R}^3 (xyt -space) intersects R if and only if their projections onto the xt - and yt -planes both intersect. Using this observation one can construct a two-level partition tree of size $O(n)$ to report in $O(n^{1/2+\epsilon} + k)$ time all lines of L intersecting R [AAE03]. Again a Q2 query can be answered within the same time bound.

The second approach, based on the *kinetic-data-structure* framework [Gui98], builds a dynamic data structure on the moving points (see [Chapter 50](#)). Roughly speaking, at any time it maintains a data structure on the current configuration of the points. As the points move, the data structure evolves. The main observation is that although the points are moving continuously, the data structure is updated only at discrete time instances when certain *events* occur, e.g., when any of the coordinates of two points become equal. This approach leads to fast query time, but at the cost of updating the structure periodically even if the trajectory of no point changes. Another disadvantage of this approach is that it can answer a query only at the current configurations of points, though it can be extended to handle queries arriving in chronological order, i.e., the time stamps of queries are in nondecreasing order. In particular, if S is a set of points moving in \mathbb{R}^1 , using a kinetic B -tree, a one-dimensional Q1 query can be answered in $O(\log n + k)$ time. The data structure processes $O(n^2)$ events, each of which requires $O(\log n)$ time. Similarly, by kinetizing range trees, a two-dimensional Q1 query can be answered in $O(\log n + k)$ time; the data structure processes $O(n^2)$ events, each of which requires $O(\log^2 n / \log \log n)$ time [AAE03].

Since range trees are too complicated, a more practical approach is to use the kinetic-data-structure framework on k -d-trees, as proposed by Agarwal et al. [AGG02]. They propose two variants of kinetic k -d-trees, each of which answers Q1 queries that arrive in chronological order in $O(n^{1/2+\epsilon})$ time, for any constant $\epsilon > 0$, process $O(n^2)$ kinetic events, and spend $O(\text{polylog } n)$ time at each event. Since kinetic k -d-trees process too many events because of the strong invariants they maintain, kinetic R-trees have also been proposed [JLL00, PAHP02], which typically require weaker invariants and thus are updated less frequently.

OPEN PROBLEMS

1. Can a ball range-counting query be answered in $O(\log n)$ time using $O(n^2)$ space?
2. If the hyperplanes bounding the query halfspaces satisfy some property—e.g., all of them are tangent to a given sphere—can a halfspace range-counting query be answered more efficiently?
3. Is there a simple, linear-size kinetic data structure that can answer Q1 queries

in $O(\sqrt{n} + k)$ time and processes near-linear events, each requiring $O(\log^c n)$ time?

36.5 INTERSECTION SEARCHING

A general intersection-searching problem can be formulated as follows: *Given a set S of objects in \mathbb{R}^d , a semigroup $(\mathbf{S}, +)$, and a weight function $w : S \rightarrow \mathbf{S}$; we wish to preprocess S into a data structure so that for a query object γ , we can compute the weighted sum $\sum w(p)$, where the sum is taken over all objects of S that intersect γ .* Range searching is a special case of intersection-searching in which S is a set of points.

An intersection-searching problem can be formulated as a semialgebraic range-searching problem by mapping each object $p \in S$ to a point $\varphi(p)$ in a parametric space \mathbb{R}^λ and every query range γ to a semialgebraic set $\psi(\gamma)$ so that p intersects γ if and only if $\varphi(p) \in \psi(\gamma)$. For example, let both S and the query ranges be sets of segments in the plane. Each segment $e \in S$ with left and right endpoints (p_x, p_y) and (q_x, q_y) , respectively, can be mapped to a point $\varphi(e) = (p_x, p_y, q_x, q_y)$ in \mathbb{R}^4 , and a query segment γ can be mapped to a semialgebraic region $\psi(\gamma)$ so that γ intersects e if and only if $\psi(\gamma) \in \varphi(e)$. A shortcoming of this approach is that λ , the dimension of the parametric space, is typically much larger than d , and thereby affecting the query time adversely. The efficiency can be significantly improved by expressing the intersection test as a conjunction of simple primitive tests (in low dimensions) and using a multilevel data structure to perform these tests. For example, a segment γ intersects another segment e if the endpoints of e lie on the opposite sides of the line containing γ and vice versa. We can construct a two-level data structure—the first level sifts the subset $S_1 \subseteq S$ of all the segments that intersect the line supporting the query segment, and the second level reports those segments of S_1 whose supporting lines separate the endpoints of γ . Each level of this structure can be implemented using a two-dimensional simplex range-searching searching structure, and hence a reporting query can be answered in $O(n/\sqrt{m} \log^c n + k)$ time using $O(m)$ space.

It is beyond the scope of this chapter to cover all intersection-searching problems. Instead, we discuss a selection of basic problems that have been studied extensively. All intersection-counting data structures described here can answer intersection-reporting queries at an additional cost proportional to the output size. In some cases an intersection-reporting query can be answered faster. Moreover, using intersection-reporting data structures, intersection-detection queries can be answered in time proportional to their query-search time. Finally, all the data structures described in this section can be dynamized at the expense of an $O(n^\epsilon)$ factor in the storage and query time.

POINT INTERSECTION SEARCHING

Preprocess a set S of objects (e.g., balls, halfspaces, simplices, Tarski cells) in \mathbb{R}^d into a data structure so that the objects of S containing a query point can be reported (or counted) efficiently. This is the inverse of the range-searching problem, and it

can also be viewed as locating a point in the subdivision induced by the objects in S . Table 36.5.1 gives some of the known results.

TABLE 36.5.1 Point intersection searching.

d	OBJECTS	$S(n)$	$Q(n)$	NOTES
$d = 2$	disks	m	$(n/\sqrt{m})^{4/3}$	counting
	disks	$n \log n$	$\log n + k$	reporting
	triangles	m	$\frac{n}{\sqrt{m}} \log^3 n$	counting
	fat triangles	$n \log^2 n$	$\log^3 n + k$	reporting
	Tarski cells	$n^{2+\epsilon}$	$\log n$	counting
$d = 3$	functions	$n^{1+\epsilon}$	$\log n + k$	reporting
	Tarski cells	$n^{3+\epsilon}$	$\log n$	counting
$d \geq 3$	simplices	m	$\frac{n}{m^{1/d}} \log^{d+1} n$	counting
	balls	$n^{d+\epsilon}$	$\log n$	counting
	balls	m	$\frac{n}{m^{1/\lceil d/2 \rceil}} \log^c n + k$	reporting
$d \geq 4$	Tarski cells	$n^{2d-4+\epsilon}$	$\log n$	counting

SEGMENT INTERSECTION SEARCHING

Preprocess a set of objects in \mathbb{R}^d into a data structure so that the objects of S intersected by a query segment can be reported (or counted) efficiently. See Table 36.5.2 for some of the known results on segment intersection searching. For the sake of clarity, we have omitted polylogarithmic factors from the query-search time whenever it is of the form n/m^α .

TABLE 36.5.2 Segment intersection searching.

d	OBJECTS	$S(n)$	$Q(n)$	NOTES
$d = 2$	simple polygon	n	$(k+1) \log n$	reporting
	segments	m	n/\sqrt{m}	counting
	circles	$n^{2+\epsilon}$	$\log n$	counting
	circular arcs	m	$n/m^{1/3}$	counting
$d = 3$	planes	m	$n/m^{1/3}$	counting
	spheres	m	$n/m^{1/3}$	counting
	triangles	m	$n/m^{1/4}$	counting

A special case of segment intersection searching, in which the objects are horizontal segments in the plane and query ranges are vertical segments, has been widely studied. In this case a query can be answered in time $O(\log n + k)$ using

$O(n \log \log n)$ space and $O(n \log n)$ preprocessing (in the RAM model), and a point can be inserted or deleted in $O(\log n)$ time [Mor03]. Slightly weaker bounds are known in the pointer-machine model.

COLORED INTERSECTION SEARCHING

Preprocess a given set S of colored objects in \mathbb{R}^d (i.e., each object in S is assigned a color) so that we can report (or count) the colors of the objects that intersect the query range. This problem arises in many contexts in which one wants to answer intersection-searching queries for nonconstant-size input objects. For example, given a set $P = \{P_1, \dots, P_m\}$ of m simple polygons, one may wish to report all polygons of P that intersect a query segment; the goal is to return the indices, and not the description, of these polygons. If we color the edges of P_i with color i , the problem reduces to colored segment intersection searching in a set of segments.

A colored orthogonal range searching query for points on a two-dimensional grid $[0, U]^2$ can be answered in $O(\log \log U + k)$ time using $O(n \log^2 U)$ storage and $O(n \log n \log^2 U)$ preprocessing [AGM02]. On the other hand, a set S of n colored rectangles in the plane can be stored into a data structure of size $O(n \log n)$ so that the colors of all rectangles in S that contain a query point can be reported in time $O(\log n + k)$ [BKMT97]. If the vertices of the rectangles in S and all the query points lie on the grid $[0, U]^2$, the query time can be improved to $O(\log \log U + k)$ by increasing the storage to $O(n^{1+\epsilon})$.

Gupta et al. [GJS94] have shown that the colored halfplane-reporting queries in the plane can be answered in $O(\log^2 n + k)$ using $O(n \log n)$ space. Agarwal and van Kreveld [AvK96] presented a linear-size data structure with $O(n^{1/2+\epsilon} + k)$ query time for colored segment intersection-reporting queries amidst a set of segments in the plane, assuming that the segments of the same color form a connected planar graph or the boundary of a simple polygon; these data structures can also handle insertions of new segments.

36.6 OPTIMIZATION QUERIES

In optimization queries, we want to return an object that satisfies certain conditions with respect to a query range. The most common example of optimization queries is, perhaps, ray-shooting queries. Other examples include segment-dragging and linear-programming queries.

RAY-SHOOTING QUERIES

Preprocess a set S of objects in \mathbb{R}^d into a data structure so that the first object (if one exists) intersected by a query ray can be reported efficiently. This problem arises in ray tracing, hidden-surface removal, radiosity, and other graphics problems. Efficient solutions to many geometric problems have also been developed using ray-shooting data structures.

A general approach to the ray-shooting problem, using segment intersection-detection structures and Megiddo's parametric-searching technique ([Chapter 37](#)),

was proposed by Agarwal and Matoušek [AM93]. The basic idea of their approach is as follows. Suppose we have a segment intersection-detection data structure for S , based on partition trees. Let ρ be a query ray. Their algorithm maintains a segment $\vec{ab} \subseteq \rho$ so that the first intersection point of \vec{ab} with S is the same as that of ρ . If a lies on an object of S , it returns a . Otherwise, it picks a point $c \in \vec{ab}$ and determines, using the segment intersection-detection data structure, whether the interior of the segment \vec{ac} intersects any object of S . If the answer is YES, it recursively finds the first intersection point of \vec{ac} with S ; otherwise, it recursively finds the first intersection point of \vec{cb} with S . Using parametric searching, the point c at each stage can be chosen so that the algorithm terminates after $O(\log n)$ steps.

In some cases the query time can be improved by a polylogarithmic factor using a more direct approach.

TABLE 36.6.1 Ray shooting.

d	OBJECTS	$S(n)$	$Q(n)$
$d = 2$	simple polygon	n	$\log n$
	s disjoint polygons	n	$\sqrt{s} \log n$
	s disjoint polygons	$(s^2 + n) \log s$	$\log s \log n$
	s convex polygons	$sn \log s$	$\log s \log n$
	segments	m	n/\sqrt{m}
	circular arcs	m	$n/m^{1/3}$
	disjoint arcs	n	\sqrt{n}
$d = 3$	convex polytope	n	$\log n$
	c -oriented polytopes	n	$\log n$
	s convex polytopes	$s^2 n^{2+\epsilon}$	$\log^2 n$
	halfplanes	m	n/\sqrt{m}
	terrain	m	n/\sqrt{m}
	triangles	m	$n/m^{1/4}$
	spheres	m	$n/m^{1/3}$
$d > 3$	hyperplanes	m	$n/m^{1/d}$
	hyperplanes	$\frac{n^d}{\log^{d-\epsilon} n}$	$\log n$
	convex polytope	m	$n/m^{1/\lfloor d/2 \rfloor}$

Table 36.6.1 gives a summary of known ray-shooting results. For the sake of clarity, we have ignored the polylogarithmic factors in the query time whenever it is of the form n/m^α .

Like simplex range searching, many practical data structures have been proposed that, despite having poor worst-case performance, work well in practice. One common approach is to construct a subdivision of \mathbb{R}^d into constant-size cells so that the interior of each cell does not intersect any object of S . A ray-shooting query can be answered by traversing the query ray through the subdivision until we find an object that intersects the ray. The worst-case query time is proportional to the maximum number of cells intersected by a segment that does not intersect any object in S . Hershberger and Suri [HS95] showed that a triangulation with $O(\log n)$

query time can be constructed when S is the boundary of a simple polygon in the plane. Agarwal et al. [AAS95] proved worst-case bounds for many cases on the number of cells in the subdivision that a line can intersect. Aronov and Fortune [AF99] have obtained a bound on the expected number of cells in the subdivision that a line can intersect.

LINEAR-PROGRAMMING QUERIES

Let S be a set of n halfspaces in \mathbb{R}^d . We wish to preprocess S into a data structure so that for a direction vector \vec{v} , we can determine the first point of $\bigcap_{h \in S} h$ in the direction \vec{v} . For $d \leq 3$, such a query can be answered in $O(\log n)$ time using $O(n)$ storage, by constructing the normal diagram of the convex polytope $\bigcap_{h \in S} h$ and preprocessing it for point-location queries. For higher dimensions, Ramos [Ram00] has proposed two data structures. His first structure can answer a query in time $(\log n)^{O(\log d)}$ using $n^{\lfloor d/2 \rfloor} \log^{O(1)} n$ space and preprocessing, and his second structure can answer a query in time $n^{1-1/\lfloor d/2 \rfloor} 2^{O(\log^* n)}$ using $O(n)$ space and $O(n^{1+\epsilon})$ preprocessing.

SEGMENT-DRAGGING QUERIES

Preprocess a set S of objects in the plane so that for a query segment e and a ray ρ , the first position at which e intersects any object of S as it is translated (dragged) along ρ can be determined quickly. This query can be answered in $O((n/\sqrt{m}) \log^c n)$ time, with $O(m)$ storage, using segment-intersection searching structures and the parametric-search technique. Chazelle [Cha88a] gave a linear-size, $O(\log n)$ query-time data structure for the special case in which S is a set of points, e is a horizontal segment, and ρ is the vertical direction. Instead of dragging a segment along a ray, one can ask the same question for dragging along a more complex trajectory (along a curve, and allowing both translation and rotation). These problems arise naturally in motion planning and manufacturing.

36.7 SOURCES AND RELATED MATERIAL

RELATED READING

Books and Monographs

[Meh84]: Multidimensional searching and computational geometry.

[dBvKOS97]: Basic topics in computational geometry.

[Mul93]: Randomized techniques in computational geometry. [Chapters 6](#) and [8](#) cover range-searching, intersection-searching, and ray-shooting data structures.

[Cha01]: Covers lower bound techniques, ϵ -nets, cuttings, and simplex range searching.

[MTT99, Sam90]: Range-searching data structures in spatial database systems.

Survey Papers

[AE99, Mat94]: Range-searching data structures.

[GG98, NW00, ST99] Indexing techniques used in databases.

[AP02]: Range-searching data structures for moving points.

[Arg02]: Secondary-memory data structures.

[Chan01]: Ray-shooting data structures.

RELATED CHAPTERS

[Chapter 24: Arrangements](#)

[Chapter 34: Point location](#)

[Chapter 37: Ray shooting and lines in space](#)

[Chapter 39: Nearest-neighbor searching in high dimensions](#)

[Chapter 44: Geometric discrepancy theory and uniform distribution](#)

[Chapter 50: Modeling motion](#)

REFERENCES

- [AAE03] P.K. Agarwal, L. Arge, and J. Erickson. Indexing moving points. *J. Comput. Syst. Sci.*, 66:207–243, 2003.
- [AAE⁺00] P.K. Agarwal, L. Arge, J. Erickson, P.G. Francis, and J.S. Vitter. Efficient searching with linear constraints. *J. Comput. Syst. Sci.*, 61:194–216, 2000.
- [AAPV01] P.K. Agarwal, L. Arge, O. Procopiuc, and J.S. Vitter. A framework for index bulk loading and dynamization. In *Proc. 28th Internat. Conf. Automata Program. Langs.*, pages 115–127, 2001.
- [AAS95] P.K. Agarwal, B. Aronov, and S. Suri. Stabbing triangulations by lines in 3d. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 267–276, New York, 1995.
- [AdBG⁺02] P.K. Agarwal, M. de Berg, J. Gudmundsson, M. Hammar, and H.J. Haverkort. Box-trees and R-trees with near-optimal query time. *Discrete Comput. Geom.*, 26:291–312, 2002.
- [AE99] P.K. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary Mathematics*, pages 1–56. Amer. Math. Soc., Providence, 1999.
- [AGG02] P.K. Agarwal, J. Gao, and L.J. Guibas. Kinetic medians and kd-trees. In *Proc. 10th European Sympos. Algorithms*, pages 15–26, 2002.
- [AGM02] P.K. Agarwal, S. Govindarajan, and S. Muthukrishnan. Range searching in categorical data: Colored range searching on grid. In *Proc. 10th European Sympos. Algorithms*, pages 17–28, 2002.
- [AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.

- [AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.
- [AP02] P.K. Agarwal and C.M. Procopiuc. Advances in indexing mobile objects. *IEEE Bull Data Eng.*, 25:25–34, 2002.
- [AvK96] P.K. Agarwal and M. van Kreveld. Polygon and connected component intersection searching. *Algorithmica*, 15:626–660, 1996.
- [Arg02] L. Arge. External memory data structures. In J. Abello, P.M. Pardalos, and M.G.C. Resende, editors, *Handbook of Massive Data Sets*, pages 313–358. Kluwer Academic Publishers, Boston, 2002.
- [ASV99] L. Arge, V. Samoladas, and J.S. Vitter. On two-dimensional indexability and optimal range search indexing. In *Proc. Annu. ACM Symp. Principles Database Syst.*, pages 346–357, 1999.
- [AF99] B. Aronov and S.J. Fortune. Approximating minimum-weight triangulations in three dimensions. *Discrete Comput. Geom.*, 21:527–549, 1999.
- [Ben75] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18:509–517, 1975.
- [Ben80] J.L. Bentley. Multidimensional divide-and-conquer. *Commun. ACM*, 23:214–229, 1980.
- [BKMT97] P. Bozanis, N. Ktsios, C. Makris, and A. Tsakalidis. New results on intersection query problems. *Comput. J.*, 40:22–29, 1997.
- [BCP93] H. Brönnimann, B. Chazelle, and J. Pach. How hard is halfspace range searching. *Discrete Comput. Geom.*, 10:143–155, 1993.
- [Chan01] A.Y. Chang. A survey of geometric data structures for ray tracing. Tech. Report TR-CIS-2001-06, Polytechnic Univ., New York, 2001.
- [CL01] B. Chazelle and A. Lvov. A trace bound for hereditary discrepancy. *Discrete Comput. Geom.*, 26:221–232, 2001.
- [Cha86] B. Chazelle. Filtering search: a new approach to query-answering. *SIAM J. Comput.*, 15:703–724, 1986.
- [Cha88a] B. Chazelle. An algorithm for segment-dragging and its implementation. *Algorithmica*, 3:205–221, 1988.
- [Cha88b] B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17:427–462, 1988.
- [Cha89] B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2:637–666, 1989.
- [Cha90a] B. Chazelle. Lower bounds for orthogonal range searching, I: The reporting case. *J. Assoc. Comput. Mach.*, 37:200–212, 1990.
- [Cha90b] B. Chazelle. Lower bounds for orthogonal range searching, II: The arithmetic model. *J. Assoc. Comput. Mach.*, 37:439–463, 1990.
- [Cha93] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9:145–158, 1993.
- [Cha97] B. Chazelle. Lower bounds for off-line range searching. *Discrete Comput. Geom.*, 17:53–66, 1997.
- [Cha98] B. Chazelle. A spectral approach to lower bounds with applications to geometric searching. *SIAM J. Comput.*, 27:545–556, 1998.

- [Cha01] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, 2001.
- [CR89] B. Chazelle and B. Rosenberg. Computing partial sums in multidimensional arrays. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 131–139, 1989.
- [CR96] B. Chazelle and B. Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom. Theory Appl.*, 5:237–247, 1996.
- [CSW92] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8:407–429, 1992.
- [dBvKOS97] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 1997.
- [DR91] P.F. Dietz and R. Raman. Persistence, amortization and randomization. In *Proc. ACM-SIAM Sympos. Discrete Algorithms*, pages 78–88. 1991.
- [DE87] D.P. Dobkin and H. Edelsbrunner. Space searching for intersecting objects. *J. Algorithms*, 8:348–361, 1987.
- [Eri96a] J. Erickson. New lower bounds for halfspace emptiness. In *Proc. 37th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 472–481, 1996.
- [Eri96b] J. Erickson. New lower bounds for Hopcroft’s problem. *Discrete Comput. Geom.*, 16:389–418, 1996.
- [Eri00] J. Erickson. Space-time tradeoffs for emptiness queries. *SIAM J. Comput.*, 19:1968–1996, 2000.
- [ELS97] G. Evangelidis, D.B. Lomet, and B. Salzberg. The hB^Π -tree: A multi-attribute index supporting concurrency, recovery and node consolidation. *VLDB J.*, 6:1–25, 1997.
- [FB74] R.A. Finkel and J.L. Bentley. Quad trees: a data structure for retrieval on composite keys. *Acta Inform.*, 4:1–9, 1974.
- [Fre79] M.L. Fredman. The complexity of maintaining an array and computing its partial sums. *J. Assoc. Comput. Mach.*, 29:250–260, 1979.
- [Fre80] M.L. Fredman. The inherent complexity of dynamic data structures which accommodate range queries. In *Proc. 21st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 191–199, 1980.
- [Fre81a] M.L. Fredman. A lower bound on the complexity of orthogonal range queries. *J. Assoc. Comput. Mach.*, 28:696–705, 1981.
- [Fre81b] M.L. Fredman. Lower bounds on the complexity of some optimal data structures. *SIAM J. Comput.*, 10:1–10, 1981.
- [GG98] V. Gaede and O. Günther. Multidimensional access methods. *ACM Comput. Surv.*, 30:170–231, 1998.
- [GAA03] S. Govindarajan, P.K. Agarwal, and L. Arge. CRB-tree: An efficient indexing scheme for range aggregate queries. In *Proc. 9th Internat. Conf. Database Theory*, 2003.
- [GBLP96] J. Gray, A. Bosworth, A. Layman, and H. Patel. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In *Proc. 12th IEEE Internat. Conf. Data Eng.*, pages 152–159, 1996.
- [Gui98] L.J. Guibas. Kinetic data structures—a state of the art report. In P.K. Agarwal, L.E. Kavraki, and M. Mason, editors, *Proc. Workshop Algorithmic Found. Robot.*, pages 191–209. A.K. Peters, Wellesley, 1998.

- [GJS94] P. Gupta, R. Janardan, and M. Smid. Efficient algorithms for generalized intersection searching on non-iso-oriented objects. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 369–378, 1994.
- [Gut84] A. Guttmann. R-trees: A dynamic index structure for spatial searching. In *Proc. ACM SIGMOD Conf. Principles Database Syst.*, pages 47–57, 1984.
- [HW87] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.
- [HS95] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.
- [HAMS97] C.-T. Ho, R. Agrawal, N. Megiddo, and R. Srikant. Range queries in OLAP data cubes. In *Proc. ACM SIGMOD Conf. Management Data*, pages 73–88, 1997.
- [HBA97] C.-T. Ho, J. Bruck, and R. Agrawal. Partial-sum queries in OLAP data cubes using covering codes. In *Proc. Annu. ACM Sympos. Principles Database Syst.*, pages 228–237, 1997.
- [Kol01] V. Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. In *Proc. 42nd Sympos. Found. Comput. Sci.*, pages 56–65, 2001.
- [LS90] D.B. Lomet and B. Salzberg. The hB-tree: A multiattribute indexing method with good guaranteed performance. *ACM Trans. Database Syst.*, 15:625–658, 1990.
- [MTT99] Y. Manolopoulos, Y. Theodoridis, and V.J. Tsotras. *Advanced Database Indexing*. Kluwer Academic Publishers, Boston, 1999.
- [Mat92] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.
- [Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.
- [Mat94] J. Matoušek. Geometric range searching. *ACM Comput. Surv.*, 26:421–461, 1994.
- [McC85] E.M. McCreight. Priority search trees. *SIAM J. Comput.*, 14:257–276, 1985.
- [Meh84] K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*, volume 3 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Heidelberg, 1984.
- [Mor03] C.W. Mortensen. Fully-dynamic two dimensional orthogonal range and line segment intersection reporting in logarithmic time. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, 2003.
- [Mul93] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1993.
- [NHS84] J. Nievergelt, H. Hinterberger, and K.C. Sevcik. The grid file: An adaptable, symmetric multi-key file structure. *ACM Trans. Database Syst.*, 9:38–71, 1984.
- [NW00] J. Nievergelt and P. Widmayer. Spatial data structures: Concepts and design choices. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 725–764. Elsevier North-Holland, Amsterdam, 2000.
- [Ove83] M.H. Overmars. *The Design of Dynamic Data Structures*, volume 156 of *Lecture Notes Comput. Sci.* Springer-Verlag, Heidelberg, 1983.
- [PAHP02] C.M. Procopiuc, P.K. Agarwal, and S. Har-Peled. Star-tree: An efficient self-adjusting index for moving points. In *Proc. 4th Workshop Algorithm Eng. Experiments*, pages 178–193, 2002.
- [Ram00] E.A. Ramos. Linear programming queries revisited. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 176–181, 2000.

- [Rob81] J.T. Robinson. The k -d-b-tree: a search structure for large multidimensional dynamic indexes. In *Proc. ACM SIGMOD Conf. Management Data*, pages 10–18, 1981.
- [ST99] B. Salzberg and V.J. Tsotras. A comparison of access methods for time evolving data. *ACM Comput. Surv.*, 31:158–221, 1999.
- [Sam90] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, 1990.
- [SRF87] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+ -tree: A dynamic index for multi-dimensional objects. In *Proc. 13th VLDB Conf.*, pages 507–517, 1987.
- [JLL00] S. Šaltenis, C.S. Jensen, S.T. Leutenegger, and M.A. Lopez. Indexing the positions of continuously moving objects. In *Proc. ACM SIGMOD Internat. Conf. Management Data*, pages 331–342, 2000.
- [Vui80] J. Vuillemin. A unifying look at data structures. *Commun. ACM*, 23:229–239, 1980.
- [Wil89] D.E. Willard. Lower bounds for the addition-subtraction operations in orthogonal range queries and related problems. *Inform. Comput.*, 82:45–64, 1989.
- [Wil82] D.E. Willard. Polygon retrieval. *SIAM J. Comput.*, 11:149–165, 1982.
- [Yao82] A.C. Yao. Space-time trade-off for answering range queries. In *Proc. 14th Annu. ACM Sympos. Theory Comput.*, pages 128–136, 1982.
- [Yao85] A.C. Yao. On the complexity of maintaining partial sums. *SIAM J. Comput.*, 14:277–288, 1985.

37 RAY SHOOTING AND LINES IN SPACE

Marco Pellegrini

INTRODUCTION

The geometry of lines in 3-space has been a part of the body of classical algebraic geometry since the pioneering work of Plücker. Interest in this branch of geometry has been revived in recent years by several converging trends in computer science. The discipline of computer graphics ([Chapter 49](#)) has pursued the task of rendering realistic images by simulating the flow of light within a scene according to the laws of elementary optical physics. In these models light moves along straight lines in 3-space and a computational challenge is to find efficiently the intersections of a very large number of rays with the objects comprising the scene. In robotics ([Chapters 47](#) and [48](#)) the chief problem is that of moving 3D objects without collisions. Effects due to the edges of objects have been studied as a special case of the more general problem of representing and manipulating lines in 3-space. Computational geometry (whose core is better termed “design and analysis of geometric algorithms”) has moved recently from the realm of planar problems to tackling directly problems that are specifically 3D. The new and sometimes unexpected computational phenomena generated by lines (and segments) in 3-space have emerged as a main focus of research.

In this chapter we will survey the present state of the art on lines and ray shooting in 3-space from the point of view of computational geometry. The emphasis is on provable nontrivial bounds for the time and storage used by algorithms for solving natural problems on lines, rays, and polyhedra in 3-space. We start by mentioning different possible choices of coordinates for lines (Section 37.1). This is an essential initial step because different coordinates highlight different properties of the lines in their interaction with other geometric objects. Here a special role is played by *Plücker coordinates* [Plu65] (Section 37.1), which represent the starting point for many of the most recent results. Then we consider how lines interact with each other (Section 37.2). We are given a finite set of lines L that act as obstacles and we will define other (infinite) sets of lines induced by L that capture some of the important properties of visibility and motion problems. We show bounds on the storage required for a complete description of such sets. Then we move a step forward by considering the same sets of lines when the obstacles are polyhedral sets, more commonly encountered in applications. We arrive in Section 37.3 at the ray-shooting problem and its variants (on-line, off-line, arbitrary direction, fixed direction, and shooting with objects other than rays). Again, the obstacles are usually polyhedral objects, but in one case we are able to report a ray-shooting result on spheres.

Section 37.4 is devoted to the problem of collision-free movements (arbitrary or translation only) of lines among obstacles. This problem arises, for example, when lines are used to model radiation or light beams (e.g., lasers). In Section 37.5 we define a few notions of distance among lines, and as a consequence we have several

natural proximity problems for lines in 3-space. Finding the closest pair in a set of lines is the most basic of such problems.

In Section 37.6 we survey what is known about the “dominance” relation among lines. This relation is central for many visibility problems in graphics. It is used, for example, in the painter’s algorithm for hidden surface removal ([Chapter 49](#)). Another direction of research has explored the relation between lines in 3-space and their orthogonal projections. A central topic here is realizability: Given a set of planar lines together with a relation, does there exist a corresponding set of lines in 3-space whose dominance is consistent with the given relation?

37.1 COORDINATES OF LINES

GLOSSARY

Homogeneous coordinates: A point (x, y, z) in Cartesian coordinates has homogeneous coordinates (x_0, x_1, x_2, x_3) , where $x = x_1/x_0$, $y = x_2/x_0$, and $z = x_3/x_0$.

Oriented lines: A line may have two distinct orientations. A line and an orientation form an oriented line.

Unoriented line: A line for which an orientation is not distinguished.

(I) Canonical coordinates by pairs of planes. The intersection of two planes with equations $y = az + b$ and $x = cz + d$ is a nonhorizontal line in 3-space, uniquely defined by the four parameters (a, b, c, d) . Thus these parameters can be taken as coordinates of such lines. In fact, the space of nonhorizontal lines is homeomorphic to \mathbb{R}^4 . Results on ray shooting among boxes and some lower bounds on stabbing are obtained using these coordinates.

(II) Canonical coordinates by pairs of points. Given two parallel horizontal planes, $z = 1$ and $z = 0$, the intersection points of a nonhorizontal line l with the two planes uniquely define that line. If $(x_0, y_0, 0)$ and $(x_1, y_1, 1)$ are two such points for l , then the quadruple (x_0, y_0, x_1, y_1) can be used as coordinates of l . Results on sets of horizontal polygons are obtained using these coordinates.

Although four is the minimum number of coordinates needed to represent an *unoriented* line, such parametrizations have proved useful only in special cases. Many interesting results have been derived using instead a five-dimensional parametrization for *oriented* lines, called *Plücker coordinates*.

(III) Plücker coordinates of lines. An oriented line in 3-space can be given by the homogeneous coordinates of two of its points. Let l be a line in 3-space and let $a = (a_0, a_1, a_2, a_3)$ and $b = (b_0, b_1, b_2, b_3)$ be two distinct points in homogeneous coordinates on l . We can represent the line l , oriented from a to b , by the matrix

$$l = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \end{pmatrix}, \quad \text{with } a_0, b_0 > 0.$$

By taking the determinants of the six 2×2 submatrices of the above 2×4 matrix

we obtain the ***homogeneous Plücker coordinates*** of the line:

$$p(l) = (\xi_{01}, \xi_{02}, \xi_{03}, \xi_{12}, \xi_{31}, \xi_{23}), \quad \text{with } \xi_{ij} = \det \begin{pmatrix} a_i & a_j \\ b_i & b_j \end{pmatrix}.$$

The six numbers ξ_{ij} are interpreted as homogeneous coordinates of a point in 5-space. For a given line l the six numbers are unique modulo a positive multiplicative factor, and they do not depend on the particular distinct points a and b that we have chosen on l . We call $p(l)$ the ***Plücker point*** of l in projective 5-dimensional space \mathbb{P}^5 .

We also define the ***Plücker hyperplane*** of the line l to be the hyperplane in \mathbb{P}^5 with vector of coefficients $v(l) = (\xi_{23}, \xi_{31}, \xi_{12}, \xi_{03}, \xi_{02}, \xi_{01})$. So the Plücker hyperplane is:

$$h(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p = 0\}.$$

For each Plücker hyperplane we have a positive and a negative halfspace given by $h^+(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p \geq 0\}$ and $h^-(l) = \{p \in \mathbb{P}^5 \mid v(l) \cdot p \leq 0\}$. Not every tuple of 6 real numbers corresponds to a line in 3-space since the Plücker coordinates must satisfy the condition

$$\xi_{01}\xi_{23} + \xi_{02}\xi_{31} + \xi_{03}\xi_{12} = 0. \quad (37.1.1)$$

The set of points in \mathbb{P}^5 satisfying Equation 37.1.1 forms the so-called ***Plücker hypersurface*** Π ; it is also called the ***Klein quadric*** or the ***Grassmannian*** (manifold). The converse is also true: every tuple of six real numbers satisfying Equation 37.1.1 is the Plücker point of some line in 3-space. Given two lines l and l' , they intersect or are parallel (i.e., they intersect at infinity) when the four defining points are coplanar. In this case the determinant of the 4×4 matrix formed by the 16 homogeneous coordinates of the four points is zero. In terms of Plücker coordinates we have the following basic lemmas.

LEMMA 37.1.1

Lines l and l' intersect or are parallel (meet at infinity) if and only if $p(l) \in h(l')$.

Note that Equation 37.1.1 states in terms of Plücker coordinates the fact that any line always meets itself.

LEMMA 37.1.2

Let l be an oriented line and t a triangle in Cartesian 3-space with vertices (p_0, p_1, p_2) . Let l_i be the oriented line through (p_i, p_{i+1}) (indices mod 3). Then l intersects t if and only if either $p(l) \in h^+(l_0) \cap h^+(l_1) \cap h^+(l_2)$ or $p(l) \in h^-(l_0) \cap h^-(l_1) \cap h^-(l_2)$.

These two lemmas allow us to map combinatorial and algorithmic problems involving lines (and polyhedral sets) in 3-space into problems involving sets of hyperplanes and points in projective 5-space (Plücker space). The main advantage is that we can use the rich collection of results on the combinatorics of high dimensional arrangements of hyperplanes (see [Chapter 24](#)). The main drawback is that we are using five (nonhomogeneous) parameters, instead of four which is the minimum number necessary. This choice has a potential for increasing the time bounds of line algorithms. We are rescued by the following theorem:

THEOREM 37.1.3 [APS93]

Given a set H of n hyperplanes in 5-dimensional space, the complexity of the cells of the arrangement $\mathcal{A}(H)$ intersected by the Plücker hypersurface Π (also called the **zone** of Π in $\mathcal{A}(H)$) is $O(n^4 \log n)$.

Although the entire arrangement $\mathcal{A}(H)$ can be of complexity $\Theta(n^5)$, if we are working only with Plücker points we can limit our constructions to the zone of Π , the complexity of which is one order of magnitude smaller. Theorem 37.1.3 is especially useful for deriving ray-shooting results.

The list of coordinatizations discussed in this section is by no means exhaustive. Other parametrizations are used, for example, in [Ame92], [AAS97], and [AS96].

A TYPICAL EXAMPLE

A typical example of the use of Plücker coordinates in 3D problems is the result for fast ray shooting among polyhedra (see [Table 37.3.1](#)). We triangulate the faces of the polyhedra and extend each edge to a full line. Each such line is mapped to a Plücker hyperplane. Lemma 37.1.2 guarantees that each cell in the resulting arrangement of Plücker hyperplanes contains Plücker points that pass through the same set of triangles. Thus to answer a ray-shooting query, we first locate the query Plücker point in the arrangement, and then search the list of triangles associated with the retrieved cell. This final step is accomplished using a binary search strategy when the polyhedra are disjoint. Theorem 37.1.3 guarantees that we need to build a point location structure only for the zone of the Plücker hypersurface, thus saving an order of magnitude over general point location methods for arrangements (see [Sections 21.3](#) and [30.7](#)).

37.2 SETS OF LINES IN 3-SPACE

With Plücker coordinates (III) to represent oriented lines, we can use the topology induced by the standard topology of 5-dimensional projective space \mathbb{P}^5 on Π as a natural topology on sets of oriented lines. Using the four-dimensional coordinatizations (I) or (II), we can impose the standard topology of \mathbb{R}^4 on the set of nonhorizontal unoriented lines. Thus we can define the concepts of “neighbourhood,” “continuous path,” “open set,” “closed set,” “boundary,” “path-connected component,” and so on, for the set \mathcal{L} of lines in 3-space. The distinction between oriented lines and unoriented lines is mainly technical and the complexity bounds hold in either case.

GROUPS OF LINES INDUCED BY A FINITE SET OF LINES

GLOSSARY

Semialgebraic set: The set of all points that satisfy a Boolean combination of a finite number of algebraic constraints (equalities and inequalities) in the Cartesian coordinates of \mathbb{R}^d . See [Chapter 33](#).

Path-connected component: A maximal set of lines that can be connected by a path of lines, a continuous function from the interval $[0, 1]$ to the space of lines.

Positively-oriented lines: Oriented lines l'_1 and l'_2 on the xy -plane are positively-oriented if the triple scalar product of vectors parallel to l'_1 , l'_2 , and the positive z -axis is positive.

Consistently-oriented lines: An oriented line l in 3-space is oriented consistently with a 3D set L of oriented lines if the projection l' of l onto a plane is positively-oriented with the projection of every line in L .

A finite set L of n lines in 3-space can be viewed as an obstacle to the free movement of other lines in 3-space. Many applications lead to defining groups of lines with some special properties with respect to the fixed lines L . The resources used by algorithms for these applications are often bounded by the “complexity” of such groups.

The boundary of a semialgebraic set in \mathbb{R}^4 is partitioned into a finite number of faces of dimension 0, 1, 2, and 3, each of which is also a semialgebraic set. The number of faces on the boundary of a semialgebraic set is the **complexity** of that set. The groups of lines that we consider are represented in \mathbb{R}^4 by semialgebraic sets, with the coefficients of the corresponding algebraic constraints a function of the given finite set of lines L .

The set $\text{Miss}(L)$ consists of lines that do not meet any line in L . The sets $\text{Vert}(L)$ and $\text{Free}(L)$ consists of lines that may be translated to infinity without collision with lines in L . The basic complexities displayed in Table 37.2.1 are derived from [CEGS96, Pel94b, Aga94].

TABLE 37.2.1 Complexity of groups of lines defined by lines.

SET OF LINES	DEFINITION	COMPLEXITY
$\text{Miss}(L)$	do not meet any line in L	$\Theta(n^4)$
1 component of $\text{Miss}(L)$	1 path-connected component	$\Theta(n^2)$
$\text{Vert}(L)$	can be translated vertically to ∞	$\Theta(n^3)$
$\text{Free}(L)$	can be translated to ∞ in some direction above L and oriented consistently with L	$\Omega(n^3), O(n^3 \log n)$
$\text{VertCO}(L)$		$\Theta(n^2)$

MEMBERSHIP TESTS

Given L , we can build a data structure during a preprocessing phase so that when presented with a new (query) line l , we can decide efficiently whether l is in one of the sets defined in the previous section. Such an algorithm implements a membership test for a group of lines. Table 37.2.2 shows the main results.

GROUPS OF LINES INDUCED BY POLYHEDRA

GLOSSARY

ϵ : A positive real number, which we may choose arbitrarily close to zero for each

TABLE 37.2.2 Membership tests for groups of lines defined by lines.

SET OF LINES	QUERY TIME	PREPROC/STORAGE	SOURCE
Miss(L)	$O(\log n)$	$O(n^{4+\epsilon})$	[Pel93b, AM93]
1 component of Miss(L)	$O(\log n)$	$O(n^{2+\epsilon})$	[Pel91]
Vert(L), VertCO(L)	$O(\log n)$	$O(n^{2+\epsilon})$	[CEGS96]
Free(L)	$O(\log n)$	$O(n^{3+\epsilon})$	[Pel94b]

algorithm or data structure. A caveat is that the multiplicative constant implicit in the big- O notation depends on ϵ and its value increases when ϵ tends to zero.

$\alpha(\cdot)$: The inverse of Ackermann's function. $\alpha(n)$ grows very slowly and is at most 4 for any practical value of n . See [Section 47.4](#).

$\beta(\cdot)$: $\beta(n) = 2^{c\sqrt{\log n}}$ for a constant c . $\beta(\cdot)$ is a function that is smaller than any polynomial but larger than any polylogarithmic factor. Formally we have that for every $a, b > 0$, $\log^a n \leq \beta(n) \leq n^b$ for any $n \geq n_0(a, b)$.

Polyhedral set P : A region of 3-space bounded by a collection of interior-disjoint vertices, segments, and planar polygons. We denote with n the total number of vertices, edges, and faces.

Star-shaped polyhedron: A polyhedron P for which there exists a point $o \in P$ such that for every point $p \in P$, the open segment op is contained in P .

Terrain: When the star-shaped polyhedron is unbounded and o is at infinity we obtain a terrain, a monotone surface (cf. [Section 26.1](#)).

A collection of polyhedra in 3-space may act as obstacles limiting the collision-free movements of lines. Following the blueprint of the previous section, the complexity of some interesting groups of lines induced by polyhedra are displayed in Table 37.2.3 (see [HS94, Pel94b, Aga94]).

TABLE 37.2.3 Complexity of groups of lines defined by polyhedra.

SET OF LINES	DEFINITION	COMPLEXITY
Miss(P)	do not meet polyhedron P	$\Theta(n^4)$
Vert(P)	can be translated vertically to ∞	$\Omega(n^3)$, $O(n^3\beta(n))$
Free(P)	can be translated to ∞ in some direction	$\Omega(n^3)$
Miss(Q), Free(Q)	Q star-shaped polyhedron or a terrain	$\Omega(n^2\alpha(n))$, $O(n^3 \log n)$

Similarly, we can define groups of *3D segments* defined by polyhedra in 3D. The set of relatively open segments that miss P is also a semialgebraic set, known as the **3D Visibility skeleton** (see [DDP97, Dur99]). Its combinatorial complexity is $\Theta(n^4)$.

OPEN PROBLEMS

1. Find an almost cubic upper bound on the complexity of the group of lines $\text{Free}(P)$ for a polyhedron P .
 2. Close the gap between the quadratic lower and the cubic upper bound for the group $\text{Free}(T)$ induced by a terrain T ([Table 37.2.3](#)).
-

SETS OF STABBING LINES

GLOSSARY

Stabber: A line l that intersects every member of a collection $\mathcal{P} = \{P_1, \dots, P_k\}$ of polyhedral sets. The sum of the sizes of the polyhedral sets in \mathcal{P} is n . The set of lines stabbing \mathcal{P} is denoted $S(\mathcal{P})$.

Box: A parallelepiped each of whose faces is orthogonal to one of the three Cartesian axes.

c -oriented: Polyhedra whose face normals come from a set of c fixed directions.

Table 37.2.4 lists the worst-case complexity of the set $S(\mathcal{P})$ and the time to find a witness stabbing line.

TABLE 37.2.4 Complexity of the set of stabbing lines and detection time.

OBJECTS	COMPLEXITY OF $S(\mathcal{P})$	FIND TIME	SOURCES
Convex polyhedra	$\Omega(n^3), O(n^3 \log n)$	$O(n^3 \beta(n))$	[PS92, Pel93a, Aga94]
Boxes	$O(n^2)$	$O(n)$	[Ame92, Meg91]
c -oriented polyhedra	$O(n^2)$	$O(n^2)$	[Pel91]
Horiz polygons	$\Theta(n^2)$	$O(n)$	[Pel91]

Note that in some cases (boxes, parallel polygons) a stabbing line can be found in linear time, even though the best bound known for the complexity of the stabbing set is quadratic. These results are obtained using linear programming techniques ([Chapter 45](#)).

We can determine whether a given line l is a stabber for a preprocessed set \mathcal{P} of convex polyhedra in time $O(\log n)$, using data structures of size $O(n^{2+\epsilon})$ that can be constructed in time $O(n^{2+\epsilon})$ [PS92].

For an *oriented* stabber l and a set \mathcal{O} of k disjoint convex bodies in R^d , the order of the intersection of the objects along l is called a *geometric permutation* (cf. [Chapter 4](#)). A recent advance of Zhou and Suri [ZS01] shows that for balls of unit radius and k large enough there are at most 4 geometric permutations. For k rectangular boxes there are at most 2^{d-1} geometric permutations, which is tight (see also [OR01]).

OPEN PROBLEMS

1. Can linear programming techniques yield a linear-time algorithm for c -oriented polyhedra?
 2. The lower bound for $S(\mathcal{P})$ for a set of pairwise *disjoint* convex polyhedra is only $\Omega(n^2)$ [PS92]. Close the gap between this and the cubic upper bound.
-

37.3 RAY SHOOTING

Ray shooting is an important operation in computer graphics and a primitive operation useful in several geometric computations (e.g., hidden surface removal, and detecting and computing intersections of polyhedra). The problem is defined as follows. Given a large collection \mathcal{P} of simple polyhedral objects, we want to know, for a given point p and direction \vec{d} , the first object in \mathcal{P} intersected by the ray defined by the pair p, \vec{d} . A single polyhedron with many faces can be represented without loss of generality by the collection of its faces, each treated as a separate polygon.

ON-LINE RAY SHOOTING IN AN ARBITRARY DIRECTION

Here we consider the on-line model in which the set \mathcal{P} is given in advance and a data structure is produced and stored. Afterward we are given the query rays one-by-one and the answer to one query must be produced before the next query is asked.

[Table 37.3.1](#) summarizes the known complexity bounds on this problem. For a given class of objects we report the query time, the storage, and the preprocessing time of the method with the best bound. In this table and in the following ones we omit the big- O symbols. Again, n denotes the sum of the sizes of all the polyhedra in \mathcal{P} . The main references on ray shooting (Table 37.3.1) are in [Pel93b, dBH⁺94] (boxes), [AM93, AM94, Pel93b, dBH⁺94, AS93b] (polyhedra), [Pel96] (horizontal polygons), [AAS97, MS97] (spheres), and [DK85, AS96a] (convex polyhedra).

GLOSSARY

Fat horizontal polygons: Convex polygons contained in planes parallel to the xy -plane, with a lower bound on the size of their minimum interior angle.

Curtains: Polygons in 3-space bounded by one segment and by two vertical rays from the endpoints of the segment.

Axis-oriented curtains: Curtains hanging from a segment parallel to the x - or y -axis.

When we drop the fatness assumption for horizontal polygons we obtain bounds that depend on K , the complexity of the set of lines missing the *edges* of the polygons. K is in the range $[n^2, \dots, n^4]$ and reaches the upper end of the range only when the polygons are very long and thin.

TABLE 37.3.1 On-line ray shooting in an arbitrary direction.

OBJECTS	QUERY	STORAGE	PREPROCESSING
Boxes, terrains, curtains	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Boxes	$n^{1+\epsilon}/m^{1/2}$	$n \leq m \leq n^2$	$m^{1+\epsilon}$
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$
Polyhedra	$n^{1+\epsilon}/m^{1/4}$	$n \leq m \leq n^4$	$m^{1+\epsilon}$
Fat horiz polygons	$\log n$	$n^{2+\epsilon}$	$n^{2+\epsilon}$
Horiz polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$
Spheres	$\log^4 n$	$n^{3+\epsilon}$	$n^{3+\epsilon}$
1 convex polyhedron	$\log n$	n	$n \log n$
s convex polyhedra	$\log^2 n$	$n^{2+\epsilon}s^2$	$n^{2+\epsilon}s^2$

Most of the data structures for ray shooting mentioned in Table 37.3.1 can be made dynamic (under insertion and deletion of objects in the scene) by using general dynamization techniques (see [Meh84]) and other more recent results [AEM92].

ON-LINE RAY SHOOTING IN A FIXED DIRECTION

We can usually improve on the general case if the direction of the rays is fixed a priori, while the source of the ray can lie anywhere in \mathbb{R}^3 . See Table 37.3.2; here k is the number of vertices, edges, faces, and cells of the arrangement of the (possibly intersecting) polyhedra. References for ray shooting in a fixed direction (Table 37.3.2) are [dB93, dBGH94].

TABLE 37.3.2 On-line ray shooting in a fixed direction.

OBJECTS	QUERY TIME	STORAGE	PREPROCESSING
Boxes	$\log n$	$n^{1+\epsilon}$	$n^{1+\epsilon}$
Boxes	$\log n(\log \log n)^2$	$n \log n$	$n \log^2 n$
Axis-oriented curtains	$\log n$	$n \log n$	$n \log n$
Polyhedra	$\log^2 n$	$n^{2+\epsilon} + k$	$n^{2+\epsilon} + k \log n$
Polyhedra	$n^{1+\epsilon}/m^{1/3}$	$n \leq m \leq n^3$	$m^{1+\epsilon}$

OFF-LINE RAY SHOOTING IN AN ARBITRARY DIRECTION

In the previous section we considered the on-line situation when the answer to the query must be generated before the next question is asked. In many situations we do not need such strict requirements. For example, we might know all the queries from the start and are interested in minimizing the total time needed to answer all of the queries (the *off-line* situation). In this case there are simpler algorithms that improve on the storage bounds of on-line algorithms:

THEOREM 37.3.1

Given a polyhedral set \mathcal{P} with n vertices, edges, and faces, and given m rays off-

line, we can answer the m ray-shooting queries in time $O(m^{0.8}n^{0.8+\epsilon} + m \log^2 n + \log n \log m)$ using $O(n+m)$ storage.

One of the most interesting applications of this result is the current asymptotically fastest algorithm for detecting whether two nonconvex polyhedra in 3-space intersect, and to compute their intersection. See Table 37.3.3; here k is the size of the intersection.

TABLE 37.3.3 Detection and computation of intersection among polyhedra.

OBJECTS	DETECTION	COMPUTATION	SOURCES
Polyhedra	$n^{1.6+\epsilon}$	$n^{1.6+\epsilon} + k \log^2 n$	[Pel93b]
Terrains	$n^{4/3+\epsilon}$	$n^{4/3+\epsilon} + k^{1/3}n^{1+\epsilon} + k \log^2 n$	[CEGS94, Pel94b]

Lower bounds on off-line ray-shooting and intersection problems in 3D are difficult to prove. It has been shown in [Eri95] that many such problems are at least as hard as Hopcroft's incidence problem (in the appropriate ambient space).

RAY-SHOOTING IN SIMPLICIAL COMPLEXES

If we have a subdivision of the free space $\mathbb{R}^3 \setminus \mathcal{P}$ into a simplicial complex we can answer ray-shooting queries by locating the tetrahedron containing the source of the ray and tracing the ray in the complex at cost $O(1)$ for each visited face of the complex. There are scenes \mathcal{P} for which any simplicial complex has some line meeting $\Omega(n)$ faces of the complex. The average time for tracing a ray in a simplicial complex is proportional to the sum of the areas of all faces in the complex. It is possible to find a complex of total surface within a constant multiplicative factor of the minimum, with $O(n^3 \log n)$ simplices in time $O(n^3 \log n)$ for general \mathcal{P} . For \mathcal{P} a point set or a single polyhedron $O(n^2 \log n)$ time suffices (see [AAS95, AF99, CD99]). These results are obtained via a generalization of Eppstein's method for two-dimensional Minimum Weighted Steiner Triangulation (2D-MWST) of a point set [Epp94]. In the 3D context the weight is the surface of the 2D faces of the complex. Starting from the set \mathcal{P} of polyhedral obstacles in \mathbb{R}^3 , an oct-tree-based decomposition of \mathbb{R}^3 is produced which is "balanced" and "smooth." It is then proved, via a charging argument, that the sum of the surfaces of all the boxes in the decomposition is within a constant factor of the surface of any Minimum Surface Steiner Simplicial Complex compatible with \mathcal{P} . From the oct-tree the final complex is derived within just a constant factor increase in the total surface.

EXTENSIONS AND ALTERNATIVE METHODS

Some ray-shooting results of Agarwal and Matoušek are obtained from the observation that a ray is traced by a family of segments $\rho(t)$, where one endpoint is the ray source and the second endpoint lies on the ray at distance t from the source. Using *parametric search* techniques (Chapter 43), Agarwal and Matoušek compute the first value of t for which $\rho(t)$ intersects \mathcal{P} , and thus answer the ray-shooting query.

An interesting extension of the concept of shooting rays against obstacles is obtained by shooting triangles and more generally simplices. We consider a family of simplices $s(t)$, indexed by real parameter $t \in \mathbb{R}^+$, where t is the volume of the simplex $s(t)$, such that the simplices form a chain of inclusions: $t_1 \leq t_2 \Rightarrow s(t_1) \subset s(t_2)$. Intuitively we grow a simplex until it first meets one of the obstacles. Surprisingly, when the obstacles are general polyhedra, shooting simplices is not harder than shooting rays.

THEOREM 37.3.2 [Pel94a]

Given a set of polyhedra \mathcal{P} with n edges we can preprocess it in time $O(m^{1+\epsilon})$ into a data structure of size m , such that the following queries can be answered in time $O(n^{1+\epsilon}/m^{1/4})$: Given a simplex s , does s avoid \mathcal{P} ? Given a family of simplices $s(t)$ as above, which is the first value of t for which $s(t)$ intersects \mathcal{P} ?

Computing the interaction between beams and polyhedral objects is a central problem in radio-therapy and radio-surgery (see e.g. [SAL93] [For99] [CHX00]).

Other popular methods for solving ray-shooting problems are based binary space partitions, kD-trees, solid modeling schemes, etc. These methods, although important in practice, are usually not fully analyzable a priori using algorithmic analysis. In [AB⁺02] Aronov et al. propose techniques that give a posteriori estimates of the cost of ray shooting.

OPEN PROBLEMS

1. Find time and storage bounds for ray-shooting general polyhedra that are sensitive to the actual complexity of a group of lines (as opposed to the worst case bound on such a complexity).
2. For a collection of s convex polyhedra there is a wide gap in storage and preprocessing between the special case $s = 1$ and the case for general s . It would be interesting to obtain a bound that depends smoothly on s .
3. No lower bound on time or storage required for ray shooting is known.

37.4 MOVING LINES AMONG OBSTACLES

ARBITRARY MOTIONS

So far we have treated lines as static objects. In this section we consider moving lines. A laser beam in manufacturing or a radiation beam in radiation therapy can be modeled as lines in 3-space moving among obstacles. The main computational problem is to decide whether a source line l_1 can be moved continuously until it coincides with a target line l_2 so that it avoids a set of obstacles \mathcal{P} . We consider the following situation where the set of obstacles \mathcal{P} is given in advance and preprocessed to obtain a data structure. When the query lines l_1 and l_2 are given the answer is produced before a new query is accepted. We have the results shown in [Table 37.4.1](#), where K is the complexity of the set of lines missing the edges of the polygons (cf. Section 37.2).

TABLE 37.4.1 On-line collision-free movement of lines among obstacles.

OBJECTS	QUERY TIME	STORAGE	PREPROC	SOURCES
Polyhedra	$\log n$	$n^{4+\epsilon}$	$n^{4+\epsilon}$	[Pel93b]
Horiz polygons	$\log^3 n$	$n^{3+\epsilon} + K$	$n^{3+\epsilon} + K \log n$	[Pel96]

OPEN PROBLEMS

It is not known how to trade off storage and query time, or whether better bounds can be obtained in an off-line situation.

TRANSLATIONS

We now restrict the type of motion and consider only translations of lines. The first result is negative: there are sets of lines which cannot be split by any collision-free translation. There exists a set L of 9 lines such that, for all directions v and all subsets $L_1 \subset L$, L_1 cannot be translated continuously in direction v without collisions with $L \setminus L_1$ [SS93]. Positive results are displayed in Table 37.4.2.

GLOSSARY

Towering property: Two sets of lines L_1 and L_2 are said to satisfy the towering property if we can translate simultaneously all lines in L_1 in the vertical direction without any collision with any lines in L_2 .

Separation property: Two sets of lines satisfy the separation property if they satisfy the towering property in some direction (not necessarily vertical).

TABLE 37.4.2 Separating lines by translations.

PROPERTY	TIME TO CHECK PROPERTY	SOURCES
Towering	$O(n^{4/3+\epsilon})$	[CEGS96]
Separation	$O(n^{3/2+\epsilon})$	[Pel94b]

37.5 CLOSEST PAIR OF LINES

GLOSSARY

Distance between lines: The Euclidean distance between two lines l_1 and l_2 in 3-space is the length of the shortest segment with one endpoint on l_1 and the other on l_2 .

Vertical distance between lines: The length of the vertical segment with one endpoint on l_1 and one endpoint of l_2 (provided a unique such segment exists).

Vertical distance between segments: The length of the vertical segment with one endpoint in s_1 and one in s_2 . If a unique such vertical segment does not exist the vertical distance is undefined.

TABLE 37.5.1 Closest and farthest pair of lines and segments.

PROBLEM	OBJECTS	TIME	SOURCES
Smallest distance	lines	$O(n^{8/5+\epsilon})$	[CEGS93]
Smallest vertical distance	lines, segments	$O(n^{8/5+\epsilon})$	[Pel94a]
Largest vertical distance	lines, segments	$O(n^{4/3+\epsilon})$	[Pel94a]

Any centrally symmetric convex polyhedron C in 3D defines a metric L_C . If C has constant combinatorial complexity, then the complexity of the Voronoi diagram of n lines in 3-space is $O(n^2\alpha(n) \log n)$ [CKS⁺98]. For Euclidean distance the best bound is $O(n^{3+\epsilon})$.

OPEN PROBLEM

1. Finding an algorithm with subquadratic time complexity for the smallest distance among segments (and more generally, among polyhedra) is a notable open question.
2. Close the gap between the complexity of Voronoi diagrams of lines induced by polyhedral metrics and the Euclidean metric.

37.6 DOMINANCE RELATION AND WEAVINGS

GLOSSARY

Dominance relation: Given a finite set L of nonvertical disjoint lines in \mathbb{R}^3 , define a dominance relation \prec among lines in L as follows: $l_1 \prec l_2$ if l_2 lies above l_1 , i.e., if, on the vertical line intersecting l_1 and l_2 , the intersection with l_1 has a smaller z -coordinate than does the intersection with l_2 .

Weaving: A weaving is a pair (L', \prec') where L' is a set of lines on the plane and \prec' is an anti-symmetric nonreflexive binary relation $\prec' \subset L' \times L'$ among the lines in L' .

Realizable: A weaving is realizable if there exists a set of lines L in 3-space such that L' is the projection of L and \prec' is the image of the dominance relation \prec for L .

Elementary cycle: A cycle in the dominance relation such that the projections of the lines in such a cycle bound a cell of the arrangement of projected lines.

Perfect: A weaving (L', \prec') is perfect if each line l alternates below and above the other lines in the order they cross l (see Figure 37.6.1a).

Bipartite weaving: Two families of segments in 3-space such that, when projecting on the xy -plane, each segment does not meet segments from its own family and meets all the segments from the other family. (A bipartite weaving of size 4×4 is shown in Figure 37.6.1b.)

Perfect bipartite weaving: Every segment alternates above and below the segments of the other family (see Figure 37.6.1b).

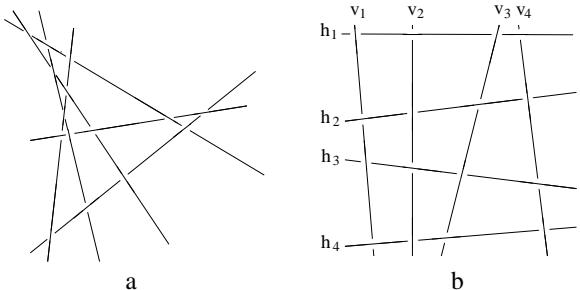


FIGURE 37.6.1

- (a) *A perfect weaving;*
- (b) *a perfect bipartite weaving.*

The dominance relation is possibly cyclic, that is, there may be three lines such that $l_1 \prec l_2 \prec l_3 \prec l_1$. Some results in [CEG⁺92, PPW93, dBOS94, Sol98] related to dominance are the following:

1. *How fast can we generate a consistent linear extension if the relation \prec is acyclic?* $O(n^{4/3+\epsilon})$ time is sufficient for the case of lines. This result has been extended to the case of segments and polyhedra. If an ordering is given as input, it is possible to *verify* that it is a linear extension of \prec in time $O(n^{4/3+\epsilon})$.
2. *How many elementary cycles in the dominance relation can n lines define?* In the case of bipartite weavings, the dominance relation can have $O(n^{3/2})$ elementary cycles and there is a family of bipartite weavings attaining the lower bound $\Omega(n^{4/3})$. For general weavings there is a construction attaining $\Omega(n^{3/2})$.
3. *If we cut the segments to eliminate cycles, how many ‘cuts’ are necessary to eliminate all cycles?* From the previous result we have that sometimes $\Omega(n^{4/3})$ cuts are necessary since a single cut can eliminate only one elementary cycle. In order to eliminates all cycles (including the nonelementary ones) in a bipartite weaving, $O(n^{9/5})$ cuts are always sufficient.
4. *How fast can we find those cuts?* There is an algorithm to find cuts in bipartite weavings in time $O(n^{9/5} \log n)$. In a general weaving, calling μ is the minimum number of cuts, there is an algorithm to cut all cycles in time $O(n^{4/3+\epsilon} \mu^{1/3})$ that produces $O(n^{1+\epsilon} \mu^{1/3})$ cuts.

-
5. The fraction of realizable weavings over all possible weavings of n lines tends to 0 exponentially as n tends to ∞ .
 6. A perfect weaving of $n \geq 4$ lines is not realizable.
 7. Perfect bipartite weavings are realizable only if one of the families has fewer than four segments.
-

37.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Books and Surveys.

[Som51, HP52, Jes03]: Extensive book-length treatments of the geometry of lines in space.

[Sto89, Sto91]: Algorithmic aspects of computing in projective spaces.

[BR79, Shi78]: Uses of the geometry of lines in robotics. For uses in graphics see [FVFH90].

[dB93]: A detailed description of many ray-shooting results.

[Spe92, Dur99, Hav01]: Pointers to the vast related literature on pragmatic aspects of ray shooting.

RELATED CHAPTERS

- [Chapter 24: Arrangements](#)
 - [Chapter 34: Point location](#)
 - [Chapter 36: Range searching](#)
 - [Chapter 38: Geometric intersection](#)
 - [Chapter 43: Parametric search](#)
 - [Chapter 47: Algorithmic motion planning](#)
 - [Chapter 49: Computer graphics](#)
-

REFERENCES

- [AAS95] P.K. Agarwal, B. Aronov, and S. Suri. Stabbing triangulations by lines in 3D. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 267–276, 1995.
- [AAS97] P.K. Agarwal, B. Aronov, and M. Sharir. Computing envelopes in four dimensions with applications. *SIAM J. Comput.*, 26:1714–1732, 1997.
- [AEM92] P.K. Agarwal, D. Eppstein, and J. Matoušek. Dynamic half-space range searching, geometric optimization and minimum spanning trees. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 80–89, 1992.
- [Aga94] P.K. Agarwal. On stabbing lines for convex polyhedra in 3D. *Comp. Geom. Theory Appl.*, 4:177–189, 1994.

- [AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.
- [AM94] P.K. Agarwal and J. Matoušek. Range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.
- [Ame92] N. Amenta. Finding a line transversal of axial objects in three dimensions. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 66–71, 1992.
- [AF99] B. Aronov and S.J. Fortune. Approximating minimum-weight triangulations in three dimensions. *Discrete Comput. Geom.*, 21:527–549, 1999.
- [APS93] B. Aronov, M. Pellegrini, and M. Sharir. On the zone of an algebraic surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9:177–188, 1993.
- [AB⁺02] B. Aronov, H. Brönnimann, A.Y. Chang, and Y.-J. Chiang. Cost prediction for ray shooting. In *Proc. 18th Annu. ACM Sympos. Geom. Comput.*, pages 293–302, 2002.
- [AS96a] P.K. Agarwal and M. Sharir. Ray shooting amidst convex polyhedra and polyhedral terrains in three dimensions. *SIAM J. Comput.*, 25:100–116, 1996.
- [AS93b] P.K. Agarwal and M. Sharir. Applications of a new space partitioning technique. *Discrete Comput. Geom.*, 9:11–38, 1993.
- [AS96] P.K. Agarwal and M. Sharir. Efficient randomized algorithms for some geometric optimization problems. *Discrete Comput. Geom.*, 16:317–337, 1996.
- [BR79] O. Bottema and B. Roth. *Theoretical Kinematics*. North-Holland, Amsterdam, 1979.
- [CEG⁺92] B. Chazelle, H. Edelsbrunner, L.J. Guibas, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink. Counting and cutting cycles of lines and rods in space. *Comput. Geom. Theory Appl.*, 1:305–323, 1992.
- [CEGS96] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Lines in space: combinatorics and algorithms. *Algorithmica*, 15:428–447, 1996.
- [CEGS93] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric search. *Discrete Comput. Geom.*, 10:183–196, 1993.
- [CEGS94] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.
- [CHX00] D.Z. Chen, X. Hu, and J. Xu. Optimal beam penetrations in two and three dimensions. Proc. ISAAC 2000, *Lecture Notes Comput. Sci.*, volume 1969, pages 491–502, Springer-Verlag, Berlin, 2000.
- [CD99] S.W. Cheng and T.K. Dey. Approximate minimum weight Steiner triangulation in three dimensions. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 205–214, 1999.
- [CKS⁺98] L.P. Chew, K. Kedem, M. Sharir, B. Tagansky, and E. Welzl. Voronoi diagrams of lines in 3-space under polyhedral convex distance functions. *J. Algorithms*, 29:238–255, 1998.
- [dB93] M. de Berg. *Ray Shooting, Depth Orders and Hidden Surface Removal*, volume 703 of *Lecture Notes Comput. Sci.* Springer-Verlag, New York, 1993.
- [dBGH94] M. de Berg, L.J. Guibas, and D. Halperin. Vertical decompositions for triangles in 3-space. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 1–10, 1994.
- [dBH⁺94] M. de Berg, D. Halperin, M.H. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray-shooting and hidden surface removal. *Algorithmica*, 12:31–53, 1994.
- [dBOS94] M. de Berg, M.H. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. *SIAM J. Comput.*, 23:437–446, 1994.

- [DK85] D.P. Dobkin and D.G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.
- [DDP97] F. Durand, G. Drettakis, and C. Puech. The visibility skeleton: a powerful and efficient multi-purpose global visibility tool. *Comput. Graph.* 31:89–100, 1997.
- [Dur99] F. Durand. *3D Visibility: Analytical Study and Applications*. Ph.D. thesis, Univ. J. Fourier, Grenoble, 1999.
- [Epp94] D. Eppstein. Approximating the minimum weight Steiner triangulation. *Discrete Comput. Geom.* 11:163–191, 1994.
- [EE99] D. Eppstein and J. Erickson. Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. *Discrete Comput. Geom.*, 22:569–592, 1999.
- [Eri95] J. Erickson. The relative complexities of some geometric problems. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 85–90, 1995.
- [For99] S.J. Fortune. Topological beam tracing. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 59–68, 1999.
- [FVFH90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, 1990.
- [Hav01] V. Havran. *Heuristic Ray Shooting Algorithms*. Ph.D. thesis, Czech Technical Univ., Praha, Czech Republic, 2001.
- [HP52] W.V.D. Hodge and D. Pedoe. *Methods of Algebraic Geometry*. Cambridge University Press, 1952.
- [HS94] D. Halperin and M. Sharir. New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. *Discrete Comput. Geom.*, 12:313–326, 1994.
- [Jes03] C.M. Jessop. *A Treatise on the Line Complex*. Cambridge University Press, 1903.
- [Meg91] N. Megiddo. Personal communication, 1991.
- [Meh84] K. Mehlhorn. *Multidimensional Searching and Computational Geometry*. Springer-Verlag, Berlin, 1984.
- [MS97] S. Mohaban and M. Sharir. Ray-shooting amidst spheres in three-dimensions and related problems. *SIAM J. Comput.* 26:654–674, 1997.
- [OR01] J. O'Rourke. Computational geometry column 41. *Internat. J. Comp. Geom. Appl.*, 11:239–242, 2001. Also in *SIGACT News*, 32:53–55 (Issue 118), 2001.
- [Pel91] M. Pellegrini. *Combinatorial and Algorithmic Analysis of Stabbing and Visibility Problems in 3-Dimensional Space*. Ph.D. thesis, Courant Institute, New York Univ., 1991. Robotics Lab Tech. Rep. 241.
- [Pel93a] M. Pellegrini. Lower bounds on stabbing lines in 3-space. *Comput. Geom. Theory Appl.*, 3:53–58, 1993.
- [Pel93b] M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9:471–494, 1993.
- [Pel94a] M. Pellegrini. On collision-free placements of simplices and the closest pair of lines in 3-space. *SIAM J. Comput.*, 23:133–153, 1994.
- [Pel94b] M. Pellegrini. On lines missing polyhedral sets in 3-space. *Discrete Comput. Geom.*, 12:203–221, 1994.
- [Pel96] M. Pellegrini. On point location and motion planning in arrangements of simplices. *SIAM J. Comput.*, 25:1061–1081, 1996.
- [Plu65] J. Plücker. On a new geometry of space. *Philos. Trans. Royal Soc. London*, 155:725–791, 1865.

- [PPW93] J. Pach, R. Pollack, and E. Welzl. Weaving patterns of lines and line segments in space. *Algorithmica*, 9:561–571, 1993.
- [PS92] M. Pellegrini and P.W. Shor. Finding stabbing lines in 3-space. *Discrete Comput. Geom.*, 8:191–208, 1992.
- [SAL93] A. Schweikard, J.R. Adler, and J.-C. Latombe. *Motion Planning in Stereotaxic Radio-surgery*. *IEEE Trans. Robot. Autom.*, 9:764–774, 1993.
- [Shi78] B.E. Shimano. *The Kinematic Design and Force Control of Computer Controlled Manipulators*. Ph.D. thesis, Dept. of Mechanical Engineering, Stanford Univ., 1978.
- [Sol98] A. Solan. Cutting Cycles of Rods in Space. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 135–142, 1998.
- [Som51] D.M.Y. Sommerville. *Analytical Geometry of Three Dimensions*. Cambridge University Press, 1951.
- [Spe92] R. Speer. An updated cross-indexed guide to the ray-tracing literature. *Comput. Graphics*, 26:41–72, 1992.
- [SS93] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 246–256, 1993.
- [Sto89] J. Stolfi. Primitives for computational geometry. Tech. Rep. 36, Digital Systems Research Center, Palo Alto, 1989.
- [Sto91] J. Stolfi. *Oriented Projective Geometry: A Framework for Geometric Computations*. Academic Press, San Diego, 1991.
- [ZS01] Y. Zhou and S. Suri. *Shape Sensitive Geometric Permutations*. In *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 234–243, 2001.

38 GEOMETRIC INTERSECTION

David M. Mount

INTRODUCTION

Detecting whether two geometric objects intersect and computing the region of intersection are fundamental problems in computational geometry. Geometric intersection problems arise naturally in a number of applications. Examples include geometric packing and covering, wire and component layout in VLSI, map overlay in geographic information systems, motion planning, and collision detection. In solid modeling, computing the volume of intersection of two shapes is an important step in defining complex solids. In computer graphics, detecting the objects that overlap a viewing window is an example of an intersection problem, as is computing the first intersection of a ray and a collection of geometric solids.

Intersection problems are fundamental to many aspects of geometric computing. It is beyond the scope of this chapter to completely survey this area. Instead we illustrate a number of the principal techniques used in efficient intersection algorithms. This chapter is organized as follows. Section 38.1 discusses intersection primitives, the low-level issues of computing intersections that are common to high-level algorithms. Section 38.2 discusses detecting the existence of intersections. Section 38.3 focuses on issues related to counting the number of intersections and reporting intersections. Section 38.4 deals with problems related to constructing the actual region of intersection. Section 38.5 considers methods for geometric intersections based on spatial subdivisions.

38.1 INTERSECTION PREDICATES

GLOSSARY

Geometric predicate: A function that computes a discrete relationship between basic geometric objects.

Boundary elements: The vertices, edges, and faces of various dimensions that make up the boundary of an object.

Complex geometric objects are typically constructed from a number of primitive objects. Intersection algorithms that operate on complex objects often work by breaking the problem into a series of primitive geometric predicates acting on basic elements, such as points, lines and curves, that form the boundary of the objects involved. Examples of geometric predicates include determining whether two line segments intersect each other or whether a point lies above, below, or on a given line.

Computing these predicates can be reduced to computing the sign of a polynomial, ideally of low degree. In many instances the polynomial arises as the determinant of a symbolic matrix.

Computing geometric predicates in a manner that is efficient, accurate, and robust can be quite challenging. Floating-point computations are fast but suffer from round-off errors, which can result in erroneous decisions. These errors in turn can lead to topological inconsistencies in object representations, and these inconsistencies can cause the run-time failures. Some of the approaches used to address robustness in geometric predicates include approximation algorithm that are robust to floating-point errors [SI94], computing geometric predicates exactly using adaptive floating-point arithmetic [Cla92, ABD⁺97], exact arithmetic combined with fast floating-point filters [BKM⁺95, FV96], and designing algorithms that are based on a restricted set of geometric predicates [BS00].

We will concentrate on geometric intersections involving flat objects (line segments, polygons, polyhedra), but there is considerable interest in computing intersections of curves and surfaces. Predicates for curve and surface intersections are particularly challenging, because the intersection of surfaces of a given algebraic degree generally results in a curve of a significantly higher degree. Computing intersection primitives typically involves solving an algebraic system equations, which can be performed either exactly by algebraic and symbolic methods [Yap93] or approximately by numerical methods [Hof89, MC91]. See [Chapter 41](#).

38.2 INTERSECTION DETECTION

GLOSSARY

Polygonal chain: A sequence of line segments joined end-to-end.

Self-intersecting: Said of a polygonal chain if any pair of nonadjacent edges intersects one another.

Bounding box: A rectangular box surrounding an object, usually axis-aligned (*isothetic*).

Intersection detection, the easiest of all intersection tasks, requires merely determining the existence of an intersection. Nonetheless, detecting intersections efficiently in the absence of simplifying geometric structure can be challenging. As an example, consider the following fundamental intersection problem, posed by John Hopcroft in the early 1980's. Given a set of n points and n lines in the plane, does any point lie on any line? See [Figure 38.2.1](#). A series of efforts to solve *Hopcroft's problem* culminated in the best algorithm known for this problem to date, due to Matoušek [Mat93], which runs in $O(n^{4/3})2^{O(\log^* n)}$. There is reason to believe that this may be close to optimal; Erickson [Eri96] has shown that, in certain models of computation, $\Omega(n^{4/3})$ is a lower bound. Agarwal and Sharir [AS90] have shown that, given two sets of line segments denoted red and blue, it is possible to determine whether there is any red-blue intersection in $O(n^{4/3+\epsilon})$ time, for any positive constant ϵ .

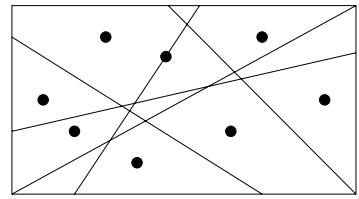


FIGURE 38.2.1
Hopcroft's Problem.

The types of objects considered in this section are polygons, polyhedra, and line segments. Let P and Q denote the two objects to be tested for intersection. Throughout, n_p and n_q denote the combinatorial complexity of P and Q , respectively, that is, the number of vertices, edges, and faces (for polyhedra). Let $n = n_p + n_q$ denote the total complexity.

Table 38.2.1 summarizes a number of results on intersection detection, which will be discussed further in this section. In the table, the terms *convex* and *simple* refer to convex and simple polygons, respectively. The notation $(s(n), q(n))$ in the “Time” column means that the solution involves preprocessing, where a data structure of size $O(s(n))$ is constructed so that intersection detection queries can be answered in $O(q(n))$ time.

TABLE 38.2.1 Intersection detection.

DIM	OBJECTS	TIME	SOURCE
2	convex-convex	$\log n$	[DK83]
	simple-simple	n	[Cha91]
	simple-simple	$(n, s \log^2 n)$	[Mou92]
	line segments	$n \log n$	[SH76]
	Hopcroft's problem	$n^{4/3} 2^{O(\log^* n)}$	[Mat93]
3	convex-convex	n	[DK85]
	convex-convex	$(n, \log n_p \log n_q)$	[DK90]

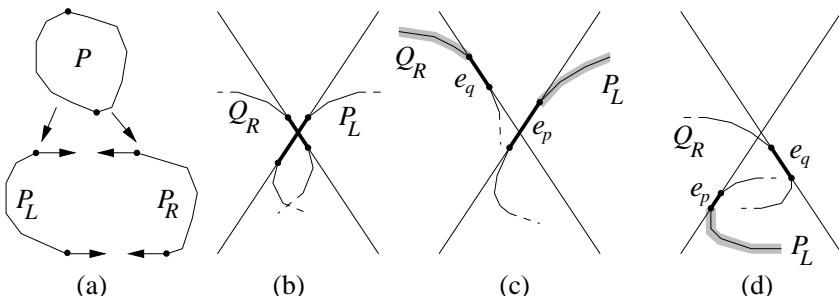
INTERSECTION DETECTION OF CONVEX POLYGONS

Perhaps the most easily understood example of how the structure of geometric objects can be exploited to yield an efficient intersection test is that of detecting the intersection of two convex polygons. There are a number of solutions to this problem that run in $O(\log n)$ time. We present one due to Dobkin and Kirkpatrick [DK83].

Assume that each polygon is given by an array of vertex coordinates, sorted in counterclockwise order. The first step of the algorithm is to find the vertices of each of P and Q with the highest and lowest y -coordinates. This can be done in $O(\log n)$ time by an appropriate modification of binary search and consideration of the direction of the edges incident to each vertex [O'R98, Section 7.6]. After these vertices are located, the boundary of each polygon is split into two semi-infinite convex chains, denoted P_L, P_R and Q_L, Q_R (see Figure 38.2.2(a)). P and Q intersect if and only if P_L and Q_R intersect, and P_R and Q_L intersect.

FIGURE 38.2.2

Intersection detection for two convex polygons.



Consider the case of P_L and Q_R . The algorithm applies a variant of binary search. Consider the median edge e_p of P_L and the median edge e_q of Q_R (shown as heavy lines in the figure). By a simple analysis of the relative positions of these edges and the intersection point of the two lines on which they lie, it is possible to determine in constant time either that the polygons intersect, or that half of at least one of the two boundary chains can be eliminated from further consideration. The cases that arise are illustrated in Figure 38.2.2(b)-(d). The shaded regions indicate the portion of the boundary that can be eliminated from consideration.

SIMPLE POLYGONS

Without convexity, it is generally not possible to detect intersections in sublinear time without preprocessing; but efficient tests do exist.

One of the important intersection questions is whether a closed polygonal chain defines the edges of a simple polygon. The problem reduces to detecting whether the chain is self-intersecting. This problem can be solved efficiently by supposing that the polygonal chain is a simple polygon, attempting to triangulate the polygon, and seeing whether anything goes wrong in the process. Some triangulation algorithms can be modified to detect self intersections. In particular, the problem can be solved in $O(n)$ time by modifying Chazelle's linear-time triangulation algorithm [Cha91]. See [Section 25.2](#).

Another variation is that of determining the intersection of two simple polygons. Chazelle observed that this can also be reduced to testing self intersections in $O(n)$ time by joining the polygons into a single closed chain by a narrow channel as shown in Figure 38.2.3.

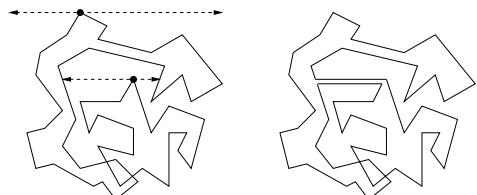


FIGURE 38.2.3

Intersection detection for two simple polygons.

DETECTING INTERSECTIONS OF MULTIPLE OBJECTS

In many applications it is important to know whether any pair of a set of objects intersects one another. Shamos and Hoey showed that the problem of detecting whether a set of n line segments in the plane have an intersecting pair can be solved in $O(n \log n)$ time [SH76]. This is done by plane sweep, which will be discussed below. They also showed that the same can be done for a set of circles. Reichling showed that this can be generalized to detecting whether any pair of m convex n -gons intersects in $O(m \log m \log n)$ time, and whether they all share a common intersection point in $O(m \log^2 n)$ time [Rei88]. Hopcroft, Schwartz, and Sharir [HSS83] showed how to detect the intersection of any pair of n spheres in 3-space in $O(n \log^2 n)$ time and $O(n \log n)$ space by applying a 3D plane sweep.

INTERSECTION DETECTION WITH PREPROCESSING

If preprocessing is allowed, then significant improvements in intersection detection time may be possible. One of the best-known techniques is to filter complex intersection tests is to compute an axis-aligned bounding box for each object. Two objects need to be tested for intersection only if their bounding boxes intersect. It is very easy to test whether two such boxes intersect by comparing their projections on each coordinate axis. For example, in Figure 38.2.4, of the 15 possible pairs of object intersections, all but 3 may be eliminated by the bounding box filter.

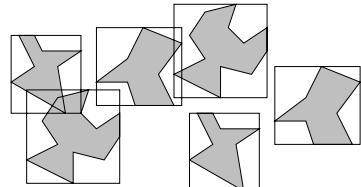


FIGURE 38.2.4
Using bounding boxes as an intersection filter.

It is hard to prove good worst-case bounds for the bounding-box filter since it is possible to create instances of n disjoint objects in which all $O(n^2)$ pairs of bounding boxes intersect. Nonetheless, this popular heuristic tends to perform well in practice. Suri and others [SHH99, ZS99] provided an explanation for this. They proved that if the boxes have bounded aspect ratio and the relative object sizes are within a constant factor each other, then (up to an additive linear term) the number of intersecting boxes is proportional to the number of intersecting object pairs. Combining this with Dopkin and Kirkpatrick's results leads to an algorithm, which given n convex polytopes in dimension d , reports all k intersecting pairs in time $O(n \log^{d-1} n + k \log^{d-1} m)$, where m is the maximum number of vertices in any polytope.

Another example is that of ray shooting in a simple polygon. This is a planar version of a well-known 3D problem in computer graphics. The problem is to preprocess a simple polygon so that given a query ray, the first intersection of the ray with the boundary of the polygon can be determined. After $O(n)$ preprocessing it is possible to answer ray-shooting queries in $O(\log n)$ time. A particularly elegant

solution was given by Hershberger and Suri [HS95]. The polygon is triangulated in a special way, called a ***geodesic triangulation***, so that any line segment that does not intersect the boundary of the polygon crosses at most $O(\log n)$ triangles. Ray-shooting queries are answered by locating the triangle that contains the origin of the ray, and “walking” the ray through the triangulation. See also [Section 25.4](#).

Mount showed how the geodesic triangulation can be used to generalize the bounding box test for the intersection of simple polygons. Each polygon is preprocessed by computing a geodesic triangulation of its exterior. From this it is possible to determine whether they intersect in $O(s \log^2 n)$ time, where s is the minimum number of edges in a polygonal chain that separates the two polygons [Mou92]. Separation sensitive intersections of polygons has been studied in the context of kinetic algorithms for collision detection. See [Chapter 50](#).

CONVEX POLYHEDRA IN 3-SPACE

Extending a problem from the plane to 3-space often involves in a significant increase in difficulty. Nonetheless, Dobkin and Kirkpatrick showed that this detection can be performed efficiently by adapting Kirkpatrick’s hierarchical decomposition of planar triangulations. Given two polyhedra P and Q having boundary complexity n_p and n_q , respectively, their algorithm runs in $O(\log n_p \log n_q)$ time, assuming that each polyhedron has been preprocessed in linear time and space [DK90].

DOBKIN-KIRKPATRICK DECOMPOSITION

Before describing the intersection algorithm, it is important to understand how the hierarchical representation works. Let $P = P_0$ be the initial polyhedron. Assume that P ’s faces have been triangulated. The vertices, edges, and faces of P ’s boundary define a planar graph with triangular faces. Let n denote the number of vertices in this graph. An important fact is that every planar graph has an independent set (a subset of pairwise nonadjacent vertices) that contains a constant fraction of the vertices formed entirely from vertices of bounded degree. Such an independent set is computed and is removed along with any incident edges and faces from P . Then any resulting “holes” in the boundary of P are filled in with triangles, resulting in a convex polyhedron with fewer vertices (cf. [Section 34.6](#)).

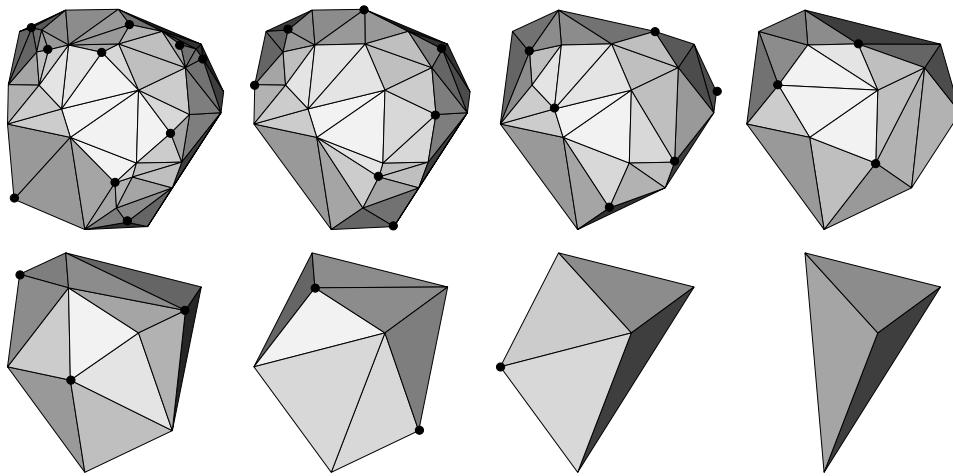
These holes can be triangulated independently of one another, each in constant time. The resulting convex polyhedron is denoted P_1 . The process is repeated until reaching a polyhedron having a constant number of vertices. The result is a sequence of polyhedra, $\langle P_0, P_1, \dots, P_k \rangle$, called the ***Dobkin-Kirkpatrick hierarchy***. Because a constant fraction of vertices are eliminated at each stage, the depth k of the hierarchy is $O(\log n)$. The hierarchical decomposition is illustrated in [Figure 38.2.5](#). The vertices that are eliminated at each stage, which form an independent set, are highlighted in the figure.

INTERSECTION DETECTION ALGORITHM

Suppose that the hierarchical representations of P and Q have already been computed. The intersection detection algorithm actually computes the *separation*, that is, the minimum distance between the two polyhedra. First consider the task of determining the separation between P and a triangle T in 3-space. We start with

FIGURE 38.2.5

Dobkin-Kirkpatrick decomposition of a convex polyhedron.



the top of the hierarchy, P_k . Because P_k and T are both of constant complexity, the separation between P_k and T can be computed in constant time. Given the separation between P_i and T , it is possible to determine the separation between P_{i-1} and T in constant time. This is done by a consideration of the newly added boundary elements of P_{i-1} that lie in the neighborhood of the two closest points.

Given the hierarchical decompositions of two polyhedra P and Q , the Dobkin-Kirkpatrick intersection algorithm begins by computing the separation at the highest common level of the two hierarchies (so that at least one of the decomposed polyhedra is of bounded complexity). They show that in $O(\log n_p + \log n_q)$ time it is possible to determine the separation of the polyhedra at the next lower level of the hierarchies. This leads to a total running time of $O(\log n_p \log n_q)$.

OPEN PROBLEM

Is it possible to detect the intersection of two preprocessed convex polyhedra in $O(\log(n_p + n_q))$ time using linear space?

38.3 INTERSECTION COUNTING AND REPORTING

GLOSSARY

Plane sweep: An algorithm paradigm based on simulating the left-to-right sweep of the plane with a vertical *sweepline*. See [Figure 38.3.1](#).

Red-blue intersection: Segment intersection between segments of two colors, where only intersections between segments of different colors are to be reported.

In many applications geometric intersections can be viewed as a discrete set of entities to be counted or reported. The problems of intersection counting and reporting have been heavily studied in computational geometry from the perspective of *intersection searching*, employing preprocessing and subsequent queries (Chapter 36). We limit our discussion here to batch problems, where the geometric objects are all given at once. In many instances, the best algorithms known for batch counting and reporting reduce the problem to intersection searching.

Table 38.3.1 summarizes a number of results on intersection counting and reporting. The quantity n denotes the combinatorial complexity of the objects, d denotes the dimension of the space, and k denotes the number of intersections. Because every pair of elements might intersect, the number of intersections k may generally be as large as $O(n^2)$, but it is frequently much smaller.

TABLE 38.3.1 Intersection counting and reporting.

PROBLEM	DIM	OBJECTS	TIME	SOURCE
Reporting	2	line segments	$n \log n + k$	[CE92][Bal95]
	2	red-blue segments (general)	$n^{4/3} \log^{O(1)} n + k$	[Aga90][Cha93]
	2	red-blue segments (disjoint)	$n + k$	[FH95]
	d	orthogonal segments	$n \log^{d-1} n + k$	[EM81]
Counting	2	line segments	$n^{4/3} \log^{O(1)} n$	[Aga90][Cha93]
	2	red-blue segments (general)	$n^{4/3} \log^{O(1)} n$	[Aga90][Cha93]
	2	red-blue segments (disjoint)	$n \log n$	[CEGS94]
	d	orthogonal segments	$n \log^{d-1} n$	[EM81, Cha88]

REPORTING LINE SEGMENT INTERSECTIONS

Consider the problem of reporting the intersections of n line segments in the plane. This problem is an excellent vehicle for introducing the powerful technique of plane sweep (Figure 38.3.1). The plane-sweep algorithm maintains an active list of segments that intersect the current sweepline, sorted from bottom to top by intersection point. If two line segments intersect, then at some point prior to this intersection they must be consecutive in the sweep list. Thus, we need only test consecutive pairs in this list for intersection, rather than testing all $O(n^2)$ pairs.

At each step the algorithm advances the sweepline to the next event: a line segment endpoint or an intersection point between two segments. Events are stored in a *priority queue* by their x -coordinates. After advancing the sweepline to the next event point, the algorithm updates the contents of the active list, tests new consecutive pairs for intersection, and inserts any newly-discovered events in the priority queue. For example, in Figure 38.3.1 the locations of the sweepline are shown with dashed lines.

Bentley and Ottmann showed that by using plane sweep it is possible to report all k intersecting pairs of n line segments in $O((n+k) \log n)$ time [BO79]. If the number of intersections k is much less than the $O(n^2)$ worst-case bound, then this is great savings over a brute-force test of all pairs.

For many years the question of whether this could be improved to $O(n \log n + k)$

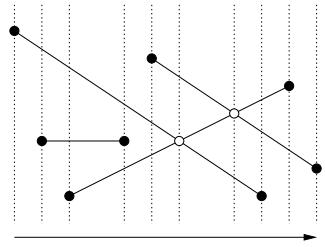


FIGURE 38.3.1

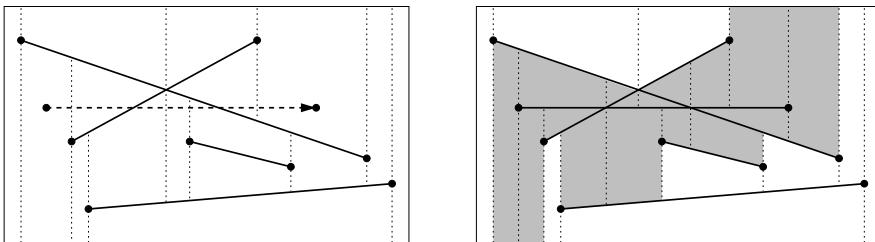
Plane sweep for line segment intersection.

was open, until Edelsbrunner and Chazelle presented such an algorithm [CE92]. This algorithm is optimal with respect to running time because at least $\Omega(k)$ time is needed to report the result, and it can be shown that $\Omega(n \log n)$ time is needed to detect whether there is any intersection at all. However, their algorithm uses $O(n + k)$ space. Balaban [Bal95] how to achieve the same running time using only $O(n)$ space. Clarkson and Shor [CS89] and later Mulmuley [Mul91] presented simpler, randomized algorithms with the same expected running time but using only $O(n)$ space.

Mulmuley's algorithm is particularly elegant. It involves maintaining a **trapezoidal decomposition**, a subdivision which results by shooting a vertical ray up and down from each segment endpoint and intersection point until it hits another segment. The algorithm inserts the segments one by one in random order by “walking” each segment through the subdivision and updating the decomposition as it goes. (This is shown in Figure 38.3.2, where the broken horizontal line on the left is being inserted and the shaded regions on the right are the newly created trapezoids.)

FIGURE 38.3.2

Incremental construction of a trapezoidal decomposition.



RED-BLUE INTERSECTION PROBLEMS

Among the numerous variations of the segment intersection problem, the most widely studied is the problem of computing intersections that arise between two sets of segments, say red and blue, whose total size is n . The goal is to compute all **bichromatic intersections**, that is, intersections that arise when a red segment intersects a blue segments. Let k denote the number of such intersections.

The case where there are no monochromatic (blue-blue or red-red) intersections is particularly important. It arises, for example, when two planar subdivisions are

overlaid, called the *map overlay* problem in GIS applications, as well as in many intersection algorithms based on divide-and-conquer. (See Figure 38.3.3.) In this case the problem can be solved by in $O(n \log n + k)$ time by any optimal monochromatic line-segment intersection algorithm. This problem seems to be somewhat simpler than the monochromatic case, because Mairson and Stolfi [MS88] showed the existence of an $O(n \log n + k)$ algorithm prior to the discovery of these optimal monochromatic algorithms. Chazelle et al. [CEGS94] presented an algorithm based on a simple but powerful data structure, called the *hereditary segment tree*. Chan [Cha94] presented a practical approach based on a plane sweep of the trapezoidal decomposition of the two sets. Guibas and Seidel [GS87] showed that, if the segments form a simple connected convex subdivision of the plane, the problem can be solved more efficiently in $O(n + k)$ time. This was extended to simply connected subdivisions that are not necessarily convex by Finke and Hinrichs [FH95].

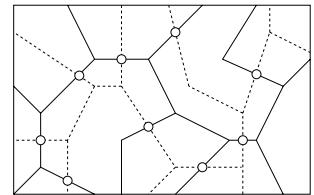


FIGURE 38.3.3
Overlaying planar subdivisions.

The problem is considerably more difficult if monochromatic intersections exist. This is because there may be quadratically many monochromatic intersections, even if there are no bichromatic intersections. Agarwal [Aga90] and Chazelle [Cha93] showed that the k bichromatic intersections can be reported in $O(n^{4/3} \log^{O(1)} n + k)$ time through the use of a partitioning technique called *cuttings*. Basch et al. [BGR96] showed that if the set of red-segments forms a connected set and the blue set does as well, then it is possible to report all bichromatic intersections in $O((n+k) \log^{O(1)} n)$ time. Agarwal et al. [AdBH⁺02] and Gupta et al. [GJS99] considered a multi-chromatic variant in which the input consists of m convex polygons and the objective is to report all intersections between pairs of polygons. They show that many of the same techniques can be applied to this problem and present algorithms with similar running times.

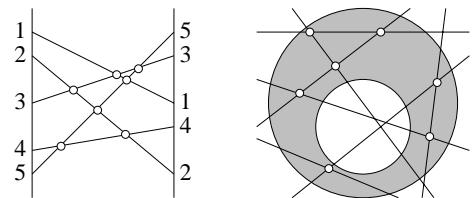
COUNTING LINE SEGMENT INTERSECTIONS

Efficient intersection counting often requires quite different techniques from reporting because it is not possible to rely on the lower bound of k needed to report the results. Nonetheless, a number of the efficient intersection reporting algorithms can be modified to count intersections efficiently. For example, methods based on cuttings [Aga90, Cha93] can be used to count the number of intersections among n planar line segments and bichromatic intersections between n red and blue segments in $O(n^{4/3} \log^{O(1)} n)$ time. If there are no monochromatic intersections then the hereditary segment tree [CEGS94] can be used to count the number bichromatic intersections in $O(n \log n)$ time.

Many of the algorithms for performing segment intersection exploit the observation that if the line segments span a closed region, it is possible to infer the number

of segment intersections within the region simply by knowing the order in which the lines intersect the boundary of the region. Consider, for example, the problem of counting the number of line intersections that occur within a vertical strip in the plane. This problem can be solved in $O(n \log n)$ time by sorting the points according to their intersections on the left side of the strip, computing the associated permutation of indices on the right side, and then counting the number *inversions* in the resulting sequence [DMN92, Mat91]. An inversion is any pair of values that are not in sorted order. See Figure 38.3.4. Inversion counting can be performed by a simple modification of the Mergesort algorithm. It is possible to generalize this idea to regions whose boundary is not simply connected [Asa94, MN01].

FIGURE 38.3.4
Intersections and inversion counting.

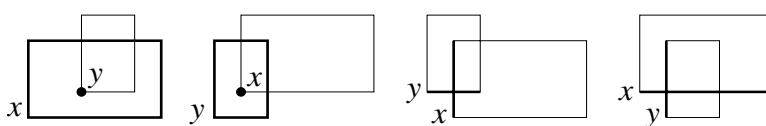


INTERSECTION SEARCHING AND RANGE SEARCHING

Range and intersection searching are powerful tools that can be applied to more complex intersection counting and reporting problems. This fact was first observed by Dobkin and Edelsbrunner [DE87], and has been applied to many other intersection searching problems since.

As an illustration, consider the problem of counting all intersecting pairs from a set of n rectangles. Edelsbrunner and Maurer [EM81] observed that intersections among orthogonal objects can be broken down to a set of orthogonal search queries (see Figure 38.3.5). For each rectangle x we can count all the intersecting rectangles of the set satisfying each of these conditions and sum them. Each of these counting queries can be answered in $O(\log n)$ time after $O(n \log n)$ preprocessing time [Cha88], leading to an overall $O(n \log n)$ time algorithm. This counts every intersection twice and counts self intersections, but these are easy to factor out from the final result. Generalizations to hyperrectangle intersection counting in higher dimensions are straightforward, with an additional factor of $\log n$ in time and space for each increase in dimension. We refer the reader to Chapter 36 for more information on intersection searching and its relationship to range searching.

FIGURE 38.3.5
Types of intersections between rectangles x and y .



38.4 INTERSECTION CONSTRUCTION

GLOSSARY

Regularization: Discarding measure-zero parts of the result of an operation by taking the closure of the interior.

Clipping: Computing the intersection of each of many polygons with an axis-aligned rectangular viewing *window*.

Kernel of a polygon: The set of points that can see every point of the polygon.
(See [Section 26.1](#).)

Intersection construction involves determining the region of intersection between geometric objects. Many of the same techniques that are used for computing geometric intersections are used for computing Boolean operations in general (e.g., union and difference). Many of the results presented here can be applied to these other problems as well. Typically intersection construction reduces to the following tasks: (1) compute the intersection between the boundaries of the objects; (2) if the boundaries do not intersect then determine whether one object is nested within the other; and (3) if the boundaries do intersect then classify the resulting boundary fragments and piece together the final intersection region.

When Boolean operations are computed on solid geometric objects, it is possible that lower-dimensional “dangling” components may result. It is common to eliminate these lower-dimensional components by a process called *regularization* (see [Section 56.1.1](#)). The *regularized* intersection of P and Q , denoted $P \cap^* Q$, is defined formally to be the closure of the interior of the standard intersection $P \cap Q$ (see Figure 38.4.1).

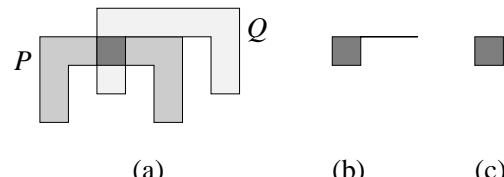


FIGURE 38.4.1

Regularized intersection: (a) Polygons P and Q ;
(b) $P \cap Q$; (c) $P \cap^* Q$.

Some results on intersection construction are summarized in [Table 38.4.1](#), where n is the total complexity of the objects being intersected, and k is the number of pairs of intersecting edges.

CONVEX POLYGONS

Determining the intersection of two convex polygons is illustrative of many intersection construction algorithms. Observe that the intersection of two convex polygons having a total of n edges is either empty or a convex polygon with at most n edges.

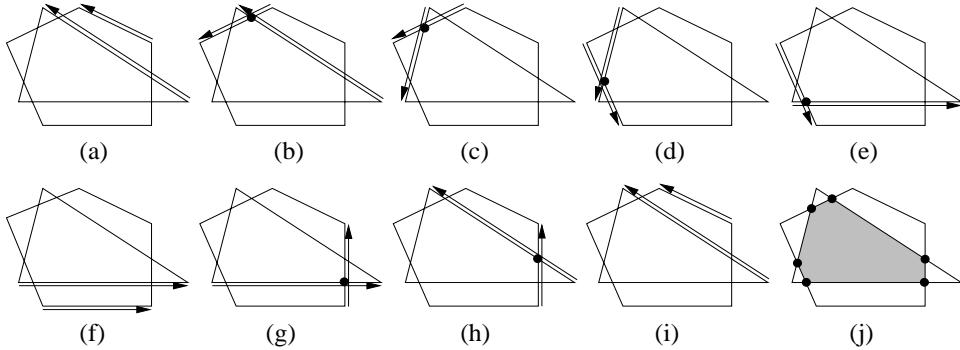
TABLE 38.4.1 Intersection construction.

DIM	OBJECTS	TIME	SOURCE
2	convex-convex	n	[SH76, OCON82]
2	simple-simple	$n \log n + k$	[CE92]
2	kernel	n	[LP79]
3	convex-convex	n	[Cha92]

O'Rourke et al. present an $O(n)$ time algorithm, which given two convex polygons P and Q determines their intersection [OCON82].

The algorithm can be viewed as a geometric generalization of merging two sorted lists. It performs a counterclockwise traversal of the boundaries of the two polygons. The algorithm maintains a pair of edges, one from each polygon. From a consideration of the relative positions of these edges the algorithm advances one of them to the next edge in counterclockwise order around its polygon. Intuitively, this is done in such a way that these two edges effectively “chase” each other around the boundary of the intersection polygon (see Figure 38.4.2(a)-(i)).

FIGURE 38.4.2
Convex polygon intersection construction.



OPEN PROBLEM

Reichling has shown that it is possible to detect whether m convex n -gons share a common point in $O(m \log^2 n)$ time [Rei88]. Is there an output-sensitive algorithm of similar complexity for constructing the intersection region?

SIMPLE POLYGONS AND CLIPPING

As with convex polygons, computing the intersection of two simple polygons reduces to first computing the points at which the two boundaries intersect and then classifying the resulting edge fragments. Computing the edge intersections and edge fragments can be performed by any algorithm for reporting line segment intersec-

tions. Classifying the edge fragments is a simple task. Margalit and Knott describe a method for edge classification that works not only for intersection, but for any Boolean operation on the polygons [MK89].

Clipping a set of polygons to a rectangular window is a special case of simple polygon intersection that is particularly important in computer graphics (see [Section 49.3](#)). One popular algorithm for this problem is the Sutherland-Hodgman algorithm [FvD⁺90]. It works by intersecting each polygon with each of the four halfplanes that bound the clipping window. The algorithm traverses the boundary of the polygon, and classifies each edge as lying either entirely inside, entirely outside, or crossing each such halfplane.

An elegant feature of the algorithm is that it effectively “pipelines” the clipping process by clipping each edge against one of the window’s four sides and then passing the clipped edge, if it is nonempty, to the next side to be clipped. This makes the algorithm easy to implement in hardware. An unusual consequence, however, is that if a polygon’s intersection with the window has multiple connected components (as can happen with a nonconvex polygon), then the resulting clipped polygon consists of a single component connected by one or more “invisible” channels that run along the boundary of the window (see Figure 38.4.3).

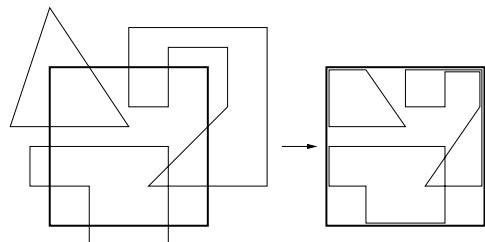


FIGURE 38.4.3
Clipping using the Sutherland-Hodgman algorithm.

INTERSECTION CONSTRUCTION IN HIGHER DIMENSIONS

Intersection construction in higher dimensions, and particularly in dimension 3, is important to many applications such as solid modeling. The basic paradigm of computing boundary intersections and classifying boundary fragments applies here as well. Muller and Preparata gave an $O(n \log n)$ algorithm that computes the intersection of two convex polyhedra in 3-space (see [PS85]). The existence of a linear-time algorithm remained open for years until Chazelle discovered such an algorithm [Cha92]. He showed that the Dobkin-Kirkpatrick hierarchical representation of polyhedra can be applied to the problem. A particularly interesting element of his algorithm is the use of the hierarchy for representing the interior of each polyhedron, and a dual hierarchy for representing the exterior of each polyhedron. Dobrindt, Mehlhorn, and Yvinec [DMY93] presented an output-sensitive algorithm for intersecting two polyhedra, one of which is convex.

Another class of problems can be solved efficiently are those involving *polyhedral terrains*, that is, a polyhedral surface that intersects every vertical line in at most one point. Chazelle et al. [CEGS94] show that the hereditary segment tree can be applied to compute the smallest vertical distance between two polyhedral terrains in roughly $O(n^{4/3})$ time. They also show the upper envelope of two polyhedral terrains can be computed in $O(n^{3/2+\epsilon} + k \log^2 n)$ time, where ϵ is an arbitrary constant and k is the number of edges in the upper envelope.

KERNELS AND THE INTERSECTION OF HALFSPACES

Because of the highly structured nature of convex polygons, algorithms for convex polygons can often avoid additional $O(\log n)$ factors that seem to be necessary when dealing with less structured objects. An example of this structure arises in computing the kernel of a simple polygon: the (possibly empty) locus of points that can see every point in the polygon (the shaded region of Figure 38.4.4). Put another way, the kernel is the intersection of inner halfplanes defined by all the sides of P . The kernel of P is a convex polygon having at most n sides. Lee and Preparata gave an $O(n)$ time algorithm for constructing it [LP79] (see also [Table 26.3.1](#)). Their algorithm operates by traversing the boundary of the polygon, and incrementally updating the boundary of the kernel as each new edge is encountered.

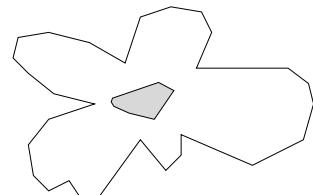


FIGURE 38.4.4
The kernel of a simple polygon.

The general problem of computing the intersection of halfplanes, when the halfplanes do not necessarily arise from the sides of a simple polygon, requires $\Omega(n \log n)$ time. See [Chapter 22](#) for more information on this problem.

38.5 METHODS BASED ON SPATIAL SUBDIVISIONS

So far we have considered methods with proven worst-case asymptotic efficiency. However, there are numerous approaches to intersection problems for which worst-case efficiency is hard to establish, but that practical experience has shown to be quite efficient on the types of inputs that often arise in practice. Most of these methods are based on subdividing space into disjoint regions, or *cells*. Intersections can be computed by determining which objects overlap each cell, and then performing primitive intersection tests between objects that overlap the same cell.

GRIDS

Perhaps the simplest spatial subdivision is based on “bucketing” with square grids. Space is subdivided into a regular grid of squares (or generally hypercubes) of equal side length. The side length is typically chosen so that either the total number of cells is bounded, or the expected number of objects overlapping each cell is bounded. Edahiro et al. [ETHA89] showed that this method is competitive with and often performs much better than more sophisticated data structures for reporting intersections between randomly generated line segments in the plane. Conventional wisdom is that grids perform well as long as the objects are small on average and their distribution is roughly uniform.

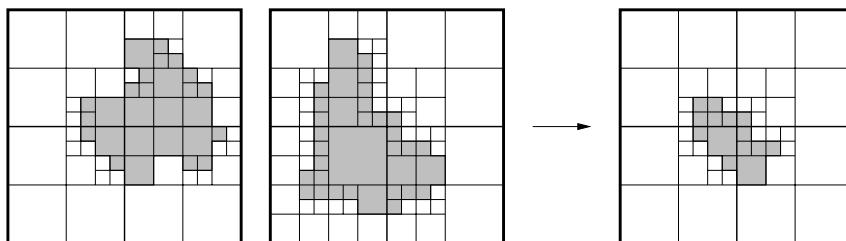
HIERARCHICAL SUBDIVISIONS

The principle shortcoming of grids is their inability to deal with nonuniformly distributed objects. Hierarchical subdivisions of space are designed to overcome this weakness. There is quite a variety of different data structures based on hierarchical subdivisions, but almost all are based on the principal of recursively subdividing space into successively smaller regions, until each region is sufficiently simple in the sense that it overlaps only a small number of objects. When a region is subdivided, the resulting subregions are its *children* in the hierarchy. Well-known examples of hierarchical subdivisions for storing geometric objects include quadtrees and k - d trees, R-trees, and binary space partition (BSP) trees. See [Sam90b] for a discussion of all of these.

Intersection construction with hierarchical subdivisions can be performed by a process of *merging* the two hierarchical spatial subdivisions. This method is described by Samet for quadtrees [Sam90a] and Naylor et al. [NAT90] for BSP trees. To illustrate the idea on a simple example, consider a quadtree representation of two black-and-white images. The problem is to compute the intersection of the two black regions. For example, in Figure 38.5.1 the two images on the left are intersected, resulting in the image on the right.

FIGURE 38.5.1

Intersection of images using quadtrees.



The algorithm recursively considers two overlapping square regions from each quadtree. A region of the quadtree is *black* if the entire region is black, *white* if the entire region is white, and *gray* otherwise. If either region is white, then the result is white. If either region is black, then the result is the other region. Otherwise both regions are gray, and we apply the procedure recursively to each of the four pairs of overlapping children.

38.6 SOURCES

Geometric intersections and related topics are covered in general sources on computational geometry [dBvK⁺00, O'R98, Mul93, Ede87, PS85, Meh84]. A good source of information on the complexity of the lower envelopes and faces in arrangements is the book by Sharir and Agarwal [SA95]. Intersections of convex objects are

discussed in the paper by Chazelle and Dobkin [CD87]. For information on data structures useful for geometric intersections see Samet’s books [Sam90a, Sam90b]. Sources on computing intersection primitives include O’Rourke’s book on computational geometry [O’R98], Yap’s book [Yap93] on algebraic algorithms, and most texts on computer graphics, for example [FvD⁺90]. For 3D surface intersections consult books on solid modeling, including those by Hoffmann [Hof89] and Mäntylä [Män88]. The *Graphics Gems* series (e.g., [Pae95]) contains a number of excellent tips and techniques for computing geometric operations including intersection primitives.

RELATED CHAPTERS

- [Chapter 22: Convex hull computations](#)
- [Chapter 24: Arrangements](#)
- [Chapter 25: Triangulations](#)
- [Chapter 36: Range searching](#)
- [Chapter 37: Ray shooting and lines in space](#)
- [Chapter 49: Computer graphics](#)
- [Chapter 53: Splines and geometric modeling](#)

REFERENCES

- [ABD⁺97] F. Avnaim, J.-D. Boissonnat, O. Devillers, F.P. Preparata, and M. Yvinec. Evaluating signs of determinants using single-precision arithmetic. *Algorithmica*, 17:111–132, 1997.
- [AdBH⁺02] P.K. Agarwal, M. de Berg, S. Har-Peled, M.H. Overmars, M. Sharir, and J. Vahrenhold. Reporting intersecting pairs of convex polytopes in two and three dimensions. *Comput. Geom. Theory Appl.*, 23:197–207, 2002.
- [Aga90] P.K. Agarwal. Partitioning arrangements of lines: II. Applications. *Discrete Comput. Geom.*, 5:533–573, 1990.
- [AS90] P.K. Agarwal and M. Sharir. Red-blue intersection detection algorithms, with applications to motion planning and collision detection. *SIAM J. Comput.*, 19:297–321, 1990.
- [Asa94] Te. Asano. Reporting and counting intersections of lines within a polygon. In *Proc. 5th Annu. Internat. Sympos. Algorithms Comput.*, volume 834 of *Lecture Notes Comput. Sci.*, pages 652–659. Springer-Verlag, Berlin, 1994.
- [Bal95] I.J. Balaban. An optimal algorithm for finding segment intersections. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 211–219, 1995.
- [BGR96] J. Basch, L.J. Guibas, and G.D. Ramkumar. Reporting red-blue intersections between two sets of connected line segments. In *Proc. 4th Annu. European Sympos. Algorithms*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 302–319. Springer-Verlag, Berlin, 1996.
- [BKM⁺95] C. Burnikel, J. Könnemann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Exact geometric computation in LEDA. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C18–C19, 1995.

- [BO79] J.L. Bentley and T.A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.
- [BS00] J.-D. Boissonnat and J. Snoeyink. Efficient algorithms for line and curve segment intersection using restricted predicates. *Comput. Geom. Theory Appl.*, 16:35–52, 2000.
- [CD87] B. Chazelle and D.P. Dobkin. Intersection of convex objects in two and three dimensions. *J. Assoc. Comput. Mach.*, 34:1–27, 1987.
- [CE92] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. Assoc. Comput. Mach.*, 39:1–54, 1992.
- [CEGS94] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.
- [Cha88] B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17:427–462, 1988.
- [Cha91] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.
- [Cha92] B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J. Comput.*, 21:671–696, 1992.
- [Cha93] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9:145–158, 1993.
- [Cha94] T.M. Chan. A simple trapezoid sweep algorithm for reporting red/blue segment intersections. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 263–268, 1994.
- [Cla92] K.L. Clarkson. Safe and effective determinant evaluation. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 387–395, October 1992.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [dBvK⁺00] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*, 2nd edition. Springer-Verlag, Berlin, 2000.
- [DE87] D.P. Dobkin and H. Edelsbrunner. Space searching for intersecting objects. *J. Algorithms*, 8:348–361, 1987.
- [DK83] D.P. Dobkin and D.G. Kirkpatrick. Fast detection of polyhedral intersection. *Theoret. Comput. Sci.*, 27:241–253, 1983.
- [DK85] D.P. Dobkin and D.G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *J. Algorithms*, 6:381–392, 1985.
- [DK90] D.P. Dobkin and D.G. Kirkpatrick. Determining the separation of preprocessed polyhedra—a unified approach. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes Comput. Sci.*, pages 400–413, Springer-Verlag, Berlin, 1990.
- [DMN92] M.B. Dillencourt, D.M. Mount, and N.S. Netanyahu. A randomized algorithm for slope selection. *Internat. J. Comput. Geom. Appl.*, 2:1–27, 1992.
- [DMY93] K. Dobrindt, K. Mehlhorn, and M. Yvinec. A complete and efficient algorithm for the intersection of a general and a convex polyhedron. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes Comput. Sci.*, pages 314–324, Springer-Verlag, Berlin, 1993.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monogr. Theoret. Comput. Sci.* Springer-Verlag, Heidelberg, 1987.
- [EM81] H. Edelsbrunner and H.A. Maurer. On the intersection of orthogonal objects. *Inform. Process. Lett.*, 13:177–181, 1981.

- [Eri96] J. Erickson. New lower bounds for Hopcroft's problem. *Discrete Comput. Geom.*, 16:389–418, 1996.
- [ETHA89] M. Edahiro, K. Tanaka, R. Hoshino, and Ta. Asano. A bucketing algorithm for the orthogonal segment intersection search problem and its practical efficiency. *Algorithmica*, 4:61–76, 1989.
- [FH95] U. Finke and K. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 119–126, 1995.
- [FV96] S.J. Fortune and C.J. van Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. *ACM Trans. Graph.*, 15:223–248, 1996.
- [FvD⁺90] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, 1990.
- [GJS99] P. Gupta, R. Janardan, and M. Smid. Efficient algorithms for counting and reporting pairwise intersections between convex polygons. *Inform. Process. Lett.*, 69:7–13, 1999.
- [GS87] L.J. Guibas and R. Seidel. Computing convolutions by reciprocal search. *Discrete Comput. Geom.*, 2:175–193, 1987.
- [Hof89] C. Hoffmann. *Geometric and Solid Modeling*. Morgan Kaufmann, San Mateo, 1989.
- [HS95] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.
- [HSS83] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. Efficient detection of intersections among spheres. *Internat. J. Robot. Res.*, 2:77–80, 1983.
- [LP79] D.T. Lee and F.P. Preparata. An optimal algorithm for finding the kernel of a polygon. *J. Assoc. Comput. Mach.*, 26:415–421, 1979.
- [Män88] M. Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Rockville, 1988.
- [Mat91] J. Matoušek. Randomized optimal algorithm for slope selection. *Inform. Process. Lett.*, 39:183–187, 1991.
- [Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.
- [MC91] D. Manocha and J.F. Canny. A new approach for surface intersection. *Internat. J. Comput. Geom. Appl.*, 1:491–516, 1991.
- [Meh84] K. Mehlhorn. *Multi-dimensional Searching and Computational Geometry*, volume 3 of *Data Structures and Algorithms*. Springer-Verlag, Heidelberg, 1984.
- [MK89] A. Margalit and G.D. Knott. An algorithm for computing the union, intersection or difference of two polygons. *Comput. & Graph.*, 13:167–183, 1989.
- [MN01] D.M. Mount and N.S. Netanyahu. Efficient randomized algorithms for robust estimation of circular arcs and aligned ellipses. *Comput. Geom. Theory Appl.*, 19:1–33, 2001.
- [Mou92] D.M. Mount. Intersection detection and separators for simple polygons. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 303–311, 1992.
- [MS88] H.G. Mairson and J. Stolfi. Reporting and counting intersections between two sets of line segments. In R.A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, volume F40 of *NATO ASI*, pages 307–325. Springer-Verlag, Berlin, 1988.
- [Mul91] K. Mulmuley. A fast planar partition algorithm, II. *J. Assoc. Comput. Mach.*, 38:74–103, 1991.

- [Mul93] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1993.
- [NAT90] B. Naylor, J.A. Amatodes, and W. Thibault. Merging BSP trees yields polyhedral set operations. *Proc. ACM Conf. SIGGRAPH 90*, pages 115–124, 1990.
- [OCON82] J. O'Rourke, C.-B. Chien, T. Olson, and D. Naddor. A new linear algorithm for intersecting convex polygons. *Comput. Graph. Image Process.*, 19:384–391, 1982.
- [O'R98] J. O'Rourke. *Computational Geometry in C*, Second Edition. Cambridge University Press, 1998.
- [Pae95] A.W. Paeth, editor. *Graphics Gems V*. Academic Press, Boston, 1995.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [Rei88] M. Reichling. On the detection of a common intersection of k convex objects in the plane. *Inform. Process. Lett.*, 29:25–29, 1988.
- [SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [Sam90a] H. Samet. *Applications of Spatial Data Structures*. Addison-Wesley, Reading, 1990.
- [Sam90b] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, 1990.
- [SH76] M.I. Shamos and D. Hoey. Geometric intersection problems. In *Proc. 17th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 208–215, 1976.
- [SHH99] S. Suri, P.M. Hubbard, and J.F. Hughes. Analyzing bounding boxes for object intersection. *ACM Trans. Graphics*, 18:257–277, 1999.
- [SI94] K. Sugihara and M. Iri. A robust topology-oriented incremental algorithm for Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, 4:179–228, 1994.
- [Yap93] C.K. Yap. *Fundamental Problems in Algorithmic Algebra*. Princeton University Press, Princeton, 1993.
- [ZS99] Y. Zhou and S. Suri. Analysis of a bounding box heuristic for object intersection. *J. Assoc. Comput. Mach.*, 46:833–857, 1999.

39 NEAREST NEIGHBORS IN HIGH-DIMENSIONAL SPACES

Piotr Indyk

INTRODUCTION

In this chapter we consider the following problem: given a set P of points in a high-dimensional space, construct a data structure which given any *query* point q finds the point in P closest to q . This problem, called ***nearest neighbor search***¹, is of significant importance to several areas of computer science, including pattern recognition, searching in multimedia data, vector compression [GG91], computational statistics [DW82], and data mining. Many of these applications involve data sets which are very large (e.g., a database containing Web documents could contain over one billion documents). Moreover, the dimensionality of the points is usually large as well (e.g., in the order of a few hundred). Therefore, it is crucial to design algorithms which scale well with the database size as well as with the dimension.

The nearest-neighbor problem is an example of a large class of ***proximity problems***, which, roughly speaking, are problems whose definitions involve the notion of distance between the input points. Apart from nearest-neighbor search, the class contains problems like closest pair, diameter, minimum spanning tree and variants of clustering problems.

Many of these problems were among the first investigated in the field of computational geometry. As a result of this research effort, many efficient solutions have been discovered for the case when the points lie in a space of *constant* dimension. For example, if the points lie in the plane, the nearest-neighbor problem can be solved with $O(\log n)$ time per query, using only $O(n)$ storage [SH75, LT80]. Similar results can be obtained for other problems as well. Unfortunately, as the dimension grows, the algorithms become less and less efficient. More specifically, their space or time requirements grow *exponentially* in the dimension. In particular, the nearest-neighbor problem has a solution with $O(d^{O(1)} \log n)$ query time, but using roughly $n^{O(d)}$ space [Cla88, Mei93]. Alternatively, if one insists on linear or near-linear storage, the best known running time bound for *random* input is of the form $\min(2^{O(d)}, dn)$, which is essentially linear in n even for moderate d . Worse still, the exponential dependence of space and/or time on the dimension (called the “curse of dimensionality”) has been observed in applied settings as well. Specifically, it is known that many popular data structures (using linear or near-linear storage), exhibit query time linear in n when the dimension exceeds a certain threshold (usually 10–20, depending on the number of points), e.g., see [W⁺98] for more information.

The lack of success in removing the exponential dependence on the dimension led many researchers to conjecture that no efficient solutions exists for these problems when the dimension is sufficiently large (e.g., see [MP69]). At the same time,

¹Many other names occur in literature, including *best match*, *post office problem* and *nearest neighbor*.

it raised the question: Is it possible to remove the exponential dependence on d , if we allow the answers to be *approximate*. The notion of approximation is best explained for nearest-neighbor search: instead of reporting a point p closest to q , the algorithm is allowed to report *any* point within distance $(1 + \epsilon)$ times the distance from q to p . Similar definitions can be naturally applied to other problems. Note that this approach is similar to designing efficient approximation algorithms for NP-hard problems.

During recent years, several researchers have shown that indeed in many cases approximation enables reduction of the dependence on dimension from exponential to polynomial. In this chapter we will survey these results. In addition, we will discuss the issue of *proving* that the curse of dimensionality is inevitable if one insists on exact answers, and survey the known results in this direction.

Although this chapter is devoted almost entirely to approximation algorithms with running times polynomial in the dimension, the notion of approximate nearest neighbor was first formulated in the context of algorithms with exponential query times. [Chapter 51.7](#) of this Handbook covers those results in more detail.

Before proceeding further, we mention that our treatment of the topic is primarily theoretical. For experimental evaluations and applications of the algorithms described in this chapter, see e.g., [GIM99, CD⁺00, HGI00, Shi00, Buh01, BT01, Ya01, Buh02, O⁺02, GS⁺03]. In addition, we focus on algorithms operating in main memory. For external memory algorithms, see e.g., recent proceedings of *SIGMOD* and *VLDB* conferences.

39.1 APPROXIMATE NEAR NEIGHBOR

Almost all algorithms for proximity problems in high-dimensional spaces proceed by reducing the problem to the problem of finding an *approximate near neighbor*, which is the decision version of the approximate nearest-neighbor problem. Thus, we start from describing the results for the former problem.

For the definitions of metric spaces and normed spaces, see [Chapter 8](#).

GLOSSARY

Approximate Near Neighbor, or (r, c) -NN: Given a set P on n points in a metric space $M = (X, D)$, design a data structure that supports the following operation: For any query $q \in X$, if there exists $p \in P$ such that $D(p, q) \leq r$, find a point $p' \in P$ such that $D(q, p') \leq cr$

Dynamic problems: Problems which involve designing a data structure for a set of points (e.g., approximate near neighbor) and support insertions and deletions of points. We distinguish dynamic problems from their *static* versions by adding the word “Dynamic” (or letter “D”) in front of their names (or acronyms). E.g., the dynamic version of the approximate near-neighbor problem is denoted by (r, c) -DNN.

Hamming metric: A metric (Σ^d, D) where Σ is a set of *symbols*, and for any $p, q \in \Sigma^d$, $D(p, q)$ is equal to the number of $i \in \{1 \dots d\}$ such that $p_i \neq q_i$.

TABLE 39.1.1 Approximate Near Neighbors.

#	APPROX.	QUERY TIME	SPACE	UPDATE TIME
1a	Source: [KOR00] (cf. [HIM03]); Randomness: Monte Carlo			
	$1 + \epsilon$	$d \log n / \min(\epsilon^2, 1)$	$n^{O(1/\epsilon^2 + \log(1+\epsilon)/(1+\epsilon))}$	$n^{O(1/\epsilon^2 + \log(1+\epsilon)/(1+\epsilon))}$
1b	Source: [Ind01a]; Randomness: Monte Carlo			
	$1 + \epsilon$	$n^{O(\frac{1+\log(1+\epsilon)}{1+\epsilon})}$	dn	$d \log^{O(1)} n$
2	Source: [HIM03]; Randomness: Monte Carlo			
	$1 + \epsilon$	$dn^{1/(1+\epsilon)}$	$n^{1+1/(1+\epsilon)} + dn$	$dn^{1/(1+\epsilon)}$
3	Source: [Ind00]; Randomness: Las Vegas			
	$1 + \epsilon$	$(d \log n / \epsilon)^{O(1)}$	$n^{1/\epsilon^{O(1)}}$	static
4	Source: [Ind00]; Randomness: Deterministic			
	$3 + \epsilon$	$(d \log n / \epsilon)^{O(1)}$	$n^{1/\epsilon^{O(1)}}$	static

RANDOM PROJECTION APPROACH

The first algorithms for (r, c) -NN in high dimensions were obtained by using the technique of random projections. This technique is applicable if the underlying metric D is induced by an l_p norm, for $p \in [0, 2]$. We first focus on the case where all input and query points are binary vectors from $\{0, 1\}^d$, and D is the Hamming distance (or equivalently, the metric is induced by the l_1 norm). The parameters of the algorithms discovered for this case are presented in Table 39.1.1.

We mention that the idea of using random projections for high-dimensional approximate nearest neighbor first appeared in the paper by Kleinberg [Kle97]. Although his algorithms still suffered from the curse of dimensionality (i.e., used exponential storage or had $\Omega(n)$ query time), his ideas provided inspiration for designing improved algorithms.

Dimensionality reduction. The key technique used to obtain results (1a), (1b), (3), and (4) is **dimensionality reduction**, i.e., a randomized procedure which reduces the dimension of Hamming space from d to $k = O(\log n / \epsilon^2)$, while preserving a certain range of distances between the input points and the query up to a factor of $1 + \epsilon$. This notion has been introduced earlier in Chapter 8 in the context of Euclidean space. In case of Hamming space, [KOR00] showed the following.

THEOREM 39.1.1

For any given $r \in \{1 \dots d\}$, $\epsilon \in (0, 1]$ and $P \in (0, 1)$, one can construct a distribution over mappings $A : \{0, 1\}^d \rightarrow \{0, 1\}^k$, $k = O(\log(1/P) / \epsilon^2)$, and a “scaling factor” S , so that for any $p, q \in \{0, 1\}^d$, if $D(p, q) \in [r, 10r]$, then $D(A(p), A(q)) = S \cdot D(p, q)(1 \pm \epsilon)$ with probability at least $1 - P$.

The factor 10 can be replaced by any constant. As in the case of Euclidean norm, the mapping A is linear. However, unlike in the Euclidean case where the mapping was defined over the set of reals \mathbb{R} , the mapping A is defined over $GF(2)$

(i.e., over the set $\{0, 1\}$ with addition and multiplication taken modulo 2). The $k \times n$ matrix A is obtained by choosing each entry of A independently at random from the set $\{0, 1\}$. The probability that an entry is equal to 1 is roughly r/d .

A different method of generating mapping A was proposed in [Ind00]. The mapping is nonlinear, but somewhat easier to analyze (and derandomize). It is based on “Locality-Sensitive Hashing,” described later in this section.

Algorithm (1a) is an immediate application of Theorem 39.1.1. Specifically, it allows us to reduce the $(r, c + \epsilon)$ -NN problem in d -dimensional space to (r, c) -NN problem in k -dimensional space. Since the *exact* nearest-neighbor problem in k -dimensional space can be solved by storing the answers to all 2^k queries q , the bound follows. Algorithm (1b) is follows by using a variation of this approach. Algorithms (3) and (4) are obtained by using a deterministic version of Theorem 39.1.1 [Ind00].

We note that one can apply the same approach to solve the near-neighbor problem in the Euclidean space. In particular, it is fairly easy to solve the $(r, 1 + \epsilon)$ -NN problem in l_2^d using $n(1/\epsilon)^{O(d)}$ space [HIM03]. Applying the Johnson-Lindenstrauss lemma leads to an algorithm with storage bound similar (although slightly worse) to the bound of algorithm (1a) [HIM03].

Locality-Sensitive Hashing. As may have been noticed, the storage bounds for algorithms (1a), (3) and (4) are quite high. On the other hand, the query time of algorithm (1b) is low only for fairly large values of ϵ [Ind01a]. In this context, algorithm (2) provides an attractive tradeoff, since even for small values of ϵ (e.g., $\epsilon = 1.0$) its running time is fairly low (e.g., $d\sqrt{n}$). The algorithm is based on the concept of **Locality-Sensitive Hashing**, or **LSH** [HIM03] (see also [K⁺95, Bro00]). A family of hash functions $h : \{0, 1\}^d \rightarrow U$ is called (r_1, r_2, P_1, P_2) -sensitive (for $r_1 < r_2$ and $P_1 > P_2$) if for any $q, p \in \{0, 1\}^d$

- If $D(p, q) \leq r_1$ then $\Pr[h(q) = h(p)] \geq P_1$,
- If $D(p, q) > r_2$ then $\Pr[h(q) = h(p)] \leq P_2$

where $\Pr[\cdot]$ is defined over the random choice of h . We note that the notion of locality-sensitive hashing can be defined for any metric space D in a natural way (see [Cha02] for sufficient and necessary conditions for existence of LSH for D). However, for Hamming space, LSH families are particularly easy: it is sufficient to take all functions h_i , $i = 1 \dots d$, such that $h_i(p) = p_i$, $p \in \{0, 1\}^d$. Because $\Pr[h(p) = h(q)] = 1 - D(p, q)/d$, it is immediate that this family is sensitive.

If we are provided with an LSH family with a “large” gap between P_1 and P_2 , the $(r_2/r_1, r_1)$ -NN problem can be solved in the following way. During preprocessing, all input points p are hashed to the bucket $h(p)$. In order to answer the query q , the algorithm retrieves the points in the bucket $h(q)$ and checks if any one of them is close to q . If the gap between P_1 and P_2 is sufficiently large, this approach can be shown to result in sublinear query time. Unfortunately, the P_1/P_2 gap guaranteed by the above LSH family is not large enough. However, the gap can be amplified by concatenating several independently chosen hash functions $h_1 \dots h_l$ (i.e., hashing the points using functions h' such that $h'(p) = (h_1(p), \dots, h_l(p))$). Details can be found in [HIM03].

A somewhat similar hashing-based algorithm (for the closest-pair problem) was earlier proposed in [K⁺95], and also in [Bro00]. Due to different problem formulation and analysis, comparing their performance with the guarantees of the

LSH approach seems difficult.

We also mention that the above algorithm can be modified to solve the approximate *nearest*-neighbor problem, within the same time bounds (i.e., without incurring any additional overhead, as is the case for the reductions presented in the next section). Details can be found in [Cha02].

Extensions to l_p norms. The approximate near-neighbor problem under l_p norms, for $p \in [1, 2]$, can be reduced to the same problem in Hamming space. The reduction is particularly easy for the l_1^d norm. If we assume that all points of interest p have coordinates in the range $\{1 \dots M\}$, then if we define $U(p) = (U(p_1), \dots, U(p_d))$ where $U(x)$ is a string of x ones followed by $M - x$ zeros, we get $\|p - q\|_1 = D(U(p), U(q))$. In general, M could be quite large, but can be reduced to $d^{O(1)}$ in the context of approximate near neighbor [HIM03]. Thus we can reduce (r, c) -NN under l_1 to (r, c) -NN in Hamming space.

In order to obtain algorithms for l_p norm where $p \in (1, 2]$, we use the fact that l_p^d can be embedded into $l_1^{O(d)}$ with bounded distortion (see [Chapter 8](#)). Alternatively, for $p = 2$, one can solve the problem directly in Euclidean space [HIM03], as described earlier.

DIVIDE-AND-CONQUER APPROACH

The dimensionality reduction and locality-sensitive hashing techniques have natural limitations. In particular, they cannot be used for solving the near-neighbor problem under the l_∞ norm. Fortunately, this norm has other nice properties which makes designing approximate nearest-neighbor data structures possible.

The only algorithm known for solving (r, c) -NN under the l_∞^d norm [Ind01b] has the following parameters, for any $\rho > 0$:

- Approximation factor: $c = O(4\lfloor \log_{1+\rho} \log 4d \rfloor)$; if $\rho = \log d$ then $c = 3$
- Space: $dn^{1+\rho}$
- Query time: $O(d \log n)$ for the static, or $(d + \log n)^{O(1)}$ for the dynamic case
- Update time: $d^{O(1)}n^\rho$ (described in [Ind01a])

The basic idea of the algorithm is to use a divide and conquer approach. In particular, consider hyperplanes H consisting of all points with one (say the i th) coordinate equal to the same value. The algorithm tries to find a hyperplane H having the property that the set of points $P_L \subset P$ which are on the left side of H and within distance $\geq r$ from H , is not “much smaller” than the set P_M of points within distance r from H . Moreover, a similar condition has to be satisfied for an analogously defined set P_R of points on the right side of H . If such H exists, we divide P into $P_1 = P_L \cup P_M$ and set $P_2 = P \setminus P_L$ and build the data structure recursively on P_1 and P_2 . It is easy to see that while processing a query q , it suffices to recurse on either P_1 or P_2 , depending on the side of H the query q lies on. Also, one can prove that the increase in storage caused by duplicating P_M is moderate. On the other hand, if H does not exist, one can prove that a large subset C of P has $O(r)$ diameter. In such a case we can pick any point from C as its representative, and apply the algorithm recursively on $P \setminus C$.

GLOSSARY

Product metrics: An f -product of metrics M_1, \dots, M_k with distance functions D_1, \dots, D_k is a metric over $M_1 \times \dots \times M_k$ with distance function D such that $D((p_1, \dots, p_k), (q_1, \dots, q_k)) = f(D_1(p_1, q_1), \dots, D_k(p_k, q_k))$.

Although the l_∞ data structure seems to rely on the geometry of the l_∞ norm, it turns out that it can be used in a much more general setting. In particular, assume that we are given k metrics $M_1 \dots M_k$ such that for each metric M_i we have a data structure for (a variant of) (r, c) -NN in metric M_i , with $Q(n)$ query time and $S(n)$ space. In this setting, it is possible to construct a data structure solving $(r, O(c \log \log n))$ -NN in the max-product metric M of M_1, \dots, M_k (i.e., an f -product with f computing the maximum of its arguments) [Ind02]. The data structure for M achieves query time roughly $O(Q(n) \log n + k \log n)$ and space $O(kS(n)n^{1+\delta})$, for any constant $\delta > 0$. The data structure could be viewed as an abstract version of the data structure for the l_∞ norm (note that the l_∞^d norm is a max-product of l_p^1 norms). For the particular case of the l_∞^d norm, it is easy to verify that the result of [Ind02] provides a $O(\log \log n)$ -approximate algorithm using space polynomial in n . At the same time, the algorithm of [Ind01b] has $O(\log \log d)$ -approximation guarantee when using the same amount of space. Interestingly, the former data structure gives an approximation bound comparable to the latter one, while being applicable in a much more general setting.

EXTENSIONS VIA EMBEDDINGS

Most of the algorithms described so far work only for l_p norms. However, they can be used for other metric spaces M , by using low-distortion embeddings of M into l_p norms. See [Chapter 8](#) for more information.

AVERAGE-CASE ALGORITHMS

The approximate algorithms described so far are designed to work for any (i.e., worst-case) input. However, researchers have also investigated *exact* algorithms for the NN problem, which achieve fast query times for *average* input. Below we describe three such results.

Near-neighbor in Hamming space. Consider the point set P where each point is chosen independently and uniformly at random from the set $\{0, 1\}^d$. In addition, assume that the nearest neighbor p of the query point q is located within distance r from q . In this setting, it was shown in [GP⁺94] that q can be retrieved in $O(dn^{r/d})$ time, using a data structure which requires $O(dn^{1+r/d})$ space. The basic idea of their approach is similar to the locality-sensitive hashing approach of [HIM03]; however, the set of projected coordinates is chosen in a deterministic fashion, to optimize certain parameters.

Nearest neighbor in the l_2^d norm. Consider the “continuous” version of the Hamming distance scenario, such that each point in P is chosen independently and uniformly at random from the set $[-1, 1]^d$. In addition, assume that the nearest

neighbor p of the query point q is located within distance $r = 2b\sqrt{d}$ for some (small) constant b . The value of b is always small enough so that r does not exceed the average distance between two random points.

Under these assumptions, it was shown in [Yia00] that the k - d -tree data structure (augmented in a proper way) enjoys $O(dn^\rho)$ query time, where ρ is a function of b . The analysis in the paper is idealized (i.e., uses approximations not shown to be rigorous).

We note that if d is large enough, then the distance between the query point and any data point is sharply concentrated around its mean (say $2t\sqrt{d}$). In this case, if $r = 2bt\sqrt{d}$, $b \in (0, 1)$, then by using locality-sensitive hashing with approximation factor $1/b$, one obtains an algorithm with query time dn^b . It appears that this bound outperforms the computational bound given in [Yia00]. However, the k - d -tree data structure used in [Yia00] uses only linear space, unlike the LSH-based approach.

Nearest neighbor in the l_∞^d norm. Consider a point set generated as before, but with the query point generated from the same distribution as the input points (and independently from the latter). In this setting, it was shown [AHL01, HL02] that there is a nearest-neighbor data structure using $O(dn)$ space, with query time $O(n \log d)$. Note that a naive algorithm would suffer from query time of $O(nd)$. The algorithm uses a clever pruning approach to quickly eliminate points that *cannot* be nearest neighbors of the query point.

39.2 REDUCTIONS TO APPROXIMATE NEAR NEIGHBOR

GLOSSARY

We define the following problems, for a given set of points P in a metric space $M = (X, D)$:

Approximate Closest Pair, or c -CP: Find a pair of points $p', q' \in P$ such that $D(p', q') \leq c \min_{p,q \in P, p \neq q} D(p, q)$

Approximate Close Pair, or (r, c) -CP: If there exists $p, q \in P, p \neq q$, such that $D(p, q) \leq r$, find a pair $p', q' \in P, p' \neq q'$, such that $D(q', p') \leq cr$.

Approximate Chromatic Closest Pair, or c -CCP: Assume that each point $p \in P$ is labeled with a color $c(p)$. The goal is to find a pair of points p, q such that $c(p) \neq c(q)$ and $D(p, q)$ is approximately minimal (as in the definition of c -CP).

Approximate Bichromatic Closest Pair, or c -BCP: As above, but $c(p)$ assumes only two values.

Approximate Chromatic/Bichromatic Close Pair, or (r, c) -CCP/ (r, c) -BCP: Decision versions of c -CCP or c -BCP (as in the definition of (r, c) -CP).

Approximate Furthest Pair, or Diameter, or c -FP: Find $p, q \in P$ such that $D(p, q) \geq \max_{p', q' \in P} D(p', q')/c$. The decision problem, called **Approximate Far Pair**, or (r, c) -FP, is defined in the natural way.

approximate Furthest Neighbor, or c -FN: A maximization version of the Approximate Near Neighbor. The decision problem, called Approximate Far Neigh-

bor or (r, c) -FN, is defined in a natural way.

Approximate Minimum Spanning Tree, or c -MST: Find a tree T spanning all points in P whose weight $w(T) = \sum_{(p,q) \in T} D(p, q)$ is at most c times larger than the weight of any tree spanning P .

approximate Bottleneck Matching, or c -BM: Assuming $|P|$ is even, find a set of $|P|/2$ non-incident edges E joining points in P (i.e., a matching), such that the following function is minimized (up to factor of c)

$$\max_{\{p,q\} \in E} D(p, q)$$

Approximate Facility Location, or c -FL: Find a set $F \subset P$ such that the following function is minimized (up to factor of c), given the cost function $c : P \rightarrow \mathbb{R}^+$

$$\sum_{p \in F} c(p) + \sum_{p \in P} \min_{f \in F} D(p, f)$$

In general, we could have two sets: P_c of *cities* and P_f of *facilities*; in this case we require that $F \subset P_f$ and we are only interested in the cost of P_c .

Spread (of a point set): The ratio between the diameter of the set to the distance between its closest pair of points.

In this section we show that the problems defined above can be efficiently reduced to the approximate near-neighbor problem discussed in the previous section.

First, we observe that any problem from the above list, say $c(1 + \delta)$ -P for some $\delta > 0$, can be easily reduced to its decision version (say (r, c) -P), if we assume that the spread of $P \cup \{q\}$ is always bounded by some value, say Δ . For simplicity, assume that the minimum distance between the points in P is 1. The reduction proceeds by building (or maintaining) $O(\log_{1+\delta} \Delta)$ data structures for (r, c) -P, where r takes values $(1 + \delta)^i / 2$ for $i = 0, 1, \dots$. It is not difficult to see that a query to $c(1 + \delta)$ -P can be answered by $O(\log \log_{1+\delta} \Delta)$ calls to these structures for (r, c) -P, via binary search.

In general, the spread of P could be unbounded. However, in many cases it is easy to ensure that $\Delta \leq n^{O(1)}$. This can be accomplished, for example, by “discretizing” the input to c -MST or c -FL. In those cases, the above reduction is very efficient.

Reductions from other problems are specified in the following table. The bounds for the time and space used by the algorithm in the “To” column are denoted by $T(n)$ and $S(n)$, respectively.

We mention that a few other reductions have been given in [KOR00, B⁺99b]. For the problems discussed in this section, they are less efficient than the reductions in the above table. Additionally, [B⁺99b] reduces the problems of computing *approximate agglomerative clustering* and *sparse partitions* to $O(n \log^{O(1)} n)$ calls to a dynamic approximate nearest-neighbor data structure. See [B⁺99b] for the definitions and algorithms.

Also, we mention that a reduction from $(1 + \epsilon)$ -approximate furthest neighbor to $(1 + \epsilon)$ -approximate nearest neighbor (for the static case and under the l_2 norm) has been given in [GIV01]. However, a direct (and dynamic) algorithm for the approximate furthest neighbor in l_2^d , achieving a better query and update times of $dn^{1/(1+\epsilon)^2}$, has been recently given in [Ind03]. The former paper also presents an

TABLE 39.2.1 Reductions to Approximate Near Neighbors.

#	FROM	TO	TIME	SPACE
1	Source: [HIM03].			
	$c(1 + \delta)$ -NN	(r, c) -NN	$T(n) \log^{O(1)} n$	$S(n) \log^{O(1)} n$
2	Source: [Epp95]; amortized time.			
	c -DBCP (r, c) -DBCP	c -DNN (r, c) -DNN	$T(n) \log^{O(1)} n$ $T(n) \log^{O(1)} n$	$S(n) \log^{O(1)} n$ $S(n) \log^{O(1)} n$
3	Source: [HIM03]; via Kruskal alg.			
	$c(1 + \delta)$ -MST	(r, c) -DBCP	$nT(n) \log^{O(1)} n$	
4	Source: [GIV01, Ind01a]; via Primal-Dual			
	$3c^3(1 + \delta)$ -FL	(r, c) -DBCP	$nT(n) \log^{O(1)} n$	
5	Source: [GIV01, Ind01a].			
	$2c$ -BM	c -DBCP	$nT(n) \log^{O(1)} n$	

algorithm for computing a $\sqrt{2} + \epsilon$ -approximate diameter (for any $\epsilon > 0$) of a given pointset in $dn \log^{O(1)} n$ time.

We now describe briefly the main techniques used to achieve the above results.

Nearest neighbor. We start from the reduction of c -NN to (r, c) -NN. As we have seen already, the reduction is easy if the spread of P is small. Otherwise, it is shown that the data set can be clustered into $n/2$ clusters, in such a way that:

- If the query point q is “close” to one of the clusters, it must be far away from a constant fraction of points in P ; thus, we can ignore these points in the search for an approximate nearest neighbor.
- If the query point q is “far” from a cluster, then all points in the clusters are equally good candidates for the *approximate* nearest neighbor; thus we can replace the cluster by its representative point.

These ideas were originally introduced in [IM98], but their data structure was quite complex and inefficient. In [HP01] Har-Peled presented a considerably simpler data structure, achieving better time and space bounds.

Bichromatic closest pair. A very powerful reduction from various variants of c -DBCP to c -DNN was given in [Epp95]. His algorithm was originally designed for the case $c = 1$, but it can be verified to work also for general $c \geq 1$ [Epp99]. Moreover, as mentioned in the original paper, the reduction does not require the distance function $D()$ be a metric.

The basic idea of the algorithm is to try to maintain a graph that contains an edge connecting the two closest bichromatic points. A natural candidate for such a graph is the graph formed by connecting each point to its nearest neighbor. This, however, does not work, because a vertex in such a graph can have very high degree, leading to high update cost. Another option would be to maintain a single path, such that the i th vertex points to its nearest neighbor of the opposite color, chosen from points not yet included in the path. This graph has low degree, but its rigid

structure makes it difficult to update it at each step. So the actual data structure is based on the path idea but allows its structure to degrade in a controlled way, and only rebuilds it when it gets too far degraded, so that the rebuilding work is spread over many updates. Then, however, one needs to keep track of the information from the degraded parts of the path, which can be done using a second shorter path, and so on. The constant factor reduction in the lengths of each successive path means the total number of paths is only logarithmic.

Minimum spanning tree. Many existing algorithms for computing MST (e.g., Kruskal's algorithm) can be expressed as a sequence of operations on a CCP data structure. For example, Kruskal's algorithm repetitively seeks the lightest edge whose endpoints belong to different components, and then merges the components. These operations can be easily expressed as operations on a CCP data structure, where each component has a different color. The contribution of [HIM03] was to show that in case of Kruskal's algorithm, using an *approximate* c -CCP data structure enables one to compute an *approximate* c -MST. Also, note that c -CCP can be implemented by $\log n$ c -BCP data structures [HIM03]. Other reductions from c -MST to c -BCP are given in [B⁺99b, IST99].

Minimum bottleneck matching. The main observation behind this algorithm is that a matching is also a spanning forest with the property that any connected component has even cardinality (call it an *even* forest). At the same time, it is possible to convert *any* even forest to a matching, in a way that increases the length of the longest edge by at most a factor of 2. Thus, it suffices to find an even forest with minimum edge length. This can be done by including longer and longer edges to the graph, and stopping at the moment when all components have even cardinality. It is not difficult to implement this procedure as a sequence of c -CCP (or c -BCP) calls.

Other algorithms. The algorithm for the remaining problem (c -FL) is obtained by implementing the primal-dual approximation algorithm [JV99]. Intuitively, the algorithm proceeds by maintaining a set of balls of increasing radii. The latter process can be implemented by resorting to c -CCP. The approximation factor follows from the analysis of the original algorithm.

39.3 LOWER BOUNDS

In the previous sections we presented many algorithms solving approximate versions of proximity problems. The main motivation for designing approximation algorithms was the “curse of dimensionality” conjecture, i.e., the conjecture that finding exact solutions to those problems requires either superpolynomial (in d) query time, or superpolynomial (in n) space. In this section we state the conjecture more rigorously, and describe the progress toward proving it.

We start from the exact near-neighbor problem. For this problem, the curse of dimensionality can be formalized as follows. Assume that $d = n^{o(1)}$, but $d = \omega(\log n)$.

Conjecture 1 Any data structure for $(r, 1)$ -NN in Hamming space over $\{0, 1\}^d$, with $d^{O(1)}$ query time, must use $n^{\omega(1)}$ space.

The conjecture as stated above is probably the weakest version of the “curse of dimensionality” phenomenon for the near-neighbor problem. It is plausible that other (stronger) versions of the conjecture could hold. In particular, at present, we do not know any data structure which simultaneously achieves $o(dn)$ query time and $2^{o(d)}$ space for the above range of d . At the same time, achieving $O(dn)$ query time with space dn , or $O(d)$ query time with space 2^d is quite simple (via linear scan or using exhaustive storage).

Also note that if $d = O(\log n)$, achieving $2^{o(d)} = o(n)$ space is impossible via a simple incompressibility argument.

Below we describe the work toward proving the conjecture. The first result addresses the complexity of a simpler problem, namely the *partial match* problem. This problem is of importance in databases and other areas and has been long investigated (e.g., see [Riv74]). Thus, the lower bounds for this problem are interesting in their own right.

GLOSSARY

Partial match: Given a set P of n vectors from $\{0, 1\}^d$, design a data structure that supports the following operation: For any query $q \in \{0, 1, *\}^d$, check if there exists $p \in P$ such that for all $i = 1 \dots d$, if $q_i \neq *$ then $p_i = q_i$.

It is not difficult to see that any data structure solving $(r, 1)$ -NN in the Hamming metric $\{0, 1\}^d$, can be used to solve the partial match problem using essentially the same space and query time. Thus, any lower bound for partial match problem implies a corresponding lower bound for the near-neighbor problem. The best currently known lower bound for the partial match has been established in [B⁺99a], following earlier work of [M⁺94]. Their lower bound holds in the *cell-probe* model, a very general model of computation, capturing e.g., the standard Random Access Machine model. Specifically, they show that any (possibly randomized) cell-probe algorithm for the partial match problem, in which the algorithm is allowed to retrieve at most $O(n^{1-\epsilon})$ bits from any memory cell in one step for $\epsilon > 0$, must either have $\Omega(\log d)$ query time or use $n^{\Omega(\log d)}$ memory cells.

For the exact near-neighbor problem, an exponentially larger bound was given in [BR00]. They showed that any (possibly randomized) cell-probe algorithm for $(r, 1)$ -NN in d -dimensional Hamming space, with cell size restriction as above, must either have query time $> t$, or use $2^{\Omega(d/t)}$ space. Thus, if $t = o(d/\log n)$, the space used must be superpolynomial in n .

The two aforementioned lower bounds are proved in a very general model, using the tools of *communication complexity*. As a result, they cannot yield lower bounds of $\omega(d/\log n)$ for the required query time, assuming $n^{\Theta(1)}$ space, as we now explain.

The communication complexity approach interprets the data structure as a communication channel between Alice (holding the query point q) and Bob (holding the database P). The goal of the communication (for Alice) is to learn the nearest neighbor of q . Since the data structure has polynomial size, each access to one of its memory cell is equivalent to Alice sending $O(\log n)$ bits of information to Bob. If we show that Alice needs to send at least b bits to Bob to solve the problem, we obtain $\Omega(b/\log n)$ lower bound for the query time. However, $b \leq d$, since Alice can always choose to transmit the whole input vector q . Thus, $\Omega(d/\log n)$ lower bound is the best result one can achieve using the communication complexity approach. A partial step toward removing this obstacle was made in [BV02], em-

ploying the *branching programs* model of computation. In particular, they focused on randomized algorithms that have very small (inversely polynomial in n) probability of error. They showed that any algorithm for the $(r, 1)$ -NN problem in the Hamming metric over $\{1 \dots d^6\}^d$, has either $\Omega(d \log(d \log d/S))$ query time or uses $\Omega(S)$ space. This holds for $n = \Omega(d^6)$. Thus, if the query time is $o(d \log d)$, then the data structure must use $2^{d^{\Omega(1)}}$ space.

This completes the survey of lower bounds for the *exact* near-neighbor search. For the approximate version of this problem a cell-probe-based lower bound was shown in [CC⁺99]. Specifically, the authors show that any deterministic data structure for the c -approximate nearest neighbor $\{0, 1\}^d$ requires either $\Omega(\log \log d / \log \log \log d)$ query time, or use $n^{\omega(1)}$ space. They assume that a memory cell can contain up to $d^{O(1)}$ bits accessible in one step. Moreover, the approximation factor c can be as high as $2^{(\log d)^{1-\epsilon}}$ for any $\epsilon > 0$.

For comparison, if randomization is allowed, then by using Theorem 39.1.1 combined with binary search one can get a data structure for the same problem (for any fixed $c > 1$), with polynomial size and query time $O(\log \log_c d)$. Note that the assumption $c > 1$ is crucial for those algorithms to achieve polynomial space bound.

REDUCTIONS

Despite the recent progress toward resolving the “curse of dimensionality” conjecture and the widespread belief in its validity, proving it seems currently beyond reach. Nevertheless, it is natural to assume the validity of the conjecture (or its variants), and see what conclusions can be derived from this assumption. Below we survey a few results of this type.

In order to describe the results, we need to state another conjecture.

Conjecture 2 *Let $d = n^{o(1)}$ but $d = \log^{\omega(1)} n$. Any data structure for the partial match problem with parameters d and n which provides $d^{O(1)}$ query time must use $2^{d^{\Omega(1)}}$ space.*

Note that, for the same ranges² of d , Conjecture 2 is analogous to Conjecture 1, but much stronger: it considers an easier problem, and states stronger bounds. However, since the partial match problem was extensively investigated on its own, and no algorithm with bounds remotely resembling the above have been discovered (cf. [CIP02] for a survey), Conjecture 2 is believed to be true.

Assuming Conjecture 2, it is possible to show lower bounds for some of the approximate nearest-neighbor problems discussed in Section 39.1. In particular, it was shown [Ind01b] that any data structure for (r, c) -NN under l_∞^d for $c < 3$ can be used to solve the partial match problem with parameter d , using essentially the same query time and storage (the number of points in the database is the same in both cases). Thus, unless Conjecture 2 is false, the 3-approximation algorithm from Section 39.1 is optimal, in the sense that it provides the smallest approximation factor possible while preserving polynomial (in d) query time and subexponential (in d) storage. Note that this result resembles the non-approximability results based on the $P \neq NP$ conjecture.

On the other hand, it was shown [CIP02] that the exact near-neighbor problem

²For $d = \log^{O(1)} n$, Conjecture 2 is true by a simple incompressibility argument. At the same time, the status of Conjecture 1 for $d \in [\omega(\log n), \log^{O(1)} n]$ is still unresolved.

under the l_∞^k norm can be reduced to solving the partial match problem with the parameter $d = (k + \log n)^{O(1)}$; the number of points n is the same for both problems. In fact, the same holds for a more general problem of *orthogonal range queries*. Thus, Conjecture 2, and its variant for the $(r, 1)$ -NN under l_∞^d (or for orthogonal range queries), are equivalent. This strengthens the belief in the validity of Conjecture 2, since the exact nearest neighbor under l_∞ norm and the orthogonal range query problem received additional attention in the Computational Geometry community.

39.4 LOW VS. HIGH DIMENSIONS IN COMPUTATIONAL GEOMETRY

It is apparent that nearest neighbors and related problems in high dimensions enjoy properties quite different from their low-dimensional counterparts (see [Chapter 51](#)). Among the main differences are:

- Exact computation seems (and is conjectured to be) intractable in high dimensions; on the other hand, very efficient algorithms exist in low-dimensional cases.
- The core problem that seems to capture the computational difficulty is the near-neighbor problem in Hamming space $\{0, 1\}^d$, a problem trivial for constant dimension.
- Unlike the low-dimensional case, the tools of *combinatorial geometry* are rarely used to design or analyze algorithms in high dimensions. This phenomenon seems to reflect the fact that the typical tools (such as complexity of arrangements, or packing bounds) lead to exponential algorithmic complexity. Instead, tools from functional analysis (most notably embeddings) are used.

Nevertheless, there seem to be interesting connections between low and high dimensional scenarios. For example, the key component of several reductions given in Section 39.2 is the result of Eppstein [Epp95]. His algorithm was originally developed with low-dimensional applications in mind; however, its framework was sufficiently general to be useful in the high-dimensional case as well.

As an example of impact in the other direction, one could mention the nearest-to-near neighbor reduction of [IM98]. When applied in the low-dimensional case, their result gave the first algorithm for $(1 + \epsilon)$ -approximate nearest neighbor, with polynomial space and polylogarithmic query time, for dimension d up to $O(\log n)$ (earlier results could provide that bound only for $d = O(\log \log n)$, due to exponential dependence of the query time on the dimension). These results were further refined in the low-dimensional context in [HP01, AM02], yielding an efficient approximate nearest-neighbor data structure for low dimensions.

Finally, we mention an example of a fruitful marriage between low- and high-dimensional techniques. Consider the following problem. For a constant d , assume we are given n $(d-1)$ -dimensional flats $H_1 \dots H_n$ living in \mathbb{R}^d , as well as a set P of n points P in \mathbb{R}^d . The goal is to compute a tree spanning the points in P , such that the total number of times a tree edge crosses a flat is as small as possible.

In [HPI00], the authors provided a c -approximate algorithm for this problem, with running time $O(n^{2d/(d+1)+\delta} + n^{1+1/c} \log^{O(1)} n)$, for any $\delta > 0$ (the factors

polynomial in $1/(c - 1)$ are omitted). Note that this time is subquadratic for any constant d and $c > 1$. The main idea of the algorithm is to observe that the number of flats crossed on the way from point p to p' is a metric, and moreover, this metric can be isometrically embedded into n -dimensional Hamming space. This allows one to use the high-dimensional approximate MST algorithms from Section 39.2. To make that algorithm run fast, one needs to perform the dimensionality reduction before computing MST (essentially as in Theorem 39.1.1). However, just computing the n -dimensional representation of each of n points in P requires $\Omega(n^2)$ time. To avoid this bottleneck, the dimensionality reduction is performed on “implicit” n -dimensional representations of the points in P , by using the partition trees of Matoušek.

RELATED CHAPTERS

- [Chapter 8: Low-distortion embeddings of discrete metric spaces](#)
- [Chapter 24: Arrangements](#)
- [Chapter 36: Range searching](#)
- [Chapter 51: Pattern Recognition](#)

REFERENCES

- [AHL01] H. Alt and L. Heinrich-Litan. Exact l_∞ -nearest neighbor search in high dimensions. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 157–163, 2001.
- [AM02] S. Arya and T. Malamatos. Linear-size approximate Voronoi diagrams. *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 147–155, 2002.
- [B⁺99a] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 312–321, 1999.
- [B⁺99b] A. Borodin, R. Ostrovsky, and Y. Rabani. Subquadratic approximation algorithms for clustering problems in high dimensional spaces. *Proc. 31st Annu. ACM Sympos. Theory Comput.*, pages 435–444, 1999.
- [BR00] O. Barkol and Y. Rabani. Tighter bounds for nearest neighbor search and related problems in the cell probe model. *Proc. 32nd Annu. ACM Sympos. Theory Comput.*, pages 388–396, 2000.
- [Bro00] A. Broder. Identifying and filtering near-duplicate documents. In *Proc. 11th Annu. Sympos. Combin. Pattern Matching*, volume 1848 of *Lecture Notes Comput. Sci.*, pages 1–10, Springer-Verlag, Berlin, 2000.
- [BT01] J. Buhler and M. Tompa. Finding motifs using random projections. *Proc. Annu. Internat. Conf. Comput. Molec. Biology*, pages 69–76, 2001.
- [Buh01] J. Buhler. Efficient large-scale sequence comparison by locality-sensitive hashing. *Bioinformatics*, 17:419–428, 2001.
- [Buh02] J. Buhler. Provably sensitive indexing strategies for biosequence similarity search. *Proc. Annu. Internat. Conf. Comput. Molec. Biology (RECOMB02)*, pages 90–99, 2002.
- [BV02] P. Beame and E. Vee. Time-space tradeoffs, multiparty communication complexity, and nearest-neighbor problems. *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 688–697, 2002.

- [CC⁺99] A. Chakrabarti, B. Chazelle, B. Gum, and A. Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the Hamming cube. *Proc. 31st Annu. ACM Sympoz. Theory Comput.*, pages 305–311, 1999.
- [CD⁺00] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J.D. Ullman, and C. Yang. Finding interesting associations without support pruning. *Proc. 16th Internat. Conf. Data Eng. (ICDE)*, pages 64–78, 2000.
- [Cha02] M. Charikar. Similarity estimation techniques from rounding. *Proc. 34th Annu. ACM Sympoz. Theory Comput.*, pages 380–388, 2002.
- [CIP02] M. Charikar, P. Indyk, and R. Panigrahy. New algorithms for subset query, partial match, orthogonal range searching and related problems. *Proc. Internat. Colloq. Automata Lang. Program.*, pages 451–462, 2002.
- [Cla88] K.L. Clarkson. A randomized algorithm for closest-point queries. *SIAM J. Comput.*, 17:830–847, 1988.
- [DW82] L. Devroye and T.J. Wagner. Nearest neighbor methods in discrimination. *Handbook of Statistics*, volume 2, P.R. Krishnaiah and L.N. Kanal, editors, Elsevier North-Holland, Amsterdam, 1982.
- [Epp95] D. Eppstein. Dynamic Euclidean minimum spanning trees and extrema of binary functions. *Discrete Comput. Geom.*, 13:111–122, 1995.
- [Epp99] D. Eppstein. Personal communication. 1999.
- [GG91] A. Gersho and R.M. Gray. *Vector Quantization and Data Compression*. Kluwer Acad., Boston, 1991.
- [GIM99] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *Proc. 25th Internat. Conf. Very Large Data Bases (VLDB)*, pages 518–529, 1999.
- [GIV01] A. Goel, P. Indyk, and K.R. Varadarajan. Reductions among high-dimensional geometric problems. *Proc. 12th ACM-SIAM Sympoz. Discrete Algorithms*, pages 769–778, 2001.
- [GP⁺94] D.H. Greene, M. Parnas, and F.F. Yao. Multi-index hashing for information retrieval. *Proc. 35th Annu. IEEE Sympoz. Found. Comput. Sci.*, pages 722–731, 1994.
- [GS⁺03] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. *Proc. 9th Internat. Conf. Comput. Vision*, pages 456–463, 2003.
- [GW97] M.X. Goemans and D.P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In *Approximation Algorithms for NP-Hard Problems*, PWS Publishing, Boston, pages 144–191, 1997.
- [HGI00] T. Haveliwala, A. Gionis, and P. Indyk. Scalable techniques for clustering the web. *WebDB Workshop*, pages 129–134, 2000.
- [HIM03] S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. Manuscript, 2003.
- [HL02] L. Heinrich-Litan. Exact l_∞ -nearest neighbor search in high dimensions. *Proc. 18th European Workshop Comput. Geom.*, pages 61–64, 2002.
- [HP01] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. *42th Annu. IEEE Sympoz. Found. Comput. Sci.*, pages 94–103, 2001.
- [HPI00] S. Har-Peled and P. Indyk. When crossings count—approximating the minimum spanning tree. *Proc. 18th Annu. ACM Sympoz. Comput. Geom.*, pages 166–175, 2000.
- [IM98] P. Indyk and R. Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. *Proc. 30th Annu. ACM Sympoz. Theory Comput.*, pages 604–613, 1998.

- [Ind00] P. Indyk. Dimensionality reduction techniques for proximity problems. *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 371–378, 2000.
- [Ind01a] P. Indyk. *High-dimensional computational geometry*. Ph.D. thesis, Dept. of Comput. Sci., Stanford Univ., 2001.
- [Ind01b] P. Indyk. On approximate nearest neighbors in l_∞ norm. *J. Comput. Syst. Sci.*, 63:627–638, 2001.
- [Ind02] P. Indyk. Approximate nearest neighbor algorithms for Frechet metric via product metrics. *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 102–106, 2002.
- [Ind03] P. Indyk. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 539–545, 2003.
- [IST99] P. Indyk, S.E. Schmidt, and M. Thorup. On reducing approximate MST to closest pair problems in high dimensions. Manuscript, 1999.
- [JV99] K. Jain and V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. *40th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 2–13, 1999.
- [Kle97] J. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. *Proc. 29th Annu. ACM Sympos. Theory Computing*, pages 599–608, 1997.
- [KOR00] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30:457–474, 2000.
- [K⁺95] R.M. Karp, O. Waarts, and G. Zweig. The bit vector intersection problem. *Proc. 36th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 621–630, 1995.
- [LT80] R.J. Lipton and R.E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9:615–627, 1980.
- [Mei93] S. Meiser. Point location in arrangements of hyperplanes. *Inform. Comput.*, 106:286–303, 1993.
- [M⁺94] P.B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *Proc. 26th Annu. ACM Sympos. Theory Comput.*, pages 103–111, 1994.
- [MP69] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, 1969.
- [O⁺02] Z. Ouyang, N. Memon, T. Suel, and D. Trendafilov. Cluster-Based Delta Compression of Collections of Files. *Proc. Internat. Conf. Web Inform. Sys. Eng. (WISE)*, pages 257–268, 2002.
- [Riv74] R.L. Rivest. *Analysis of Associative Retrieval Algorithms*. Ph.D. thesis, Dept. of Comput. Sci., Stanford Univ., 1974.
- [SH75] M.I. Shamos and D. Hoey. Closest point problems. *Proc. 16th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 152–162, 1975.
- [Shi00] N. Shivakumar. *Detecting digital copyright violations on the Internet*. Ph.D. thesis, Dept. of Comput. Sci., Stanford Univ., 2000.
- [W⁺98] R. Weber, H.J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. *Proc. 24th Internat. Conf. Very Large Data Bases (VLDB)*, pages 194–205, 1998.
- [Ya01] C. Yang. MACS: Music Audio Characteristic Sequence Indexing for Similarity Retrieval. *Proc. Workshop Appl. Signal Proc. Audio Acoustics*, pages 361–370, 2001.
- [Yia00] P.N. Yiannilos. Locally lifting the curse of dimensionality for nearest neighbor search. *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms*, pages 361–370, 2000.

40 RANDOMIZATION AND DERANDOMIZATION

Otfried Cheong, Ketan Mulmuley, and Edgar Ramos

INTRODUCTION

Randomized (or probabilistic) algorithms and constructions were applied successfully in many areas of theoretical computer science before they were used widely in computational geometry. Following influential work in the mid-1980s, randomized algorithms became popular in geometry, and now a significant proportion of published research in computational geometry employs randomized algorithms or proof techniques. For many problems the best algorithms known are randomized, and even if both randomized and deterministic algorithms of comparable asymptotic complexity are available, the randomized algorithms are often much simpler and more efficient in an actual implementation. In some cases, the best deterministic algorithm known for a problem has been obtained by “derandomizing” a randomized algorithm.

This chapter focuses on the randomized algorithmic *techniques* being used in computational geometry, and not so much on particular results obtained using these techniques. Efficient randomized algorithms for specific problems are discussed in the relevant chapters throughout this Handbook.

GLOSSARY

Probabilistic or “Monte Carlo” algorithm: Traditionally, any algorithm that uses random bits. Now often used in contrast to *randomized algorithm* to denote an algorithm that is allowed to return an incorrect or inaccurate result, or fail completely, but with small probability. Monte Carlo methods for numerical integration provide an example. Algorithms of this kind are not used frequently in computational geometry.

Randomized or “Las Vegas” algorithm: An algorithm that uses random bits and is guaranteed to produce a correct answer; its running time and space requirements may depend on random choices. Typically, one tries to bound the expected running time (or other resource requirements) of the algorithm. In this chapter, we will only consider randomized algorithms in this sense.

Expected running time: The expected value of the running time of the algorithm, that is, the average running time over all possible choices of the random bits used by the algorithm. No assumptions are made about the distribution of input objects in space. When expressing bounds as a function of the input size, the worst case over all inputs of that size is given. Normally the random choices made by the algorithm are hidden from the outside, in contrast with average running time.

Average running time: The average of the running time, over all possible inputs. Some suitable distribution of inputs is assumed.

To illustrate the difference between expected running time and average running

time, consider the *Quicksort* algorithm. If it is implemented so that the pivot element is the first element of the list (and the assumed input distribution is the set of all possible permutations of the input set), then it has $O(n \log n)$ *average* running time. By providing a suitable input (here, a sorted list), an adversary can force the algorithm to perform worse than the average. If, however, Quicksort is implemented so that the pivot element is chosen at random, then it has $O(n \log n)$ *expected* running time, for any possible input. Since the random choices are hidden, an adversary cannot force the algorithm to behave badly, although it may perform poorly with some positive probability.

Randomized divide-and-conquer: A divide-and-conquer algorithm that uses a random sample to partition the original problem into subproblems (Section 40.1).

Randomized incremental algorithm: An incremental algorithm where the order in which the objects are examined is a random permutation (Section 40.2).

Tail estimate: A bound on the probability that a random variable deviates from its expected value. Tail estimates for the running time of randomized algorithms are useful but seldom available (Section 40.10).

High-probability bound: A strong tail estimate, where the probability of deviating from the expected value decreases as a fast-growing function of the input size n . The exact definition varies between authors, but a typical example would be to ask that for any $\alpha > 0$, there exists a $\beta > 0$ such that the probability that the random variable $X(n)$ exceeds $\alpha E[X(n)]$ be at most $n^{-\beta}$.

Derandomization: Obtaining a deterministic algorithm by “simulating” a randomized one (Section 40.6).

Trapezoidal map: A planar subdivision $\mathcal{T}(S)$ induced by a set S of line segments with disjoint interiors in the plane (cf. Section 34.3). $\mathcal{T}(S)$ can be obtained by passing vertical attachments through every endpoint of the given segments, extending upward and downward until each hits another segment, or extending to infinity; see Figure 40.0.1. Every face of the subdivision is a trapezoid (possibly degenerated to a triangle, or with a missing top or bottom side), hence the name.

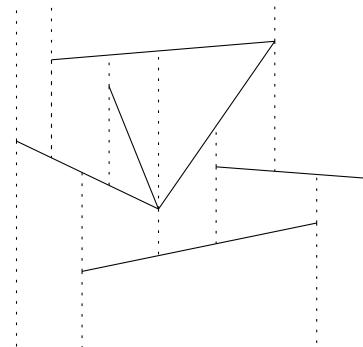


FIGURE 40.0.1

The trapezoidal map of a set of 6 line segments.

We will use the problem of computing the trapezoidal map of a set of line segments with disjoint interiors as a running example throughout this chapter. We assume for presentation simplicity that no two distinct endpoints have the same x -coordinate, so that every trapezoid is adjacent to at most four segments. (This can be achieved by slight rotation of the vertical direction.)

The trapezoidal map can also be defined for intersecting line segments. In that situation, vertical attachments must be added to intersection points as well,

and the map may consist of a quadratic number of trapezoids. The trapezoidal map is also called the *vertical decomposition* of the set of line segments. Decompositions similar to this play an important role in randomized algorithms, because most algorithms assume that the structure to be computed has been subdivided into elementary objects. (Section 40.5 explains why this assumption is necessary.)

40.1 RANDOMIZED DIVIDE-AND-CONQUER

GLOSSARY

Top-down sampling: Sampling with small, usually constant-size random samples, and recursing on the subproblems.

Cutting: A subdivision Ξ of space into simple cells Δ (of constant description complexity, most often simplices). The *size* of a cutting is the number of cells.

ϵ -cutting Ξ : For a set X of n geometric objects, a cutting such that every cell $\Delta \in \Xi$ intersects at most n/r of the objects in X (also called a $1/r$ -cutting with $\epsilon = 1/r$ when convenient). See also [Section 36.3](#).

Bottom-up sampling: Sampling with random samples large enough that the subproblems may be solved directly (without recursion).

Bernoulli sampling: The “standard” way of obtaining a random sample of size r from a given n -element set uses a random number generator to choose among all the possible subsets of size r , with equal probability for each subset (also obtained as the first r elements in a random permutation of n elements). In Bernoulli sampling, we instead toss a coin for each element of the set independently, and accept it as part of the sample with probability r/n . While the size of the sample may vary, its expected size is r , and essentially all the bounds and results of this chapter hold for both sampling models.

Gradation: A hierarchy of samples for a set X of objects obtained by bottom-up sampling:

$$X = X_1 \supset X_2 \supset X_3 \supset \cdots \supset X_{r-1} \supset X_r = \emptyset.$$

With Bernoulli sampling, a new element can be inserted into the gradation by flipping a coin at most r times, leading to efficient dynamic data structures ([Section 40.1](#)).

Geometric problems lend themselves to solution by divide-and-conquer algorithms. It is natural to solve a geometric problem by dividing space into regions (perhaps with a grid), and solving the problem in every region separately. When the geometric objects under consideration are distributed uniformly over the space, then gridding or “slice-and-dice” techniques seem to work well. However, when object density varies widely throughout the environment, then the decomposition has to be fine in the areas where objects are abundant, while it may be coarse in places with low object density. Random sampling can help achieve this: the density of a random sample R of the set of objects will approach that of the original set. Therefore dividing space according to the sample R will create small regions where the geometric objects are dense, and larger regions that are sparsely populated.

We can distinguish two main types of randomized divide-and-conquer algorithm, depending on whether the size of the sample is rather small or quite large.

TOP-DOWN SAMPLING

Top-down sampling is the most common form of random sampling in computational geometry. It uses a random sample of small, usually constant, size to partition the problem into subproblems. We sketch the technique by giving an algorithm for the computation of the trapezoidal map of a set of segments in the plane.

Given a set S of n line segments with disjoint (relative) interiors, we take a sample $R \subset S$ consisting of r segments, where r is a constant. We compute the trapezoidal map $\mathcal{T}(R)$ of R . It consists of $O(r)$ trapezoids. For every trapezoid $\Delta \in \mathcal{T}(R)$, we determine the ***conflict list*** S_Δ , the list of segments in S intersecting Δ . We construct the trapezoidal map of every set S_Δ recursively, clip it to the trapezoid Δ , and finally glue all these maps together to obtain $\mathcal{T}(S)$.

The running time of this algorithm can be analyzed as follows. Because r is a constant, we can afford to compute $\mathcal{T}(R)$ and the lists S_Δ naively, in time $O(r^2)$ and $O(nr)$ respectively. Gluing together the small maps can be done in time $O(n)$. But what about the recursive calls? If we denote the size of S_Δ by n_Δ , then bounding the n_Δ becomes the key issue here. It turns out that the right intuition is to assume that the n_Δ are about n/r . Assuming this, we get the recursion

$$T(n) \leq O(r^2 + nr) + O(r)T(n/r),$$

which solves to $T(n) = O(n^{1+\epsilon})$, where $\epsilon > 0$ is a constant depending on r . By increasing the value of r , ϵ can be made arbitrarily small, but at the same time the constant of proportionality hidden in the O -notation increases.

The truth is that one cannot really assume that $n_\Delta = O(n/r)$ holds for every trapezoid Δ at the same time. Valid bounds are as follows. For randomly chosen R of size r , we have:

- The ***pointwise bound***: With probability increasing with r ,

$$n_\Delta \leq C \frac{n}{r} \log r, \quad (40.1.1)$$

for all $\Delta \in \mathcal{T}(R)$, where the constant C does not depend on r and n .

- The ***higher-moments bound***: For any constant $c \geq 1$, there is a constant $C(c)$ (independent of r and n) such that

$$\sum_{\Delta \in \mathcal{T}(R)} (n_\Delta)^c = C(c) \left(\frac{n}{r} \right)^c |\mathcal{T}(R)|. \quad (40.1.2)$$

In other words, while the maximum n_Δ can be as much as $O((n/r) \log r)$, on the average the n_Δ behave as if they indeed were $O(n/r)$.

Both bounds can be used to prove that $T(n) = O(n^{1+\epsilon})$, with the dependence on ϵ being somewhat better using the latter bound. The difference between the two bounds becomes more marked for larger values of r , as will be detailed below. (For a more general result that subsumes these two bounds, see [Theorem 40.5.2](#).)

The same scheme used to compute $\mathcal{T}(S)$ will also give a data structure for point location in the trapezoidal map. This data structure is a tree, constructed as follows. If the set S is small enough, simply store $\mathcal{T}(S)$ explicitly. Otherwise,

take a random sample R , and store $\mathcal{T}(R)$ in the root node. Subtrees are created for every $\Delta \in \mathcal{T}(R)$. These subtrees are constructed recursively, using the sets S_Δ .

By the pointwise bound, the depth of the tree is $O(\log n)$ with high probability, and therefore the query time is also $O(\log n)$. The storage requirement is easily seen to be $O(n^{1+\epsilon})$ as above.

The algorithmic technique described in this section is surprisingly robust. It works for a large number of problems in computational geometry, and for many problems it is the only known approach to solve the problem. It does have two major drawbacks, however. First, it seems to be difficult to remove the ϵ -term in the exponent, and truly optimal random-sampling algorithms are scarce. Second, the practicality of this method remains to be established. If the size of the random sample is chosen too small, then the problem size may not decrease fast enough to guarantee a fast-running algorithm, or even termination. Few papers in the literature calculate this size constant, and so for most applications it remains unclear whether the size of the random sample can be chosen considerably smaller than the problem size in practice.

CUTTINGS

The only use of randomization in the above algorithm was to subdivide the plane into a number of simply-shaped regions Δ , such that every region is intersected by only a few line segments. Such a subdivision is called a *cutting* Ξ for the set X of n segments; if every $\Delta \in \Xi$ intersects at most n/r of the objects in X , it is a $1/r$ -cutting. Cuttings are interesting in their own right, and have been studied intensively. This research has led to a number of results on the deterministic construction of efficient cuttings, with useful properties that go beyond those of the simple cutting based on a random sample discussed above (Section 40.7). Cuttings form the basis for many algorithms and search structures in computational geometry; see [Section 36.2](#). As a result, most recent geometric divide-and-conquer algorithms no longer explicitly use randomization, and randomized divide-and-conquer is currently in the process of being replaced by divide-and-conquer based on cuttings.

In practice, however, cuttings may still be constructed most efficiently using random sampling. There are two basic techniques, which we illustrate again using a set X of n line segments with disjoint interiors in the plane.

- **ϵ -net based cuttings:** The easiest way to obtain a $1/r$ -cutting is to take a random sample $N \subset X$ of size $O(r \log r)$. If N is a $1/r$ -net for the range space (X, Γ) (defined in Section 40.4 and Section 36.2), then the trapezoidal map of N is a $1/r$ -cutting of size $O(r \log r)$. If not, we try a different sample.
- **Splitting the excess:** The construction based on ϵ -nets can be improved as follows. First take a random sample N of X of size $O(r)$, and compute its trapezoidal map. Every trapezoid Δ may be intersected by $O((n/r) \log r)$ segments. If we take a random sample of these segments, and form their trapezoidal map again (restricted to Δ), the pieces obtained are intersected by at most n/r segments. The size of this cutting is only $O(r)$, which is optimal.

Har-Peled [HP00] investigates the constants achievable for cuttings of lines in the plane.

BOTTOM-UP SAMPLING

In bottom-up sampling, the random sample is so large that the resulting subproblems are small enough to be solved directly. However, it is no longer trivial to compute the auxiliary structures needed to subdivide the problem. We again illustrate with the trapezoidal map.

Given a set S of n line segments, we take a sample R of size $n/2$, and compute the trapezoidal map of R recursively. For every $\Delta \in \mathcal{T}(R)$, we compute the list S_Δ of segments in $S \setminus R$ intersecting Δ . This can be done by locating an endpoint of every segment in $S \setminus R$ in $\mathcal{T}(R)$ and traversing $\mathcal{T}(R)$ from there. If we use a planar point location structure (Section 34.3), this takes time $O(n \log n + \sum_{\Delta \in \mathcal{T}(R)} n_\Delta)$. For every Δ , we then compute the trapezoidal map $\mathcal{T}(S_\Delta)$, and clip it to Δ . This can be done naively in time $O(n_\Delta^2)$. Finally, we glue together all the little maps.

The running time of the algorithm is bounded by the recursion

$$T(n) \leq T(n/2) + O(n \log n) + \sum_{\Delta \in \mathcal{T}(R)} O(n_\Delta^2).$$

The pointwise bound shows that with high probability, $n_\Delta = O(\log n)$ for all Δ . That would imply that the last term in the recursion is $O(n \log^2 n)$. Here, the higher-moments bound turns out to give a strictly better result, as it shows that the expected value of that term is only $O(n)$. The recursion therefore solves to $O(n \log^2 n)$.

Bottom-up sampling has the potential to lead to more efficient algorithms than top-down sampling, because it avoids the blow-up in problem size that manifests itself in the n^ϵ -term in top-down sampling. However, it needs more refined ingredients—as the constructions of $\mathcal{T}(R)$ and the lists S_Δ demonstrate—and therefore seems to apply to fewer problems.

As with top-down sampling, bottom-up sampling can be used for point location. These search structures have the advantage that they can often easily be made dynamic (Section 40.3).

40.2 RANDOMIZED INCREMENTAL ALGORITHMS

GLOSSARY

Backwards analysis: Analyzing the time complexity of an algorithm by viewing it running backwards in time [Sei93].

Conflict graph: A bipartite graph whose arcs represent conflicts (usually intersections) between objects to be added and objects already constructed.

History graph: A directed, acyclic graph that records the history of changes in the geometric structure being maintained. Also known as an *influence graph* or *I-DAG* (influence-directed acyclic graph).

Many problems in computational geometry permit a natural computation by an incremental algorithm. Incremental algorithms need only process one new object at a time, which often implies that changes in the geometric data structure remain localized in the neighborhood of the new object.

As an example, consider the computation of the trapezoidal map of a set of line segments (cf. Fig. 34.3.2; for another example, see [Section 22.3](#)). To add a new line segment s to the map, one would first identify the trapezoids of the map intersected by s . Those trapezoids must be split, creating new trapezoids, some of which then must be merged along the segment s . All these update operations can be accomplished in time linear in the sum of the number of old trapezoids that are destroyed and the number of new trapezoids that are created during the insertion of s . This quantity is called the *structural change*.

This results in a rather simple algorithm to compute the trapezoidal map of a set of line segments. Starting with the empty set, we treat the line segments one-by-one, maintaining the trapezoidal map of the set of line segments inserted so far.

However, a general disadvantage of incremental algorithms is that the total structural change during the insertions of n objects, and hence the running time of the algorithm, depends strongly on the order in which the objects are processed. In our case, it is not difficult to devise a sequence of n line segments leading to a total structural change of $\Theta(n^2)$. Even if we know that a good order of insertion exists (one that implies a small structural change), it seems difficult to determine this order beforehand. And this is exactly where randomization can help: we simply treat the n objects in random order. In the case of the trapezoidal map, we will show below that if the n segments are processed in random order, the *expected* structural change in every step of the algorithm is only constant.

BACKWARDS ANALYSIS

An easy way to see this is via *backwards analysis*. We first observe that it suffices to bound the number of trapezoids created in each stage of the algorithm. All these trapezoids are incident to the segment inserted in that stage. We imagine the algorithm removing the line segments from the final map one-by-one. In each step, we must bound the number of trapezoids incident to the segment s removed. Now we make two observations:

- The trapezoidal map is a planar graph, with every trapezoid incident to at most 4 segments. Hence, if there are m segments in the current set, the total number of trapezoid-segment incidences is $O(m)$.
- Since the order of the segments is a random permutation of the set of segments, each of the m segments is equally likely to be removed.

These two facts suffice to show that the expected number of trapezoids incident to s is constant. In fact, this number is bounded by the average degree of a segment in a trapezoidal map.

It follows that the expected total structural change during the course of the algorithm is $O(n)$. To obtain an efficient algorithm, however, we need a second ingredient: whenever a new segment s is inserted, we need to identify the trapezoids of the old map intersected by s . Two basic approaches are known to solve this problem: the conflict graph and the history graph.

CONFLICT GRAPH

A conflict graph is a bipartite graph whose nodes are the not-yet-added segments on one side and the trapezoids of the current map on the other side. There is an arc between a segment s and a trapezoid Δ if and only if s intersects Δ , in which case we say that s is in conflict with Δ .

It is possible to maintain the conflict graph during the course of the incremental algorithm. Whenever a new segment is inserted, all the conflicts of the newly-created trapezoids are found. This is not difficult, because a segment can only conflict with a newly-created trapezoid if it was previously in conflict with the old trapezoids at the same place. Thus the trapezoids intersected by the new segment s are just the neighbors of s in the conflict graph.

The time necessary to maintain the conflict graph can be bounded by summing the number of conflicts of all trapezoids created during the course of the algorithm. It follows from the higher-moments bound (Eq. 40.1.2) that the average number of conflicts of the trapezoids present after inserting the first r segments—note that these segments form a random sample of size r of S —is $O(n/r)$. Intuitively, we can assume that this is also correct if we look only at the trapezoids that are *created* by the insertion of the r th segment. Since the expected number of trapezoids created in every step of the algorithm is constant, the expected total time is $\sum_{i=1}^n O(n/r) = O(n \log n)$.

Note that an algorithm using a conflict graph needs to know the entire set of objects (segments in our example) in advance.

HISTORY GRAPH

A different approach uses a history graph, which records the history of changes in the maintained structure.

In our example, we can maintain a directed acyclic graph whose nodes correspond to trapezoids constructed during the course of the algorithm. The leaves are the trapezoids of the current map; all inner nodes correspond to trapezoids that have already been destroyed (with the root corresponding to the entire plane). When we insert a segment s , we create new nodes for the newly-created trapezoids, and create a pointer from an old trapezoid to every new one that overlaps it. Hence, there are at most four outgoing pointers for every inner node of the history graph.

We can now find the trapezoids intersected by a new segment s by performing a graph search from the root, using say, depth-first search on the connected subgraph consisting of all trapezoids intersecting s . Note that this search performs precisely the same computations that would have been necessary to maintain the conflict graph during the sequence of updates, but at a different time. We can therefore consider a history graph as a lazy implementation of a conflict graph: it postpones each computation to the moment it is actually needed. Consequently, the analysis is exactly the same as for conflict graphs.

Algorithms using a history graph are *on-line* or *semidynamic* in the sense that they do not need to know about a point until the moment it is inserted.

ABSTRACT FRAMEWORK AND ANALYSIS

Most randomized incremental algorithms in the literature follow the framework sketched here for the computation of the trapezoidal map: the structure to be

computed is maintained while the objects defining it are inserted in random order. To insert a new object, one first has to find a “conflict” of that object (the *location step*), then local updates in the structure are sufficient to bring it up to date (the *update step*). The cost of the update is usually linear in the size of the change in the combinatorial structure being maintained, and can often be bounded using backwards analysis. The location step can be implemented using either a conflict graph or a history graph. In both cases, the analysis is the same (since the actual computations performed are also often identical). To avoid having to prove the same bounds repeatedly for different problems, researchers have defined an axiomatic framework that captures the combinatorial essence of most randomized incremental algorithms. This framework, which uses *configuration spaces*, provides ready-to-use bounds for the expected running time of most randomized incremental algorithms. See [Section 40.5](#).

POINT LOCATION THROUGH HISTORY GRAPH

In our trapezoidal map example, the history graph may be used as a point location structure for the trapezoidal map: given a query point q , find the trapezoid containing q by following a path from the root to a leaf node of the history graph. At each step, we continue to the child node corresponding to the trapezoid containing q .

The search time is clearly proportional to the length of the path. Backwards analysis shows that the expected length of this path is $O(\log n)$ for any fixed query point. Even stronger, one can show that the maximum length of any search path in the history graph is $O(\log n)$ with high probability.

If point location is the goal, the history graph can be simplified: instead of storing trapezoids, the inner nodes of the graph can denote two different kinds of elementary tests (“Does a point lie to the left or right of another point?” and “Does a point lie above or below a line?”). The final result is then an efficient and practical planar point location structure [Sei91].

This observation can also lead to a somewhat different location step inside the randomized incremental algorithm. Instead of performing a graph search with the whole segment s , point location can be used to find the trapezoid containing one endpoint of s . From there, a traversal of the trapezoidal map allows locating all trapezoids intersected by s .

APPLICATIONS

The randomized incremental framework has been successfully applied to a large variety of problems. We list a number of important such applications. Details on the results can be found in the chapters dealing with the respective area, or in one of the surveys cited in [Section 40.12](#).

- Trapezoidal decomposition formed by segments in the plane, and point location structures for this decomposition ([Section 34.3](#)).
- Triangulation of simple polygons: an optimal randomized algorithm with linear running time, and a simple algorithm with running time $O(n \log^* n)$ ([Section 26.2](#)).
- Convex hulls of points in d -dimensional space, output-sensitive convex hulls

in \mathbb{R}^3 (Section 22.3).

- Voronoi diagrams in different metrics, including higher order and abstract Voronoi diagrams (Section 23.3).
 - Linear programming in finite-dimensional space ([Chapter 45](#)).
 - Generalized linear programming: optimization problems that are combinatorially similar to linear programming (Section 45.4).
 - Hidden surface removal (Section 28.8 and [Chapter 49](#)).
 - Constructing a single face in an arrangement of (curved) segments in the plane, or in an arrangement of triangles or surface patches in \mathbb{R}^3 (Sections 24.5 and 47.2); computing zones in an arrangement of hyperplanes in \mathbb{R}^d (Section 24.4).
-

40.3 DYNAMIC ALGORITHMS

DYNAMIC RANDOMIZED INCREMENTAL

Any on-line randomized incremental algorithm can be used as a semidynamic algorithm, a dynamic algorithm that can only perform insertions of objects. The bound on the expected running time of the randomized incremental algorithm then turns into a bound on the *average* running time, under the assumption that every permutation of the input is equally likely. (The relation between the two uses of the algorithms is similar to that between randomized and ordinary Quicksort as mentioned in the Introduction.)

This observation has motivated researchers to extend randomized incremental algorithms so that they can also manage deletions of objects. Then bounds on the average running time of the algorithm are given, under the assumption that the input sequence is a *random update sequence*. In essence, one assumes that for an addition, every object currently not in the structure is equally likely to be inserted, while for a deletion every object currently present is equally likely to be removed (the precise definition varies between authors).

Two approaches have been suggested to handle deletions in history-graph based incremental algorithms. The first adds new nodes at the leaf level of the history graph for every deletion. This works for a wide variety of problems and is relatively easy to implement, but after a number of updates the history graph will become “dirty”: it will contain elements that are no longer part of the current structure but which still must be traversed by the point-location steps. Therefore, the history graph needs periodic “cleaning.” This can be accomplished by discarding the current graph, and reconstructing it from scratch using the elements currently present.

In the second approach, for every deletion the history graph is transformed to the state it would have been had the object never been inserted. The history graph is therefore always “clean.” However, in this model deletions are more complicated, and it therefore seems to apply to fewer problems.

DYNAMIC SAMPLING AND GRADATIONS

A rather different approach permits a number of search structures based on bottom-up sampling to be dynamized surprisingly easily. Such a search structure consists of a *gradation* using Bernoulli sampling (Section 40.1): The gradation is a hierarchy of $O(\log n)$ levels. Every object is included in the first level, and is chosen independently to be in the second level with probability $\frac{1}{2}$. Every object in the second level is propagated to level 3 with probability $\frac{1}{2}$, and so forth. Whenever an object is added to or removed from the current set, the search structure is updated to the proper state. When adding an object, it suffices to flip a coin at most $\log n$ times to determine where to place the object. Using this technique, it is possible to give high-probability bounds on the search time and sometimes also on the update time [Mul93].

40.4 RANGE SPACES

“Pointwise bounds” of the form in Equation 40.1.1 can be proved in the axiomatic framework of range spaces, which then leads to immediate application to a wide variety of geometric settings.

GLOSSARY

Range space: A pair (X, Γ) , with X a universe (possibly infinite), and Γ a family of subsets of X . The elements of Γ are called **ranges**. Typical examples of range spaces are of the form (\mathbb{R}^d, Γ) , where Γ is a set of geometric figures, such as all line segments, halfspaces, simplices, balls, etc. (cf. Section 36.2).

Shattered: A set $A \subseteq X$ is shattered if every subset A' of A can be expressed as $A' = A \cap \gamma$, for some range $\gamma \in \Gamma$.

In the range space $(\mathbb{R}^2, \mathcal{H})$, where \mathcal{H} is the set of all closed halfplanes, a set of three points in convex position is shattered. However, no set of four points is shattered. See Figure 40.4.1: whether the point set is in convex position or not, there always is a subset (encircled) that cannot be expressed as $A \cap h$ for any halfplane h .



FIGURE 40.4.1

No set of four points can be shattered by halfplanes.

In the range space $(\mathbb{R}^2, \mathcal{C})$, where \mathcal{C} is the set of all convex polygons, any set of points lying on a circle is shattered.

Vapnik-Chervonenkis dimension (VC-dimension): The VC-dimension of a range space (X, Γ) is the smallest integer d such that there is no shattered subset $A \subseteq X$ of size $d + 1$. If no such d exists, the VC-dimension is said to be infinite.

Range spaces (\mathbb{R}^d, Γ) , where Γ is the set of line segments, of simplices, of balls, or of halfspaces, have finite VC-dimension. For example, the range space $(\mathbb{R}^2, \mathcal{H})$ has VC-dimension 3. The range space $(\mathbb{R}^2, \mathcal{C})$, however, has infinite VC-dimension.

Shatter function: For a range space (X, Γ) , the shatter function $\pi_\Gamma(m)$ is defined as

$$\pi_\Gamma(m) = \max_{A \subset X, |A|=m} |\{A \cap \gamma \mid \gamma \in \Gamma\}|.$$

If the VC-dimension of the range space is infinite, then $\pi_\Gamma(m) = 2^m$. Otherwise the shatter function is bounded by $O(m^d)$, where d is the VC-dimension. (So the shatter function of any range space is either exponential or polynomially bounded.) If the shatter function is polynomial, the VC-dimension is finite. The order of magnitude of the shatter function is not necessarily the same as the VC-dimension; for instance, the range space $(\mathbb{R}^2, \mathcal{H})$ has VC-dimension 3 and shatter function $O(m^2)$. Since the VC-dimension is often difficult to compute, some authors have defined the *VC-exponent* as the order of magnitude of the shatter-function.

ϵ -net: A subset $N \subseteq X$ is called an ϵ -net for the range space (X, Γ) if $N \cap \gamma \neq \emptyset$ for every $\gamma \in \Gamma$ with $|\gamma|/|X| > \epsilon$ (here, $\epsilon \in [0, 1]$ and X is finite). It is often more convenient to write $1/r$ for ϵ , with $r > 1$.

ϵ -approximation: A subset $A \subseteq X$ is called an ϵ -approximation for the range space (X, Γ) if, for every $\gamma \in \Gamma$, we have

$$\left| \frac{|A \cap \gamma|}{|A|} - \frac{|\gamma|}{|X|} \right| \leq \epsilon.$$

An ϵ -approximation is also an ϵ -net, but not necessarily vice versa.

Linear range space: The range space $(\mathbb{R}^d, \mathcal{L}_k^d)$, where \mathcal{L}_k^d consists of unions of polytopes of total complexity at most k in \mathbb{R}^d .

Linearizable range space: A typical range space $(\mathcal{X}, \mathcal{X}_C)$ is defined by a set of geometric “objects” \mathcal{X} and a set of geometric “cells” \mathcal{C} : A cell $\Delta \in \mathcal{C}$ defines a range \mathcal{X}_Δ that consists of all the objects $x \in \mathcal{X}$ that intersect Δ . $(\mathcal{X}, \mathcal{X}_C)$ is linearizable if it can be embedded into a linear range space, that is, if there are constants d, k and maps $\varphi : \mathcal{X} \rightarrow \mathbb{R}^d$ and $\psi : \mathcal{C} \rightarrow \mathcal{L}_k^d$, such that for $x \in \mathcal{X}$ and $\Delta \in \mathcal{C}$, $x \cap \Delta \neq \emptyset$ iff $\varphi(x) \in \psi(\Delta)$ [YY85, AM94].

ϵ -NETS AND ϵ -APPROXIMATIONS

The pointwise bound translates into the abstract framework of range spaces as follows:

THEOREM 40.4.1

Let (X, Γ) be a range space with X finite and of finite VC-dimension d . Then a random sample $R \subset X$ of size $C(d)r \log r$ is a $1/r$ -net for (X, Γ) with probability whose complement to 1 is polynomially small in r . The constant $C(d)$ depends only on d .

This theorem forms the basis for “traditional” randomized divide-and-conquer algorithms, such as the one for the trapezoidal map of line segments sketched in

Section 40.1. The pointwise bound used there follows from the theorem. Consider the range space (S, Γ) , where $\Gamma := \{\gamma(\Delta) \mid \Delta \text{ an open trapezoid}\}$, and $\gamma(\Delta)$ is the set of all segments in S intersecting Δ . The VC-dimension of this range space is finite. The easiest way to see this is by looking at the shatter function. Consider a set of m line segments. Extend them to full lines, pass $2m$ vertical lines through all endpoints, and look at the arrangement of these $3m$ lines. Clearly, for any two trapezoids Δ and Δ' whose corners lie in the same faces of this arrangement we have $\gamma(\Delta) = \gamma(\Delta')$. Consequently, there are at most $O(m^8)$ different ranges, and that crudely bounds the shatter function as $O(m^8)$. Thus the VC-dimension is finite and Theorem 40.4.1 applies: with probability increasing rapidly with r , the sample R of size r is an ϵ -net for S with $\epsilon = \Omega((1/r) \log r)$. Assume this is the case, and consider some trapezoid $\Delta \in \mathcal{T}(R)$. The interior of Δ does not intersect any segment in R , so by the property of ϵ -nets, the range $\gamma(\Delta)$ can intersect at most ϵn segments of S . And so we have $n_\Delta = O((n/r) \log r)$.

The construction of ϵ -nets has been so successfully derandomized that ϵ -nets now are used routinely in deterministic algorithms (Section 40.7). At least in theory, the top-down sampling algorithm of Section 40.1 need no longer be considered a randomized algorithm.

ϵ -approximations are used in the deterministic computation of ϵ -nets (Section 40.7). They are also interesting in their own right, since some geometric problems—for instance, the computation of centerpoints or ham-sandwich cuts (see Section 14.2)—can be solved approximately by solving them exactly for an ϵ -approximation. A $1/r$ -approximation can be found by taking a random sample of size $O(r^2 \log r)$. This bound is not tight.

VC-dimension and ϵ -nets are also frequently used in statistics [VC71] (from which they derive) and in learning theory.

LINEARIZING RANGE SPACES

Most range spaces that appear in geometric problems can be linearized. A general procedure to obtain φ and ψ is the following [MS96]: Start with a first-order predicate Π in the theory of closed fields—one formed from polynomial inequalities using Boolean connectives and quantifiers, where the parameters defining x are regarded as variables, and the parameters defining Δ are regarded as constants—that describes when $x \cap \Delta \neq \emptyset$; then, using a quantifier elimination method, rewrite Π as a disjunction of several conjunctions of polynomial inequalities; finally, by introducing a variable for each monomial that appears in the polynomial inequalities, obtain linear inequalities that correspond to a linear cell.

For example, consider a set S of line segments and let $R \subseteq S$; we are interested in the conflict sets S_Δ for $\Delta \in \mathcal{T}(R)$. Whether a segment $s = uv \in S$ intersects $\Delta \in \mathcal{T}(R)$ can be written as a predicate in which the parameters defining s are regarded as variables, and the parameters defining Δ are regarded as constants: either $u \in \Delta$, or $v \in \Delta$, or uv intersects one of the edges of Δ .

In specific cases, a more efficient embedding (with smaller values of d and k) is possible. An important example is the range space defined on points in \mathbb{R}^d by balls. It can be linearized by lifting the points to the paraboloid $x_{d+1} = x_1^2 + \cdots + x_d^2$ in \mathbb{R}^{d+1} .

One application of linearization is the computation of conflict lists. Consider the computation of the conflict lists S_Δ for $\Delta \in \mathcal{T}(R)$, $R \subseteq S$. By linearization, S_Δ

is equal to the set $\varphi(S) \cap \psi(\Delta)$. We have now the problem of locating the points $\varphi(S)$ in the arrangement of the hyperplanes H in \mathbb{R}^d that bound the linear cells $\psi(\Delta)$. We construct a point location data structure for the arrangement of H , and use it to locate each of the points $\varphi(s)$, $s \in S$. If $r = |\mathcal{T}(R)|$ and $n = |S|$, then the cost of this procedure is $O(r^{d+1} + n \log r + k)$, where k is the number of conflicts reported. This approach is very general and easily parallelizable; on the other hand, it can be relatively inefficient in comparison with other problem-specific methods that make more use of the particular geometry of the problem.

Linearization is also used in the deterministic construction of cuttings (Section 40.7).

OPEN PROBLEM

For general range spaces, the bound $O(r \log r)$ in Theorem 40.4.1 is the best possible. However, for many geometrically-defined spaces the best lower bound is $\Omega(r)$. Can the upper bound be improved for some geometric range spaces? This is perhaps a difficult problem [MSW90].

40.5 CONFIGURATION SPACES

The framework of configuration spaces is somewhat more complicated than range spaces, but facilitates proving higher-moment bounds as in Equation 40.1.2. Terminology, axiomatics, and notation vary widely between authors. Note that the term “configuration space” is used in robotics with a different meaning (see [Chapters 47](#) and [48](#)).

GLOSSARY

Configuration space: A four-tuple (X, \mathcal{T}, D, K) . X is a finite set of geometric objects (the universe of size n). \mathcal{T} is a mapping that assigns to every subset $S \subseteq X$ a set $\mathcal{T}(S)$; the elements of $\mathcal{T}(S)$ are called **configurations**. $\Pi(X) := \bigcup_{S \subseteq X} \mathcal{T}(S)$ is the set of all configurations occurring over some subset of X . D and K assign to every configuration $\Delta \in \Pi(X)$ subsets $D(\Delta)$ and $K(\Delta)$ of X . Elements of the set $D(\Delta)$ are said to **define** the configuration (they are also called **triggers**) and the elements of the set $K(\Delta)$ are said to **kill** the configuration (they are also said to be in **conflict** with the configuration and are sometimes called **stoppers**).

Conflict size of Δ : The number of elements of $K(\Delta)$.

We will require the following axioms:

- (i) The number $d = \max\{|D(\Delta)| \mid \Delta \in \Pi(X)\}$ is a constant (called the **maximum degree** or the **dimension** of the configuration space). Moreover, the number of configurations sharing the same defining set is bounded by a constant.
- (ii) For any $\Delta \in \mathcal{T}(S)$, $D(\Delta) \subseteq S$ and $S \cap K(\Delta) = \emptyset$.
- (iii) If $\Delta \in \mathcal{T}(S)$ and $D(\Delta) \subseteq S' \subseteq S$, then $\Delta \in \mathcal{T}(S')$.

(iii') If $D(\Delta) \subseteq S$ and $K(\Delta) \cap S = \emptyset$, then $\Delta \in \mathcal{T}(S)$.

Note that axiom (iii) follows from (iii'); see below.

EXAMPLES

1. *Trapezoidal map.* The universe X is a set of segments in the plane, and $\mathcal{T}(S)$ is the set of trapezoids in the trapezoidal map of S . The defining set $D(\Delta)$ is the set of segments that are necessary to define Δ (at most four segments suffice, so $d = 4$), and the killing set $K(\Delta)$ is the set of segments that intersect the trapezoid. It is easy to verify that conditions (i), (ii), (iii), (iii') all hold.
2. *Delaunay triangulation.* X is a set of points in the plane (assume that no four points lie on a circle), and $\mathcal{T}(S)$ is the set of triangles of the Delaunay triangulation of S . $D(\Delta)$ consists of the vertices of triangle Δ (so $d = 3$), while $K(\Delta)$ is the set of points lying inside the circumcircle of the triangle. Again, axioms (i), (ii), (iii), (iii') all hold.
3. *Convex hulls in 3D.* The universe X is a set of points in 3D (assume that no four points are coplanar), and $\mathcal{T}(S)$ is the set of facets of the convex hull of S . The defining set of a facet Δ is the set of its vertices ($d = 3$), and the killing set is the set of points lying in the outer open halfspace defined by Δ . Note that there can be two configurations sharing the same defining set. Again, axioms (i)–(iii') all hold.
4. *Single cell.* The universe X is a set of possibly intersecting segments in the plane, and $\mathcal{T}(S)$ is the set of trapezoids in the trapezoidal map of S that belongs to the cell of the line segment arrangement containing the origin (Figure 40.5.1). The defining and killing sets are defined as in the case of the trapezoidal map of the whole arrangement above. In this situation, axiom (iii') does not hold. Whether or not a given trapezoid appears in $\mathcal{T}(S)$ depends on segments other than the ones in $D(\Delta) \cup K(\Delta)$. Axioms (i), (ii), (iii) are nevertheless valid.

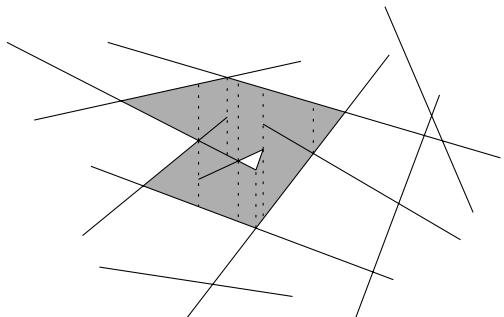


FIGURE 40.5.1

A single cell in an arrangement of line segments.

5. *Counterexample.* Let X be a set of line segments, and let $\mathcal{T}(S)$ be a decomposition of the arrangement that is obtained by drawing vertical extensions for faces with an even number of edges, and horizontal extensions for faces

with an odd number of edges. Axioms (i) and (ii) hold, but neither (iii) nor (iii') is satisfied.

Note that when (ii) and (iii') both hold, then $\Delta \in \mathcal{T}(S)$ if and only if $D(\Delta) \subseteq S$ and $K(\Delta) \cap S = \emptyset$. In other words, the mapping \mathcal{T} is then completely defined by the functions D and K . In fact, in the first three examples we can decide from local information alone whether or not a configuration appears in $\mathcal{T}(S)$. For instance, a triangle Δ is in the Delaunay triangulation of S if and only if the vertices of Δ are in S , and no point of S lies in the circumcircle of Δ .

As mentioned above, axiom (iii) follows from (iii'), but not conversely. Axiom (iii) requires a kind of monotonicity: if Δ occurs in $\mathcal{T}(S)$ for some S , then we cannot destroy it by removing elements from S unless we remove some element in $D(\Delta)$.

We may say that the configuration spaces of the first three examples are defined *locally* and *canonically*. The fourth example is *canonical*, but *nonlocal*. The last example is not canonical and cannot be treated with the methods described here. (Fortunately, this is an artificial example with no practical use—but see the open problems below.)

HIGHER-MOMENTS AND EXPONENTIAL DECAY LEMMA

The higher-moments bound for configuration spaces generalizes the bound for trapezoidal maps, Equation 40.1.2:

THEOREM 40.5.1 *Higher-moments bound*

Let (X, \mathcal{T}, D, K) be a configuration space satisfying axioms (i), (ii), (iii), and let R be a random sample of X of size r . For any constant c , we have

$$E \left[\sum_{\Delta \in \mathcal{T}(R)} |K(\Delta)|^c \right] = O((n/r)^c E[|\mathcal{T}(R)|]).$$

(Technically, rather than R , a sample R' of size $\lfloor r/2 \rfloor$ should appear on the right, but $E[|\mathcal{T}(R')|] = O(E[|\mathcal{T}(R)|])$ in all cases of interest). In other words, as far as the c th-degree average is concerned, the conflict size behaves as if it were $O(n/r)$, instead of $O((n/r) \log r)$ from the pointwise bound.

Let (X, \mathcal{T}, D, K) and R be as in Theorem 40.5.1. For any natural number t , we define $\mathcal{T}_t(R)$ to be the subset of configurations of $\mathcal{T}(R)$ whose conflict size exceeds the “natural” value n/r by at least the factor t :

$$\mathcal{T}_t(R) := \{\Delta \in \mathcal{T}(S) \mid |K(\Delta)| \geq tn/r\}.$$

The following exponential-decay lemma [AMS98] states that the number of such configurations decreases exponentially with t :

THEOREM 40.5.2 *Exponential decay lemma*

Let (X, \mathcal{T}, D, K) be a configuration space satisfying axioms (i), (ii), (iii), and let R be a random sample of X of size r . For any t with $1 \leq t \leq r/d$ (where d is as in axiom (i)), we have

$$E[|\mathcal{T}_t(R)|] = O(2^{-t}) \cdot E[|\mathcal{T}(R')|],$$

where $R' \subseteq X$ denotes a random sample of size $\lfloor r/t \rfloor$.

The exponential decay lemma implies both the higher-moments bound, by adding over t , and the pointwise bound, by Markov's inequality.

RANDOMIZED INCREMENTAL CONSTRUCTION

Many, if not most, randomized incremental algorithms in the literature can be analyzed using the configuration space framework. Given the set X , the goal of the randomized incremental algorithm is to compute $\mathcal{T}(X)$. This is done by maintaining $\mathcal{T}(X^i)$, for $1 \leq i \leq n$, where $X^i = \{x_1, x_2, \dots, x_i\}$ and the x_i form a random permutation of X .

To bound the number of configurations created during the insertion of x_i into X^{i-1} , we observe that by axiom (iii) these configurations are exactly those $\Delta \in \mathcal{T}(X^i)$ with $x_i \in D(\Delta)$. The expected number of these can be bounded by

$$\frac{d}{i} E[|\mathcal{T}(X^i)|]$$

using backwards analysis. Here, d is the maximum degree of the configuration space.

The expected total change in the conflict graph or history graph can be bounded by summing $|K(\Delta)|$ over all Δ created during the course of the algorithm. Using axioms (i) to (iii'), we can derive the following bound:

$$\sum_{i=1}^n d^2 \frac{n-i}{i} \frac{E[|\mathcal{T}(X^i)|]}{i}.$$

(The exact form of this expression depends on the model used.) The book [Mul93] treats randomized incremental algorithms systematically using the configuration space framework (assuming axiom (iii')).

LAZY RANDOMIZED INCREMENTAL CONSTRUCTION

In problems that have nonlocal definition, such as the computation of a single cell in an arrangement of segments, single cells in arrangements of surface patches, or zones in arrangements, the update step of a randomized incremental construction becomes more difficult. Besides the local updates in the neighborhood of the newly inserted object, there may also be global changes. For instance, when a line segment is inserted into an arrangement of line segments, it may cut the single cell being computed into several pieces, only one of which is still interesting. The technique of lazy randomized incremental construction [BDS95] deals with these problems by simply postponing the global changes to a few “clean-up” stages. Since the setting of all these problems is nonlocal, the analysis uses only axioms (i), (ii), (iii).

OPEN PROBLEM

The canonical framework of randomized incremental algorithms sketched above is sometimes too restrictive. For instance, to make a problem fit into the framework, one often has to assume that objects are in general position. While many algorithms

could deal with special cases (e.g., four points on a circle in the case of Delaunay triangulations) directly, the analysis does not hold for those situations, and one has to resort to a symbolic perturbation scheme to save the analysis. Can a more relaxed framework for randomized incremental construction be given [Sei93]?

40.6 DERANDOMIZATION TECHNIQUES

Even when an efficient randomized algorithm for a problem is known, researchers still find it worthwhile to obtain a deterministic algorithm of the same efficiency. The reasons for doing this are varied, from scientific curiosity (what is the real power of randomness?), to practical reasons (truly random bits are quite expensive), to a preference for “deterministic” that may not be strictly rational. Sometimes a deterministic algorithm for a given problem may be obtained by “simulating” or “derandomizing” a randomized algorithm. Derandomization has turned out to be a powerful theoretical tool: for several problems the only known worst-case optimal deterministic algorithm has been obtained by derandomization. The most famous example is computing the convex hull of n points in d -dimensional space (Section 22.3).

General derandomization techniques can be used to produce a deterministic counterpart of random sampling in both configuration spaces and range spaces. As a result, it is possible to obtain in polynomial time a sample that satisfies the higher-moment bound, or that is a net or an approximation. Taking advantage of separability and composition properties of approximations, these constructions can be made efficient. In most applications, deterministic sampling is the base of a deterministic divide-and-conquer algorithm or data structure, which is almost as efficient as the randomized counterpart.

On the other hand, incremental algorithms are considerably harder to derandomize: the convex hull algorithm mentioned above is essentially the only successful case. The problem is that in an incremental algorithm each insertion must be “globally good,” while in the divide-and-conquer case, items are chosen locally in a neighborhood that shrinks as the algorithm progresses. In some cases, such as linear programming, a derandomized divide-and-conquer approach leads to a deterministic algorithm with better dependency on the dimension than previously known methods (prune-and-search), but there still remains a large gap with respect to the best randomized algorithm (which is an incremental one).

METHOD OF CONDITIONAL PROBABILITIES

The *method of conditional probabilities* (also called the *Raghavan-Spencer method*) [Spe87, Rag88] implements a binary search of the probability space to determine an event with the desired properties (guaranteed by a probabilistic analysis). Given a configuration space (X, \mathcal{T}, D, K) , the goal is to obtain a random sample of size (approximately) r that satisfies the higher-moments bound. Let $X = \{x_1, \dots, x_n\}$ and Ω be the probability space on $\{0, 1\}^n$, and consider the probability distribution on Ω induced by selecting each component equal to 1 independently with probability $p = r/n$ (for convenience, we use Bernoulli sampling). Let $F : \Omega \rightarrow \mathbb{R}$ be the random variable that assigns to the vector (q_1, \dots, q_n) the

value $\sum_{\Delta \in \mathcal{T}(R)} f(|K(\Delta) \cap X|)$, where $x_i \in R$ iff $q_i = 1$, and $f(x) = x^k$ (for the k th moment; using $f(x) = e^{c(r/n)x}$ with an appropriate constant c , one can achieve the exponential decay bound). We know that $E[F] \leq M$ with $M = Cf(n/r)t(r)$, where $t(r)$ is an upper bound for $E[|\mathcal{T}(R)|]$. The method is based on the following relation, for $0 \leq i < n$:

$$\begin{aligned} E[F|q_1 = v_1, \dots, q_i = v_i] \\ = p \cdot E[F|q_1 = v_1, \dots, q_i = v_i, q_{i+1} = 1] + \\ (1-p) \cdot E[F|q_1 = v_1, \dots, q_i = v_i, q_{i+1} = 0] \\ \geq \min\{E[F|q_1 = v_1, \dots, q_i = v_i, q_{i+1} = 1], E[F|q_1 = v_1, \dots, q_i = v_i, q_{i+1} = 0]\} \end{aligned}$$

If these conditional expectations can be computed *efficiently*, then this implies an efficient procedure to select v_{i+1} so that

$$E[F|q_1 = v_1, \dots, q_i = v_i] \geq E[F|q_1 = v_1, \dots, q_i = v_i, q_{i+1} = v_{i+1}].$$

Iterating this procedure, one finally obtains a solution (v_1, \dots, v_n) that satisfies the probabilistic bound

$$M \geq E[F] \geq E[F|q_1 = v_1, \dots, q_n = v_n].$$

If the locality property holds, the conditional probabilities involved can indeed be computed in polynomial time: Let $X_i = \{x_1, x_2, \dots, x_i\}$ and $R_i = \{x_j \in X : q_j = 1, j \leq i\}$, then $E[F|q_1 = v_1, \dots, q_i = v_i]$ is equal to

$$\begin{aligned} & \sum_{\Delta \in \Pi(X)} \Pr\{\Delta \in \mathcal{T}(R) | q_1 = v_1, \dots, q_i = v_i\} f(|K(\Delta) \cap X|) \\ = & \sum_{\Delta \in \Pi(X) : D(\Delta) \cap X_i \subseteq R_i, K(\Delta) \cap R_i = \emptyset} p^{|D(\Delta) \setminus S_i|} (1-p)^{|K(\Delta) \cap (X \setminus X_i)|} f(|K(\Delta) \cap X|), \end{aligned}$$

which can be approximated with sufficient accuracy. Similarly, $(1/r)$ -nets and $(1/r)$ -approximations of sizes $O(r \log r)$ and $O(r^2 \log r)$ can be computed in polynomial time.

k -WISE INDEPENDENT DISTRIBUTIONS

The method of conditional probabilities is highly sequential. An approach that is more suitable for parallel algorithms is to construct a probability space of polynomial size, and to execute the algorithm on each vector of this space. This is possible, for example, when the variables q_i need only be k -wise independent rather than being fully independent: for any indices i_1, \dots, i_k , and 0-1 values, v_1, \dots, v_k ,

$$\Pr\{q_{i_1} = v_1, \dots, q_{i_k} = v_k\} = \prod_{j=1}^k \Pr\{q_{i_j} = v_{i_j}\} = \prod_{j=1}^k p^{v_j} (1-p)^{1-v_j}.$$

A probability space and distribution of size $O(n^k)$ with such k -wise independence can be computed effectively [Jof74, KM94]. Let $\rho \geq n$ be a prime number and suppose that $p_1, \dots, p_n \in [0, 1]$ satisfy $p_i = j_i/\rho$, for some integers j_i . Define a probability space with at most n^k points, as follows. For each $\langle a_0, a_1, \dots, a_{k-1} \rangle$ in $\{0, 1, \dots, \rho - 1\}^k$, let

$$X_i = a_0 + a_1 i + a_2 i^2 + \dots + a_{k-1} i^{k-1} \bmod \rho,$$

for $1 \leq i \leq n$, assign probability $1/\rho^k$ and associate the vector $\langle Y_1, \dots, Y_n \rangle$ where $Y_i = 1$ if $X_i \in \{0, 1, \dots, j_i - 1\}$ and $Y_i = 0$ otherwise. The 0-1 probability space defined by the vectors $\langle Y_1, \dots, Y_n \rangle$ is a k -wise independent 0-1 probability space for p_1, \dots, p_n . With this construction, arbitrary probabilities can be approximated (within a factor of 2) by an appropriate choice of ρ . Using a larger space of size $O(n^{2k})$, arbitrary probabilities can be achieved exactly [KM94].

For some randomized algorithms one can show that they still work under k -wise independency for an appropriate k . For example, a quasi-random permutation with k -wise independence suffices for the randomized incremental approach to work [Mul96] (thus $O(\log n)$ random bits suffice rather than the $\Omega(n \log n)$ bits needed to define a fully random permutation). To verify that k -wise independence suffices, a tail inequality under k -independence is used [SSS93, BR94]. Let q_1, \dots, q_n be a sequence of k -wise independent random variables in $\{0, 1\}$, with $k \geq 2$ even, let $Q = \sum_{i=1}^n q_i$, $\mu = \mathbb{E}[Q]$ and assume that $\mu \geq k$, then

$$\Pr\{Q = 0\} < \frac{C_k}{\mu^{k/2}} \quad (40.6.1)$$

where C_k is a constant depending on k . Let R be a $2k$ -wise independent random sample from X with uniform probability p . For $\Delta \in \Pi(X)$, note that (no independence assumption needed)

$$\Pr\{D(\Delta) \subseteq R, K(\Delta) \cap R = \emptyset\} = \Pr\{D(\Delta) \subseteq R\} \cdot \Pr\{K(\Delta) \cap R = \emptyset | D(\Delta) \subseteq R\}.$$

Let d be an upper bound on $|D(\Delta)|$. The first factor can be computed using $2k$ -wise independence assuming $2k \geq d$:

$$\Pr\{D(\Delta) \subseteq R\} = p^{|D(\Delta)|}.$$

To upper bound the second factor, let $t_\Delta = p|K(\Delta)|$; then using the tail bound above, for $t_\Delta \geq k$:

$$\Pr\{K(\Delta) \cap R = \emptyset | D(\Delta) \subseteq R\} \leq \frac{C}{t_\Delta^{k-d/2}},$$

since after fixing $D(\Delta) \subseteq R$, the remaining random variables are still $(2k - d)$ -wise independent. Choosing k so that $c \leq k - d/2 + 2$, one can verify that the c th moment bound holds. Similarly, $1/r$ -nets and $1/r$ -approximations with sizes $O(rn^\delta)$ and $O(r^2n^\delta)$ can be computed in polynomial time (or fast in parallel with polynomial work), where $\delta = O(1/k)$. It does not seem possible, however, to achieve the exponential decay bound with a limited-independence space of polynomial size.

For fixed k , the size of the space can be reduced if a certain deviation from k -wise independence is allowed [NN90]. Furthermore, the approach of testing all the vectors in the probability space can be combined with the approach of performing a binary search so that even a space of superpolynomial size is usable [MNN89, BRS89]. Still, these approaches do not lead to the exponential decay bound, or to nets or approximations of size matching the probabilistic analysis.

CONSTRAINT-BASED PROBABILITY SPACES

An alternative approach that is implementable in parallel constructs a probability distribution tailored to a particular algorithm and even to a specific input [Nis92,

KM93, KK94, MRS01], leading to smaller probability spaces. The approach models the sampling process using *randomized finite automata* (RFA), and fools the automaton using a probability distribution D_n with support of size E_0 that depends polynomially on the error and on the size of the problem. Once the probability distribution has been constructed, it is only a matter of testing the algorithm for each point in D_n .

For each configuration Δ we construct an RFA M_Δ as follows: It consists of $n + 1$ levels $N_{\Delta,j}$, $0 \leq j \leq n$, each with two states $\langle j, \text{Yes} \rangle$ and $\langle j, \text{No} \rangle$, with transitions that reflect whether $\Delta \in \mathcal{T}(R)$: $\langle j - 1, \text{No} \rangle$ is always connected to $\langle j, \text{No} \rangle$; if $x_j \in D(\Delta)$ then $\langle j - 1, \text{Yes} \rangle$ is connected to $\langle j, \text{Yes} \rangle$ under $q_j = 1$ and to $\langle j, \text{No} \rangle$ under $q_j = 0$; if $x_j \in K(\Delta)$ then $\langle j - 1, \text{Yes} \rangle$ is connected to $\langle j, \text{Yes} \rangle$ under $q_j = 0$ and to $\langle j, \text{No} \rangle$ under $q_j = 1$; if $x_j \notin D(\Delta) \cup K(\Delta)$ then $\langle j - 1, \text{Yes} \rangle$ is connected to $\langle j, \text{Yes} \rangle$, and $\langle j - 1, \text{No} \rangle$ is connected to $\langle j, \text{No} \rangle$, in either case. D_n is determined by a recursive approach in which the generic procedure $\text{fool}(l, l')$ constructs a distribution that fools the transition probabilities between level l and l' in *all* the RFAs as follows. It computes, using $\text{fool}(l, l'')$ and $\text{fool}(l'', l')$ recursively, distributions D_1 and D_2 , each of size at most $E_0(1 + o(1))$, that fool the transitions between states in levels l and $l'' = \lfloor (l + l')/2 \rfloor$, and between states in levels l'' and l ; a procedure $\text{reduce}(D_1 \times D_2)$ then combines D_1 and D_2 into a distribution D of size at most $E_0(1 + o(1))$ that fools the transitions between states in levels l and l' in all the RFAs. Let $\tilde{D} = D_1 \times D_2$ be the product distribution with $\text{support}(\tilde{D}) = \{w_1 w_2 : w_i \in \text{support}(D_i)\}$ and $\Pr_{\tilde{D}}\{w_1 w_2\} = \Pr_{D_1}\{w_1\} \Pr_{D_2}\{w_2\}$, a randomized version of reduce is to retain each $w \in \tilde{D}$ with probability $q(w) = E_0/|\text{support}(\tilde{D})|$ into $\text{support}(D)$ with $\Pr_D\{w\} = \Pr_{\tilde{D}}\{w\}/q(w)$. Thus, for all pairs of states s, t in the RFAs the transition probabilities are preserved in expectation:

$$\mathbb{E}[\Pr_D\{s \rightarrow t\}] = \sum_{w : s \xrightarrow{w} t} \frac{\Pr_{\tilde{D}}\{w\}}{q(w)} q(w) = \Pr_{\tilde{D}}\{s \rightarrow t\}, \quad (40.6.2)$$

where the sum is over all the strings w that lead from state s to state t . This selection also implies that the expected size of $\text{support}(D)$ is $\sum_w q(w) = E_0$. This randomized combining can be derandomized using a 2-wise independent probability space. The bottom of the recursion is reached when the number of levels between l and l' is at most $\log E_0$, and then the distribution (of size E_0) is implemented by $\log E_0$ unbiased bits. E_0 depends polynomially on $1/\delta$, where δ is the relative error that is allowed for the transition probabilities. Taking δ as a small constant suffices to obtain a constant approximation of the moment bounds.

Similarly, this method can be used to compute, fast and in parallel, $(1/r)$ -nets and $(1/r)$ -approximations whose sizes nearly match the probabilistic bounds.

APPLICATIONS

Some interesting examples for which optimal deterministic algorithms have been obtained using derandomization are the following:

- *Convex hulls*: The only optimal deterministic algorithm for the computation of the convex hull of n points in \mathbb{R}^d space is the derandomization of a randomized-incremental algorithm. The reader is referred to [BCM99] for the details (this reference is much more readable than the original paper [Cha93b]) ([Chapter 22](#)).
- *Output-sensitive convex hull in \mathbb{R}^d* : An optimal algorithm was obtained us-

ing derandomization [CM95]; afterward a surprisingly simple solution avoiding derandomization was found [Cha96].

- *Diameter of a point set in \mathbb{R}^d :* After a sequence of improvements, an optimal algorithm using derandomization was found [Ram01]. Currently, the best solution that avoids derandomization has a running time with an extra $\log n$ factor [Bes98].
 - *Linear programming:* In \mathbb{R}^d , the best deterministic solution is achieved through derandomization and has running time $O(C_d n)$ with $C_d = \exp(O(d \log d))$ [CM96]; in contrast, with randomization $C_d \approx \exp(\sqrt{d})$ is possible [MSW96].
 - *Segment intersection:* A first algorithm for reporting segment intersections in linear space and optimal time used derandomization [AGR95], followed shortly by a relatively simple algorithm that avoids derandomization [Bal95]. Optimal parallel algorithms have been obtained with derandomization and have not been matched by other approaches.
-

OPEN PROBLEMS

Is derandomization truly necessary to obtain an optimal algorithm in cases such as the convex hull in d dimensions or the diameter in dimension 3?

40.7 DETERMINISTIC APPROXIMATIONS, NETS, AND CUTTINGS

APPROXIMATIONS

Method of conditional probabilities. Direct application of the method leads to a sequential polynomial-time construction of approximations with optimal size (almost matching the probabilistic bound). A more efficient construction [Mat95, CM96, BCM99] is based on the following two properties, which are easily verified:

- **Separability:** Let $X_1, X_2 \subseteq X$ be disjoint subsets of (almost) equal cardinality with $X = X_1 \cup X_2$, and let A_i be an ϵ -approximation for (X_i, Γ_{X_i}) , for $i = 1, 2$. Then $A = A_1 \cup A_2$ is an ϵ -approximation for (X, Γ) .
- **Composition:** Let A_1 be an ϵ_1 -approximation for (X, Γ) and let A_2 be an ϵ_2 -approximation for (A_1, Γ_{A_1}) . Then A_2 is an $(\epsilon_1 + \epsilon_2)$ -approximation for (X, Γ) .

Let **Poly-Approx** (X, r) be an algorithm that computes a $(1/r)$ -approximation for X of size $Cr^2 \log r$ using time $O(n^{D+1})$, where $n = |X|$. The two properties naturally lead to a divide-and-conquer approach to compute an approximation:

Approx (X, r) :

If $|X| \leq Cr^2 \log r$ then return X .

Split X into two subsets X_1 and X_2 of (almost) equal size.

Let $A_1 \leftarrow \text{Approx}(X_1, r)$; $A_2 \leftarrow \text{Approx}(X_2, r)$.

Return **Poly-Approx** $(A_1 \cup A_2, r)$.

Approx returns a $(1/r')$ -approximation, where $r' = r / \log n$ (an approximation

factor $1/r$ accumulates over each of the $\log n$ levels) of size $Cr^2 \log r$. The running time is dominated by the work required at the bottom of the recursion and so is $O(n \cdot (r^2 \log r)^D)$. To obtain the $(1/r)$ -approximation that we actually want it is necessary to decrease the approximation factor as the computation progresses from the bottom to the top of the recursion tree (starting with $1/r$ at the leaves). This is possible without affecting the running time because the total size of the sets that are handled in each level is decreasing. This approach also leads to fast parallel algorithms, noting that the separability property can be extended to many sets X_i .

Construction via simplicial partitions. For linearizable range spaces, including most applications, a more efficient construction is possible [Mat91b]. Let P be a set of n points in the plane, a simplicial (triangle) partition is a tuple $\Pi = ((P_1, \Delta_1), \dots, (P_m, \Delta_m))$ where the P_i 's (classes) form a disjoint partition of P and Δ_i is a triangle (simplex) with $P_i \subseteq \Delta_i$. The **crossing number** of Π is defined as the maximum number of Δ_i 's that can be simultaneously crossed by a line. Simplicial partitions with small crossing number exist and can be computed efficiently:

THEOREM 40.7.1 *Simplicial Partition Theorem*

Let P be a set of n points in \mathbb{R}^2 , let s be an integer parameter with $2 \leq s < n$, and let $r = n/s$. Then there exists a partition $\Pi = (P_1, \dots, P_m)$ of P whose classes satisfy $s \leq |P_i| < 2s$ (so $m = \Theta(r)$) and whose crossing number is $O(r^{1/2})$. Furthermore, the construction can be performed in time $O(n \log r)$ for $s \geq n^\delta$, any $\delta > 0$.

The construction makes heavy use of cuttings. Given a set P of n points, the partition Π for parameter s with crossing number $\kappa = O(r^{1/2})$, guaranteed by the theorem, can be used to construct an approximation A (with respect to triangles) by taking an arbitrary point from each class into A : Let Δ be a triangle, the number of classes intersecting the boundary of Δ is at most 3κ . Therefore, we obtain the following bound for the error:

$$\left| \frac{|A \cap \Delta|}{|A|} - \frac{|P \cap \Delta|}{|P|} \right| = O\left(\frac{\kappa}{m}\right) = O\left(\frac{1}{r^{1/2}}\right).$$

Thus, this approach provides a $(1/r)$ -approximation of size $O(r^2)$. This is interesting, of course, only when r is small, $r \leq n^{1/2}$. The construction time is $O(n \log r)$.

NETS

Method of conditional probabilities. Again, direct application of the method leads to a polynomial-time construction of a $(1/r)$ -net of size $O(r \log r)$. A more efficient construction uses the following observation: A $(1/2r)$ -net for a $(1/2r)$ -approximation is a $(1/r)$ -net. Therefore, efficient algorithms for computing approximations translate into efficient algorithms for computing nets. The running time is dominated by the computation of the $(1/2r)$ -approximation.

Construction using sensitive approximations. A random sample of size $O(r^2 \log r)$ satisfies the bound [BCM99]:

$$\left| \frac{|A \cap R|}{|A|} - \frac{|P \cap R|}{|P|} \right| \leq \frac{1}{2r} \left(\sqrt{\frac{|R|}{|X|}} + \frac{1}{r} \right).$$

Such a sample is called a *sensitive* $(1/r)$ -approximation. Sensitive approximations have the separability and composition properties, and can be computed in time $O(n \cdot (r^2 \log r)^D)$ as usual approximations. Now, a sensitive $(1/r)$ -approximation is also a $(1/r)^2$ -net, so a $(1/r)$ -net of size $O(r \log r)$ can be computed in time $O(n \cdot (r \log r)^D)$. For a linearizable range space and r small, construction in time $O(n \log r)$ is possible via approximations based on simplicial partitions.

Construction using greedy-cover algorithm. The problem of finding a $(1/r)$ -net for a range space (X, Γ) can be posed as the problem of finding a vertex cover for a hypergraph. The latter problem is solved by Lovász's *greedy cover algorithm*, resulting in a sample N of size $s = O(r \log n)$. A net of size $O(r \log r)$ can be constructed with a modified version of the greedy algorithm [CF90], without recurring to derandomization at all.

CUTTINGS

Derandomizing constructions. The method of splitting the excess (Section 40.1) can be derandomized in polynomial time to obtain a $(1/r)$ -cutting of optimal size. More efficiently, one can first compute a $(1/2r)$ -approximation and then a $(1/2r)$ -cutting for the approximation. This still does not lead to an optimal running time.

For the case of hyperplane arrangements, cuttings of optimal size can be computed in optimal time using a hierarchical subdivision [Cha93a]. It uses nets that are sensitive: as expected from a random sample, the number of vertices in the net is proportional to the number of vertices in the original set of hyperplanes. The idea is that the number of “spurious” vertices introduced by the hierarchical subdivision remains asymptotically smaller than the number of vertices actually in the arrangement, as in the “global accounting” technique of Section 40.4.

Direct constructions. A cutting for lines in the plane can be obtained without derandomization by splitting the levels in the arrangement of the lines [Mat98]. In higher dimensions, a direct construction is possible that follows the prune-and-search approach of Megiddo [Mat91a, DS00]; this method produces cuttings of size much larger than optimal.

DETERMINISTIC SAMPLING

The basic divide-and-conquer approach uses a sample of size r satisfying the higher-moment bound. Such a “ $(1/r)$ -sample” can be constructed deterministically in polynomial time via derandomization. A more efficient construction first constructs a $(1/2r)$ -approximation, and then a $(1/2r)$ -sample for the approximation. The result is a $(1/r)$ -sample. If $r \leq n^\delta$ for certain $\delta > 0$, then the construction time is dominated by the computation of the $(1/2r)$ -approximation.

OPEN PROBLEMS

Can one obtain deterministically in polynomial time a sample out of a configuration space satisfying axiom (iii) (but not (iii')), so that the higher-moment bound holds?

Can samples be obtained in parallel nearly satisfying the probabilistic bounds using a general approach that does not tailor the space and distribution to the particular algorithm and input?

40.8 OPTIMAL DIVIDE-AND-CONQUER

Divide-and-conquer based on sampling, either random or deterministic, rarely results directly in optimal algorithms. As discussed in Section 40.1, the running time includes an extra factor n^ϵ in most cases. If the size r of the sample is a function of n , say $r = n^\delta$, then the extra factor can often be reduced to $\log^\epsilon n$.

The main problem is the fact that in principle the total conflict list size can grow beyond what is permissible for optimality. We illustrate some tricks that are useful to avoid this, using our running example, the computation of the trapezoidal map of n segments S in the plane.

Pruning. We consider the case in which the segments in S are non-intersecting. We want to enforce that the total conflict list size remains $O(n)$. We use a $(1/r)$ -cutting with $r = n^\delta$ for $\delta > 0$ appropriately small. For every trapezoid $\Delta \in \mathcal{T}(R)$, we determine the list S_Δ of segments intersecting Δ . Instead of directly recursing on S_Δ , we determine those segments in S_Δ that have an endpoint in Δ , and those that “cross” Δ , that is, intersect it without having an endpoint in Δ . Some of the crossing segments are then used to further subdivide Δ into *noncrossing* trapezoids Δ_n for which S_{Δ_n} has only noncrossing segments, and *crossing* trapezoids Δ_c for which S_{Δ_c} has only crossing segments. This takes time $O(n_\Delta \log n_\Delta)$, and we then recurse on the smaller trapezoids Δ_n . In a crossing trapezoid, the segments cross without intersecting and so they form a “stair” from which the output can be produced directly. The total conflict size for noncrossing trapezoids is at most $2n$ at each level, and each level generates crossing trapezoids with total conflict list size $O(n)$. Since the size of a subproblem at the i th level of recursion is at most $n_i = n^{(1-\epsilon)^i}$, then $\sum_i \log n_i = O(\log n)$ and the total construction time is $O(n \log n)$. This technique has been used in the deterministic computation of 2D Voronoi diagrams [RS92, AGR94].

Global accounting. We consider the case in which all the segments intersect. Let T_0 consist of a single sufficiently large trapezoid containing all the segments. Given $\Delta \in T_{i-1}$, if $|S_\Delta| \leq n/r_0^i$ then Δ is not subdivided and remains in T_i unaltered, otherwise Δ is subdivided using a sample N_Δ of size $C_N r_0 \log r_0$, where r_0 is a sufficiently, such that (i) R_Δ is a $(1/r_0)$ -net for S_Δ and (ii) the number of vertices of the arrangement of R_Δ inside Δ is at most twice the expected number for a random sample, that is $2(C_N r_0 \log r_0 / |S_\Delta|)^2 v(S, \Delta)$, where $v(S, \Delta)$ is the number of vertices of S inside Δ . The size of the decomposition in Δ is proportional to the number of vertices of R_Δ inside Δ plus the number of intersections of R_Δ with the boundary of Δ . Thus, assuming R_Δ has properties (i) and (ii),

$$\begin{aligned} |T_i| &\leq \sum_{\Delta \in T_{i-1}} \left\{ A \left(\frac{r_0 \log r_0}{n_\Delta} \right)^2 v(S, \Delta) + B r_0 \log r_0 \right\} \\ &\leq A \left(\frac{r_0 \log r_0}{n/r_0^i} \right)^2 \sum_{\Delta \in T_{i-1}} v(S, \Delta) + B \sum_{\Delta \in T_{i-1}} r_0 \log r_0 \\ &\leq A r_0^{2i+2} \log^2 r_0 + B r_0 \log r_0 |T_{i-1}|, \end{aligned}$$

using that $|S_\Delta| \geq n/r_0^i$ and $\sum_{\Delta \in T_{i-1}} v(S, \Delta) \leq n^2$, the total number of vertices in the arrangement. The solution to this recurrence is $|T_i| = O(r_0^{2i+2} \log^2 r_0)$,

and so $|T_l| = O(n^2)$ for the last level l . This approach can be derandomized, the verification of the number of vertices in the sample is then made with the help of a suitable approximation (Section 40.7). This approach has been used to compute cuttings of hyperplanes [Cha93a].

Clustering. We consider again the case of non-intersecting segments. The idea is to group trapezoids into subproblems so that the size of the boundary between subproblems is small and, hence, the number of elements shared by subproblems is small. Let $\mathcal{T} = \mathcal{T}(R)$ be the trapezoidal decomposition for a sample $R \subseteq S$ and let $\mathcal{G} = (V, E)$ be the dual graph of \mathcal{T} : nodes in \mathcal{G} correspond to trapezoids in \mathcal{T} , and two nodes are connected by an arc if the corresponding trapezoids share a vertical edge. The separator theorem for planar graphs guarantees the existence of a set of $O(\sqrt{|V|})$ nodes that separates the graph into two disconnected subgraphs each with at most $1/2$ of the nodes and, because of the bounded vertex degree, this implies an arc separator of the same size. The clustering is performed by iterating the separator theorem, until *clusters* (groups) consisting of at most t nodes are obtained. Thus, with $l = \lceil \log(|V|/t) \rceil$ the total separator size is

$$O\left(\sum_{i=0}^l 2^i \left(\frac{|V|}{2^i}\right)^{1/2}\right) = O\left(\frac{|V|}{t^{1/2}}\right).$$

Let Λ be the set of resulting clusters. For $\square \in \Lambda$, let S_\square denote the conflict list of \square with respect to S (the set of $s \in S$ that intersect \square). Let's assume that $|S_\Delta| \leq \kappa$ for $\Delta \in \mathcal{T}$. To bound $\sum_{\square \in \Lambda} |S_\square|$, the total conflict list size for the clustering, note that a segment that does not intersect boundaries between clusters conflicts with only one cluster. Thus,

$$\sum_{\square \in \Lambda} |S_\square| = n + O\left(\frac{|V|}{t^{1/2}} \cdot \kappa\right).$$

In our application R is a $(1/r)$ -net, so $|V| = O(r \log r)$ and $\kappa = n/r$. Choosing $t = r^{1/2}$, we have

$$\sum_{\square \in \Lambda} |S_\square| = n \cdot \left(1 + O\left(\frac{\log r}{r^{1/4}}\right)\right),$$

and for each $\square \in \Lambda$, $|S_\square| \leq \kappa \cdot t = n/r^{1/2}$. Choosing $r = r(n)$ sufficiently large, $r = \Omega(\log^c n)$ for some constant c , the total subproblem size remains $O(n)$ through all levels of the recursive computation. This approach has been used for computing in parallel convex-hulls in \mathbb{R}^3 [DDD⁺95] and for computing optimally the diameter in \mathbb{R}^3 [Ram01].

40.9 OPTIMIZATION

In some problems that seek to optimize some parameter r , randomization is useful to perform efficiently a search for the optimal value r^* without explicitly building the space in which the search is performed. Normally, the search is among the vertices in an arrangement too large to build explicitly. To obtain a deterministic algorithm, derandomized sampling is often used in combination with **parametric search** (Chapter 43): Derandomization is used to obtain an algorithm that decides

whether the optimal value r^* is larger than, equal to, or smaller than a given value r , and also another *generic* algorithm whose computation for the value r^* is different than for any other r . The decision algorithm is then used as a guiding oracle to run the generic algorithm until the value r^* is determined. For efficiency, the generic algorithm must perform comparisons that involve r in few batches. Several examples of this application can be found in [CEGS93, AST92]. In some cases parametric search has been successfully replaced with a search in an appropriate cutting of the arrangement in which the search is performed. The *slope selection* problem is an example of this [BC98].

40.10 BETTER GUARANTEES

Bounds for the expected performance of randomized algorithms are usually available. Sometimes stronger results are desired. If the analysis of the algorithm cannot be extended to provide such bounds, then some techniques may help to achieve them:

Randomized space vs. deterministic space. Any randomized algorithm using expected space S and expected time T can be converted to an algorithm that uses deterministic space $2S$, and whose expected running time is at most $2T$. We simply need to maintain a count of the memory allocated by the algorithm. Whenever it exceeds $2S$, we stop the computation and restart it again with fresh choices for the random variables. The expected number of retrials is one.

Tail estimates. The knowledge that the expected running time of a given program is one second does not exclude the possibility that it sometimes takes one hour. Markov's inequality implies that the probability that this happens is at most $1/3600$. While this seems innocuous, it implies that it is likely to occur if we repeat this particular computation, say, 10000 times.

For randomized incremental construction, better tail estimates are available only for the *space complexity* [CMS93, MSW93b], and for the running time of segment intersection in the plane [MSW93a] and LP-type optimization [GW00]. In other cases, one can still apply a simple modification to the algorithm to yield a stronger bound. We run it for two seconds. If it does not finish the computation within two seconds, then we abandon the computation and restart with fresh choices for the random variables. Clearly, the probability that the algorithm does not terminate within one hour is at most 2^{-1800} . Alt et al. [AGM⁺96] work out this technique in detail.

40.11 PROBABILISTIC PROOF TECHNIQUES

Randomized algorithms are related to probabilistic proofs and constructions in combinatorics, which precede them historically. Conversely, the concepts developed to design and analyze randomized algorithms in computational geometry can be used as tools in proving purely combinatorial results. Many of these results are based on the following theorem:

THEOREM 40.11.1

Let (X, \mathcal{T}, D, K) be a configuration space satisfying axioms (i), (ii), (iii), and (iii') of Section 40.5. For $S \subseteq X$ and $0 \leq k \leq n$, let

$$\Pi^k(S) := \{\Delta \in \Pi(X) \mid |K(\Delta) \cap S| \leq k\}$$

denote the set of configurations with at most k conflicts in S .

Then $|\Pi^k(S)| = O(k^d)E[|\mathcal{T}(R)|]$, where R is a random sample of S of size n/k , and d is as in axiom (i).

Note that $\Pi^0(S) = \mathcal{T}(S)$. The theorem relates the number of configurations with at most k conflicts to those without conflict.

An immediate application is to prove a bound on the number of vertices of level at most k in an arrangement of lines in the plane (the level of a vertex is the number of lines lying above it; see [Section 21.2](#)). We define a configuration space (X, \mathcal{T}, D, K) where X is the set of lines, $\mathcal{T}(S)$ is the set of vertices of the upper envelope of the lines, $D(\Delta)$ are the two lines forming the vertex Δ (so $d = 2$), and $K(\Delta)$ is the set of lines lying above Δ . Theorem 40.11.1 implies that the number of vertices of level up to k is bounded by $O(nk)$. The same argument works in any dimension.

Sharir and others have proved a number of combinatorial results using this technique [Sha94, AES99, ASS96, SS97]. They define a configuration space and need to bound $|\mathcal{T}(S)|$. They do this by proving a geometric relationship between the configurations with zero conflicts (the ones appearing in $\mathcal{T}(S)$) and the configurations with at most k conflicts. Applying Theorem 40.11.1 yields a recursion that bounds $|\mathcal{T}(S)|$ in terms of $|\mathcal{T}(R)|$. A refined approach that uses a sample of size $n - 1$ (instead of n/k) has been suggested by Tagansky [Tag96].

Sharir [Sha01] reviews this technique and gives a new proof for Theorem 40.11.1 based on the *Crossing lemma* ([Chapter 24](#)).

40.12 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[Cla92, Mul00]: General surveys of randomized algorithms in computational geometry.

[Sei93]: An introduction to randomized incremental algorithms using backwards analysis.

[GS93]: Surveys computations with arrangements, including randomized algorithms.

[AS01] Surveys randomized techniques in geometric optimization problems.

[Aga91]: A survey on geometric partitions.

[Mul93]: This monograph is an extensive treatment of randomized algorithms in computational geometry.

[Mat00]: An introduction to derandomization for geometric algorithms, with many references.

[Kar91, MR95]: A survey and a book on randomized algorithms and their analysis in computer science, including derandomization techniques.

[AS92]: This monograph is a good reference on probabilistic proof techniques in combinatorics. It also deals with derandomization.

RELATED CHAPTERS

Because randomized algorithms have been used successfully in nearly all areas of computational geometry, they are mentioned throughout Parts C and D of this Handbook. Areas where randomization plays a particularly important role include:

- [Chapter 22: Convex hull computations](#)
 - [Chapter 24: Arrangements](#)
 - [Chapter 36: Range searching](#)
 - [Chapter 45: Linear programming](#)
-

REFERENCES

- [AES99] P.K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29:912–953, 1999.
- [Aga91] P.K. Agarwal. Geometric partitioning and its applications. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: Papers from the DIMACS Special Year*. Amer. Math. Soc., Providence, 1991.
- [AGM⁺96] H. Alt, L.J. Guibas, K. Mehlhorn, R.M. Karp, and A. Wigderson. A method for obtaining randomized algorithms with small tail probabilities. *Algorithmica*, 16:543–547, 1996.
- [AGR94] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Parallel algorithms for higher-dimensional convex hulls. In *Proc. 35th Annu. IEEE Symp. Found. Comput. Sci.*, pages 683–694, 1994.
- [AGR95] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Computing faces in segment and simplex arrangements. In *Proc. 27th Annu. ACM Symp. Theory Comput.*, pages 672–682, 1995.
- [AM94] P.K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.
- [AMS98] P.K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. *SIAM J. Comput.*, 27:491–505, 1998.
- [AS92] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley & Sons, New York, 1992.
- [AS01] P.K. Agarwal and S. Sen. Randomized algorithms for geometric optimization problems. In P.P. and. Rajasekaran, J.H. Reif, and J. Rolim, editors, *Handbook of Randomization*, pages 151–201. Kluwer Academic, Boston, 2001.
- [ASS96] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.
- [AST92] P.K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric searching in geometric optimization. In *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pages 72–82, 1992.

- [Bal95] I.J. Balaban. An optimal algorithm for finding segment intersections. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 211–219, 1995.
- [BC98] H. Brönnimann and B. Chazelle. Optimal slope selection via cuttings. *Comput. Geom. Theory Appl.*, 10:23–29, 1998.
- [BCM99] H. Brönnimann, B. Chazelle, and J. Matoušek. Product range spaces, sensitive sampling, and derandomization. *SIAM J. Comput.*, 28:1552–1575, 1999.
- [BDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.
- [Bes98] S.N. Bespamyatnikh. An efficient algorithm for the three-dimensional diameter problem. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 137–146, 1998.
- [BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 276–287, 1994.
- [BRS89] B. Berger, J. Rompel, and P.W. Shor. Efficient NC algorithms for set cover with applications to learning and geometry. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, volume 30, pages 54–59, 1989.
- [CEGS93] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Diameter, width, closest line pair and parametric searching. *Discrete Comput. Geom.*, 10:183–196, 1993.
- [CF90] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10:229–249, 1990.
- [Cha93a] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9:145–158, 1993.
- [Cha93b] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.
- [Cha96] T.M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete Comput. Geom.*, 16:361–368, 1996.
- [Cla92] K.L. Clarkson. Randomized geometric algorithms. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 117–162. World Scientific, Singapore, 1992.
- [CM95] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. *Comput. Geom. Theory Appl.*, 5:27–32, 1995.
- [CM96] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:579–597, 1996.
- [CMS93] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3:185–212, 1993.
- [DDD⁺95] F. Dehne, X. Deng, P. Dymond, A. Fabri, and A.A. Khokhar. A randomized parallel 3D convex hull algorithm for coarse grained multicomputers. In *Proc. 7th Annu. ACM Sympos. Parallel Algorithms Architect.*, pages 27–33, 1995.
- [DS00] M. Dyer and S. Sen. Fast and optimal parallel multidimensional search in PRAMS with applications to linear programming and related problems. *SIAM J. Comput.*, 30:1443–1461, 2000.
- [GS93] L.J. Guibas and M. Sharir. Combinatorics and algorithms of arrangements. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 9–36. Springer-Verlag, Heidelberg, 1993.
- [GW00] B. Gärtner and E. Welzl. Random sampling in geometric optimization: New insights and applications. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 91–99, 2000.

- [HP00] S. Har-Peled. Constructing planar cuttings in theory and practice. *SIAM J. Comput.*, 29:2016–2039, 2000.
- [Jof74] A. Joffe. On a set of almost deterministic k -independent random variables. *Ann. Probab.*, 2:161–162, 1974.
- [Kar91] R.M. Karp. An introduction to randomized algorithms. *Discrete Appl. Math.*, 34:165–201, 1991.
- [KK94] D. Karger and D. Koller. (De)randomized construction of small sample spaces in NC. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 252–263, 1994.
- [KM93] D. Koller and N. Megiddo. Constructing small sample spaces satisfying given constraints. In *Proc. 25th Annu. ACM Sympos. Theory Comput.*, pages 268–277, 1993.
- [KM94] H. Karloff and Y. Mansour. On construction of k -wise independent random variables. In *Proc. Annu. ACM Sympos. Theory Comput.*, pages 564–573, 1994.
- [Mat91a] J. Matoušek. Computing the center of planar point sets. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: Papers from the DIMACS Special Year*, pages 221–230. Amer. Math. Soc., Providence, 1991.
- [Mat91b] J. Matoušek. Efficient partition trees. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 1–9, 1991.
- [Mat95] J. Matoušek. Approximations and optimal geometric divide-and-conquer. *J. Comput. Syst. Sci.*, 50:203–208, 1995.
- [Mat98] J. Matoušek. On constants for cuttings in the plane. *Discrete Comput. Geom.*, 20:427–448, 1998.
- [Mat00] J. Matoušek. Derandomization in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 559–595. Elsevier North-Holland, Amsterdam, 2000.
- [MNN89] R. Motwani, J. Naor, and M. Naor. The probabilistic method yields deterministic parallel algorithms. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 8–13, 1989.
- [MR95] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [MRS01] S. Mahajan, E.A. Ramos, and K.V. Subramanyam. Solving some discrepancy problems in NC. *Algorithmica*, 29:371–395, 2001.
- [MS96] J. Matoušek and O. Schwarzkopf. A deterministic algorithm for the three-dimensional diameter problem. *Comput. Geom. Theory Appl.*, 6:253–262, 1996.
- [MSW90] J. Matoušek, R. Seidel, and E. Welzl. How to net a lot with little: Small ϵ -nets for disks and halfspaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 16–22, 1990.
- [MSW93a] K. Mehlhorn, M. Sharir, and E. Welzl. Tail estimates for the efficiency of randomized incremental algorithms for line segment intersection. *Comput. Geom. Theory Appl.*, 3:235–246, 1993.
- [MSW93b] K. Mehlhorn, M. Sharir, and E. Welzl. Tail estimates for the space complexity of randomized incremental algorithms. *Comput. Geom. Theory Appl.*, 4:185–246, 1993.
- [MSW96] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.
- [Mul93] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1993.

- [Mul96] K. Mulmuley. Randomized geometric algorithms and pseudorandom generators. *Algorithmica*, 16:450–463, 1996.
- [Mul00] K. Mulmuley. Randomized algorithms in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 703–724. Elsevier North-Holland, Amsterdam, 2000.
- [Nis92] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12:449–461, 1992.
- [NN90] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. In *Proc. 29th Annu. ACM Sympos. Theory Comput.*, pages 213–223, 1990.
- [Rag88] P. Raghavan. Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *J. Comput. Syst. Sci.*, 37:130–143, 1988.
- [Ram01] E.A. Ramos. An optimal deterministic algorithm for computing the diameter of a three-dimensional point set. *Discrete Comput. Geom.*, 26:233–244, 2001.
- [RS92] J.H. Reif and S. Sen. Optimal parallel randomized algorithms for three-dimensional convex hulls and related problems. *SIAM J. Comput.*, 21:466–485, 1992.
- [Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1:51–64, 1991.
- [Sei93] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms Combin.*, pages 37–68. Springer-Verlag, Berlin, 1993.
- [Sha94] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.
- [Sha01] M. Sharir. The Clarkson-Shor technique revisited and extended. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 252–256, 2001.
- [Spe87] J. Spencer. *Ten lectures on the probabilistic method*. CBMS-NSF. SIAM, 1987.
- [SS97] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a three-dimensional arrangement of surfaces and its applications. *Discrete Comput. Geom.*, 18:269–288, 1997.
- [SSS93] J.P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 331–340, 1993.
- [Tag96] B. Tagansky. A new technique for analyzing substructures in arrangements of piecewise linear surfaces. *Discrete Comput. Geom.*, 16:455–479, 1996.
- [VC71] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16:264–280, 1971.
- [YY85] A.C. Yao and F.F. Yao. A general approach to D -dimensional geometric queries. In *Proc. 17th Annu. ACM Sympos. Theory Comput.*, pages 163–168, 1985.

41 ROBUST GEOMETRIC COMPUTATION

Chee K. Yap

INTRODUCTION

Nonrobustness refers to qualitative or catastrophic failures in geometric algorithms arising from numerical errors. Section 41.1 provides background on these problems. Although nonrobustness is already an issue in “purely numerical” computation, the problem is compounded in “geometric computation.” In Section 41.2 we characterize such computations. Researchers trying to create robust geometric software have tried two approaches: making fixed-precision computation robust (Section 41.3), and making the exact approach viable (Section 41.4). Another source of nonrobustness is the phenomenon of degenerate inputs. General methods for treating degenerate inputs are described in Section 41.5.

41.1 NUMERICAL NONROBUSTNESS ISSUES

Numerical nonrobustness in scientific computing is a well-known and widespread phenomenon. The root cause is the use of *fixed-precision numbers* to represent real numbers, with precision usually fixed by the machine word size (e.g., 32 bits). The unpredictability of floating-point code across architectural platforms in the 1980’s was resolved through a general adoption of the IEEE standard 754-1985. But this standard only makes program behavior predictable and consistent across platforms; the errors are still present. Ad hoc methods for fixing these errors (such as treating numbers smaller than some ϵ as zero) cannot guarantee their elimination.

If nonrobustness is problematic in purely numerical computation, it apparently becomes intractable in “geometric” computation. In Section 41.2, we elucidate the concept of geometric computations. Based on this understanding, we conclude that nonrobustness problems within fixed-precision computation cannot be solved by purely arithmetic solutions (better arithmetic packages, etc.). Rather, a suitable *fixed-precision geometry* is needed to substitute for the original geometry (which is usually Euclidean). We describe such approaches in Section 41.3.

In Section 41.4, we describe the *exact approach* for achieving robust geometric computation. This demands some type of *big number* package as well as further considerations. Indeed, current research is converging on an exciting new form of computational model that we may call *guaranteed precision computation*.

In Section 41.5, we address a different but common cause of numerical nonrobustness, namely, *data degeneracy*. Although this problem has some connection to fixed-precision arithmetic, it is an issue even with the exact approach.

GLOSSARY

Fixed-precision computation: A mode of computation in which every number is represented using some fixed number L of bits, usually 32 or 64. For floating point numbers, L is partitioned into $L = L_M + L_E$ for the mantissa and the exponent respectively. **Double-precision mode** is a relaxation of fixed precision: the intermediate values are represented in $2L$ bits, but these are finally truncated back to L bits.

Nonrobustness: The property of code failing on certain kinds of inputs. Here we are mainly interested in nonrobustness that has a numerical origin: the code fails on inputs containing certain patterns of numerical values. Degenerate inputs are just extreme cases of these “bad patterns.”

Benign vs. catastrophic errors: Fixed-precision numerical errors are fully expected and so are normally considered to be “benign.” In purely numerical computations, errors become “catastrophic” when there is a severe loss of precision. In geometric computations, errors are “catastrophic” when the computed results are qualitatively different from the true answer (e.g., the combinatorial structure is wrong) or when they lead to unexpected or *inconsistent* states of the programs.

Big number packages: Software packages for representing arbitrary precision numbers (usually integers or rational numbers), and in which some basic operations on these numbers are performed exactly. For instance, $+, -, \times$ are implemented exactly with *BigIntegers*. With *BigRationals*, division can also be exact. Other operations such as $\sqrt{}$ still need approximations or rounding.

41.2 THE NATURE OF GEOMETRIC COMPUTATION

If the root cause of numerical nonrobustness is arithmetic, then it may appear that the problem can be solved with the right kind of arithmetic package. We may roughly divide the approaches into two camps, depending on whether one uses finite precision arithmetic or insists on exactness (or at least the possibility of computing to arbitrary precision). While arithmetic is an important topic in its own right, our focus here will be on geometric rather than purely arithmetic approaches for achieving robustness.

To understand why nonrobustness is especially problematic for geometric computation, we need to understand what makes a computation “geometric.” Indeed, we are revisiting the age-old question “*What is Geometry?*” that has been asked and answered many times in mathematical history, by Euclid, Descartes, Hilbert, Dieudonné and others. But as in many other topics, the perspective stemming from modern computational viewpoint sheds new light. Geometric computation clearly involves numerical computation, but there is something more. We use the aphorism **GEOMETRIC = NUMERIC + COMBINATORIAL** to capture this. Instead of “combinatorial” we could have substituted “discrete” or sometimes “topological.” What is important is that this combinatorial part is concerned with discrete relations among geometric objects. Examples of discrete relations are “a point lies on a line,” “a point lies inside a simplex,” “two disks intersect.” The geometric

objects here are points, lines, simplices and disks. Following Descartes, each object is defined by numerical parameters. Each discrete relation is reduced to the truth of suitable numerical inequalities involving these parameters. Geometry arises when such discrete relations are used to characterize configurations of geometric objects.

The mere presence of combinatorial structures in a numerical computation does not make a computation “geometric.” There must be some nontrivial **consistency condition** holding between the numerical data and the combinatorial data. Thus, we would not consider the classical shortest-path problems on graphs to be geometric: the numerical weights assigned to edges of the graphs are not restricted by any consistency condition. Note that common restrictions on the weights (positivity, integrality, etc.) are not consistency restrictions. But the related **Euclidean shortest-path problem** ([Chapter 27](#)) is geometric. See Table 41.2.1 for further examples from well-known problems.

TABLE 41.2.1 Examples of geometric and nongeometric problems.

PROBLEM	GEOMETRIC?
Matrix multiplication, determinant	no
Hyperplane arrangements	yes
Shortest paths on graphs	no
Euclidean shortest paths	yes
Point location	yes
Convex hulls, linear programming	yes
Minimum circumscribing circles	yes

Alternatively, we can characterize a computation as “geometric” if it involves constructing or searching a geometric structure (which may only be implicit). The incidence graph of an arrangement of hyperplanes ([Chapter 24](#)), with suitable additional labels and constraints, is a primary example of such a structure. A **geometric structure** is comprised of four components:

$$D = (G, \lambda, \Phi(\mathbf{z}), I), \quad (41.2.1)$$

where $G = (V, E)$ is a directed graph, λ is a labeling function on the vertices and edges of G , Φ is the consistency predicate, and I the input assignment. Intuitively, G is the combinatorial part, λ the geometric part, and Φ constrains λ based on the structure of G . The **input assignment** is $I : \{z_1, \dots, z_n\} \rightarrow \mathbb{R}$ where the z_i 's are called **structural variables**. We informally identify I with the sequence “ $\mathbf{c} = (c_1, \dots, c_n)$ ” where $I(z_i) = c_i$. The c_i 's are called **(structural) parameters**. For each $u \in V \cup E$, the label $\lambda(u)$ is a Tarski formula of the form $\xi(\mathbf{x}, \mathbf{z})$, where $\mathbf{z} = (z_1, \dots, z_n)$ are the structural variables and $\mathbf{x} = (x_1, \dots, x_d)$. This formula defines a **semialgebraic set** ([Chapter 33](#)) parameterized by the structural variables. For given \mathbf{c} , the semialgebraic set is $f_{\mathbf{c}}(v) = \{\mathbf{a} \in \mathbb{R}^d \mid \xi(\mathbf{a}, \mathbf{c}) \text{ holds}\}$. Following Tarski, we are identifying semialgebraic sets in \mathbb{R}^d with d -dimensional geometric objects. The consistency relation $\Phi(\mathbf{z})$ is another Tarski formula. In practice $\Phi(\mathbf{z})$ has the form $(\forall x_1, \dots, x_d)\phi(\lambda(u_1), \dots, \lambda(u_m))$ where u_1, \dots, u_m ranges over elements of $V \cup E$ and ϕ can be systematically constructed from the graph G .

As an example of this notation, consider an arrangement S of hyperplanes in \mathbb{R}^d . The combinatorial structure $D(S)$ is the incidence graph $G = (V, E)$ of

the arrangement and V is the set of faces of the arrangement. The parameter \mathbf{c} consists of the coefficients of the input hyperplanes. If \mathbf{z} is the corresponding structural parameters then the input assignment is $I(\mathbf{z}) = \mathbf{c}$. The geometric data associates to each node v of the graph the Tarski formula $\lambda(v)$ involving \mathbf{x}, \mathbf{z} . When \mathbf{c} is substituted for \mathbf{z} , then the formula $\lambda(v)$ defines a face $f_{\mathbf{c}}(v)$ (or $f(v)$ for short) of the arrangement. We use the convention that an edge $(u, v) \in E$ represents an “incidence” from $f(u)$ to $f(v)$, where the dimension of $f(u)$ is one more than that of $f(v)$. So $f(v)$ is contained in the closure of $f(u)$. Let $\text{aff}(X)$ denote the affine span of a set $X \subseteq \mathbb{R}^d$. Then $(u, v) \in E$ implies $\text{aff}(f(v)) \subseteq \text{aff}(f(u))$ and $f(u)$ lies on one of the two open halfspaces defined by $\text{aff}(f(u))$. We let $\lambda(u, v)$ be the Tarski formula $\xi(\mathbf{x}, \mathbf{z})$ that defines the open halfspace in $\text{aff}(f(u))$ that contains $f(u)$. Again, let $f(u, v) = f_{\mathbf{c}}(u, v)$ denote this open halfspace. The consistency requirement is that (a) the set $\{f(v) : v \in V\}$ is a partition of \mathbb{R}^d , and (b) for each $u \in V$, the set $f(u)$ is nonempty with an irredundant representation of the form

$$f(u) = \bigcap \{f(u, v) \mid (u, v) \in E\}.$$

Although the above definition is complicated, all of its elements are necessary in order to capture the following additional concepts. We can suppress the input assignment I , so there are only structural variables \mathbf{z} (which is implicit in λ and Φ) but no parameters \mathbf{c} . The triple

$$\widehat{D} = (G, \lambda, \Phi(\mathbf{z}))$$

becomes an *abstract geometric structure*, and $D = (G, \lambda, \Phi(\mathbf{z}), I)$ is an *instance* of \widehat{D} . The structure D in Equation 41.2.1 is *consistent* if the predicate $\Phi(\mathbf{c})$ holds. An abstract geometric structure \widehat{D} is *realizable* if it has some consistent instance. Two geometric structures D, D' are *structurally similar* if they are instances of a common abstract geometric structure. We can also introduce metrics on structurally similar geometric structures: if \mathbf{c} and \mathbf{c}' are the parameters of D, D' then define $d(D, D')$ to be Euclidean norm of $\mathbf{c} - \mathbf{c}'$.

41.3 FIXED-PRECISION APPROACHES

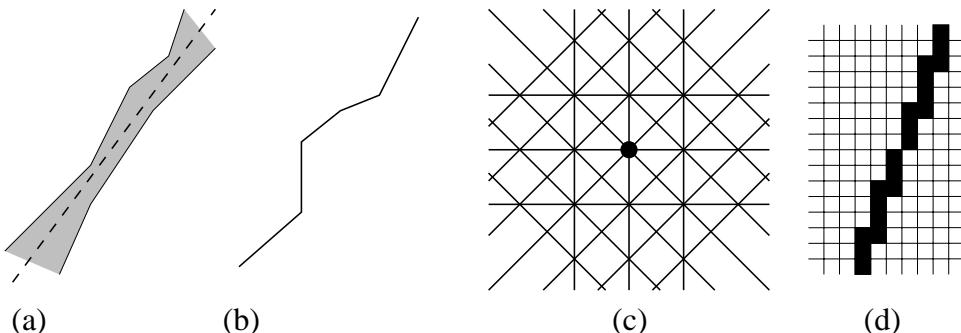
This section surveys the various approaches within the fixed-precision paradigm. Such approaches have strong motivation in the modern computing environment where fast floating point hardware has become a de facto standard in every computer. If we can make our geometric algorithms robust within machine arithmetic, we are assured of the fastest possible implementation. We may classify the approaches into several basic groups. We first illustrate our classification by considering the simple question: “What is the concept of a line in fixed-precision geometry?” Four basic answers to this question are illustrated in [Figure 41.3.1](#) and in [Table 41.3.1](#).

WHAT IS A FINITE-PRECISION LINE?

We call the first approach *interval geometry* because it is the geometric analogue of interval arithmetic. Segal and Sequin [SS85] and others define a zone surrounding the line composed of all points within some ϵ distance from the actual line.

FIGURE 41.3.1

Four concepts of finite-precision lines.



The second approach is called ***topologically consistent distortion***. Greene and Yao [GY86] distorted their lines into polylines, where the vertices of these polylines are constrained to be at grid points. Note that although the “fixed-precision representation” is preserved, the number of bits used to represent these polylines can have arbitrary complexity.

TABLE 41.3.1 Concepts of a finite-precision line.

	APPROACH	SUBSTITUTE FOR IDEAL LINE	SOURCE
(a)	Interval geometry	a line fattened into a tubular region	[SS85]
(b)	Topological distortion	a polyline	[GY86]
(c)	Rounded geometry	a line whose equation has bounded coefficients	[Sug89]
(d)	Discretization	a suitable set of pixels	computer graphics

The third approach follows a tack of Sugihara [Sug89]. An ideal line is specified by a linear equation, $ax + by + c = 0$. Sugihara interprets a “fixed-precision line” to mean that the coefficients in this equation are integer and bounded: $|a|, |b| < K, |c| < K^2$ for some constant K . Call such lines *representable* (see Figure 41.3.1(c) for the case $K = 2$). There are $O(K^4)$ representable lines. An arbitrary line must be “rounded” to the closest (or some nearby) representable line in our algorithms. Hence we call this ***rounded geometry***.

The last approach is based on *discretization*: in traditional computer graphics and in the pattern recognition community, a “line” is just a suitable collection of pixels. This is natural in areas where pixel images are the central objects of study, but less applicable in computational geometry, where compact line representations are desired. This approach will not be considered further in this chapter.

INTERVAL GEOMETRY

In interval geometry, we thicken a geometric object into a zone containing the object. Thus a point may become a disk, and a line becomes a strip between two parallel lines: this is the simplest case and is treated by Segal and Sequin [SS85, Seg90]. They called these “toleranced objects,” and in order to obtain correct predicates, they enforce *minimum feature separations*. To do this, features that are too close must be merged (or pushed apart).

Guibas, Salesin, and Stolfi [GSS89] treat essentially the same class of thick objects as Segal and Sequin, although their analysis is mostly confined to geometric data based on points. Instead of insisting on minimum feature separations, their predicates are allowed to return the DON’T KNOW truth value. Geometric predicates (called ϵ -predicates) for objects are systematically treated in this paper.

In general we can consider zones with nonconstant descriptive complexity, e.g., a planar zone with polygonal boundaries. As with interval arithmetic, a zone is generally a conservative estimate because the precise region of uncertainty may be too complicated to compute or to maintain. In applications where zones expand rapidly, there is danger of the zone becoming catastrophically large: Segal [Seg90] reports that a sequence of duplicate-rotate-union operations repeated eleven times to a cube eventually collapsed it to a single vertex.

TOPOLOGICALLY-CONSISTENT DISTORTION

Sugihara and Iri [SI89b, SIII00] advocates an approach based on preserving topological consistency. These ideas have been applied to several problems, including geometric modeling [SI89a] and Voronoi diagrams for point sets [SI92]. In their approach, one first chooses some topological property (e.g., planarity of the underlying graph) and construct geometric algorithms that preserve the chosen property. Two difficulties in this prescription are (1) how to choose appropriate topological properties, and (2) in what sense does this “work”? Greene and Yao consider the problem of maintaining certain “topological properties” of an arrangement of finite-precision line segments. They introduce polylines as substitutes for ideal line segments in order to preserve certain properties of ideal arrangements (e.g., two line segments intersect in a *connected* subset). Each polyline is a distortion of an ideal segment σ when constrained to pass through the “hooks” of σ (i.e., grid points nearest to the intersections of σ with other line segments). But this may generate new intersections (derived hooks) and the cascaded effects must be carefully controlled. The grid model of Greene-Yao has been taken up by several other authors [Hob99, GM95, GGHT97]. Extension to higher dimensions is harder: there is a solution of Fortune [For98] in 3-dimension. Further developments include the numerically stable algorithms in [FM91]. The interesting twist here is the use of pseudolines rather than polylines.

Hoffmann, Hopcroft, and Karasick [HHK88] address the problem of intersecting polygons in a consistent way. Phrased in terms of our notion of “geometric structure” (Section 41.2) their goal is to compute a combinatorial structure G that is *consistent* in the sense that G is the structure underlying a consistent geometric structure $D = (G, \lambda, \Phi, \mathbf{c}')$. Here, \mathbf{c}' need not equal the actual input parameter vector \mathbf{c} . They show that the intersection of two polygons R_1, R_2 can be efficiently

computed, i.e., a consistent G representing $R_1 \cap R_2$ can be computed. However, in their framework, $R_1 \cap (R_2 \cap R_3) \neq (R_1 \cap R_2) \cap R_3$. Hence they need to consider the triple intersection $R_1 \cap R_2 \cap R_3$. Unfortunately, this operation seems to require a nontrivial amount of geometric theorem proving ability.

This suggests that the problem of verifying consistency of combinatorial structures (the “reasoning paradigm” [HHK88]) is generally hard. Indeed, the NP-hard existential theory of reals can be reduced to such problems. In some sense, the ultimate approach to ensuring consistency is to design “parsimonious algorithms” in the sense of Fortune [For89]. This also amounts to theorem proving as it entails deducing the consequences of all previous decisions along a computation path.

STABILITY

This is a metric form of topological distortion where we place a priori bounds on the amount of distortion. It is analogous to backward error analysis in numerical analysis. Framed as the problem of computing the graph G underlying some geometric structure D (as above, for [HHK88]), we could say an algorithm is *ϵ -stable* if there is a consistent geometric structure $D = (G, \lambda, \Phi, \mathbf{c}')$ such that $\|\mathbf{c} - \mathbf{c}'\| < \epsilon$ where \mathbf{c} is the input parameter vector. We say an algorithm has *strong* (resp. *linear*) stability if ϵ is a constant (resp., $O(n)$) where n is the input size. Fortune and Milenkovic [FM91] provide both linearly stable and strongly stable algorithms for line arrangements. Stable algorithms have been achieved for two other problems on planar point sets: maintaining a triangulation of a point set [For89], and Delaunay triangulations [For92, For95a]. The latter problem can be solved stably using either an incremental or a diagonal-flipping algorithm that is $O(n^2)$ in the worst case. Jaromczk and Wasilkowski [JW94] presented stable algorithms for convex hulls. Stability is a stronger requirement than topological consistency, e.g., the topological algorithms ([SI92]) have not been proved stable.

ROUNDED GEOMETRY

Sugihara [Sug89] shows that the above problem of “rounding a line” can be reduced to the classical problem of *simultaneous approximation by rationals*: given real numbers a_1, \dots, a_n , find integers p_1, \dots, p_n and q such that $\max_{1 \leq i \leq n} |a_i q - p_i|$ is minimized. There are no efficient algorithms to solve this exactly, although lattice reduction techniques yield good approximations. The above approach of Greene and Yao can also be viewed as a geometric rounding problem. The “rounded lines” in the Greene-Yao sense is a polyline with unbounded combinatorial complexity; but rounded lines in the Sugihara sense still have constant complexity. Milenkovic and Nackman [MN90] show that rounding a collection of disjoint simple polygons while preserving their combinatorial structure is NP-complete. In Section 41.5, rounded geometry is seen in a different light.

ARITHMETICAL APPROACHES

Certain approaches might be described as mainly based on arithmetic considerations (as opposed to geometric considerations). Ottmann, Thiemt, and Ullrich [OTU87] show that the use of an accurate scalar product operator leads to improved

robustness in segment intersection algorithms; that is, the onset of qualitative errors is delayed. A case study of Dobkin and Silver [DS88] shows that permutation of operations combined with random rounding (up or down) can give accurate predictions of the total round-off error. By coupling this with a multiprecision arithmetic package that is invoked when the loss in significance is too severe, they are able to improve the robustness of their code. There is a large literature on computation under the interval arithmetic model (e.g., [Ull90]). It is related to what we call interval geometry above. There are also systems providing programming language support for interval analysis.

41.4 EXACT APPROACH

As the name suggests, this approach proposes to compute without any error. The initial interpretation is that every numerical quantity is computed exactly. While this has an natural meaning when all numerical quantities are rational, it is not obvious what this means for values such as $\sqrt{2}$ which cannot be exactly represented “explicitly.” Informally, a number representation is explicit if it facilitates efficient comparison operations. In practice, this amounts to representing numbers by one or more integers in some positional notation (this covers the usual representation of rational numbers as well as floating point numbers). Although we could achieve numerical exactness in some modified sense, this turns out to be unnecessary. The solution to the nonrobustness only requires a weaker notion of exactness: it is enough to ensure “geometric exactness.” In the “Geometric = Numeric + Combinatorial” formulation, the exactness is not to be found in the numeric part, but in the combinatorial part, as this encodes the geometric relations. Hence this approach is called ***Exact Geometric Computation*** (EGC), and it entails the following:

Input is exact. We cannot speak of exact geometry unless this is true. This assumption can be an issue if the input is inherently approximate. Sometimes we can simply treat the approximate inputs as “nominally” exact, as in the case of an input set of points without any constraints. Otherwise, there are two options: (1) “clean up” the inexact input, by transforming it to data that is exact; or (2) formulate a related problem in which the inexact input can be treated as exact (e.g., inexact input points can be viewed as the *exact* centers of small balls). So the convex hull of a set of points becomes the convex hull of a set of balls. The cleaning up process in (1) may be nontrivial as it may require perturbing the data to achieve some consistency property and lies outside our present scope. The transformation (2) typically introduces a computationally harder problem. Not much research is currently available for such transformed problems. In any case, (1) and (2) still end up with exact inputs for a well-defined computational problem.

Numerical quantities may be implicitly represented. This is necessary if we want to represent irrational values exactly. In practice, we will still need explicit numbers for various purposes (e.g., comparison, output, display, etc.). So a corollary is that numerical approximations will be important, a remark that was not obvious in the early days of EGC.

All branching decisions in a computation are errorless. At the heart of EGC is the idea that all “critical” phenomena in geometric computations are determined by the particular sequence branches taken in a *computation tree*. The key observation is that the sequence of branching decisions completely decides the combinatorial nature of the output. Hence if we make only errorless branches, the combinatorial part of a geometric structure D (see [Section 41.2](#)) will be correctly computed. To ensure this, we only need to evaluate *test values* to one bit of relative precision, i.e., enough to determine the sign correctly.

For problems (such as convex hulls) requiring only rational numbers, exact computation is possible. In other applications rational arithmetic is not enough. The most general setting in which exact computation is known to be possible is the framework of *algebraic problems* [Yap97].

GLOSSARY

Computation tree: A geometric algorithm in the algebraic framework can be viewed as an infinite sequence T_1, T_2, T_3, \dots of computation trees. Each T_n is restricted to inputs of size n , and is a finite tree with two kinds of nodes: (a) nonbranching nodes, (b) branching nodes. Assume the input to T_n is a sequence of n real parameters x_1, \dots, x_n . A nonbranching node at depth i computes a value v_i , say $v_i \leftarrow f_i(v_1, \dots, v_{i-1}, x_1, \dots, x_n)$. A branching node tests a previous computed value v_i and makes a 3-way branch depending on the sign of v_i . In case v_i is a complex value, we simply that the sign of the real part of v_i . Call any v_i that is used solely in a branching node a **test value**. The branch corresponding to a zero test value is the **degenerate branch**.

Exact Geometric Computation (EGC): Preferred name for the general approach of “exact computation,” as it accurately identifies the goal of determining geometric relations exactly. The exactness of the computed numbers is either unnecessary, or should be avoided if possible.

Composite Precision Bound: This is specified by a pair $[r, a]$ where $r, a \in \mathbb{R} \cup \{\infty\}$. For any $z \in \mathbb{C}$, let $z[r, a]$ denote the set of all $\tilde{z} \in \mathbb{C}$ such that $|z - \tilde{z}| \leq \max\{2^{-a}, |z|2^{-r}\}$. When $r = \infty$, then $z[\infty, a]$ comprises all the numbers \tilde{z} that approximates z with an absolute error of 2^{-a} ; we say this approximation \tilde{z} has **a absolute bits**. Similarly, $z[r, \infty]$ comprises all numbers \tilde{z} that approximates z with a relative error of 2^{-r} ; we say this approximation \tilde{z} has **r relative bits**.

Constant Expressions: Let Ω be a set of complex algebraic operators; each operator $\omega \in \Omega$ is a partial function $\omega : \mathbb{C}^{a(\omega)} \rightarrow \mathbb{C}$ where $a(\omega) \in \mathbb{N}$ is the arity of ω . If $a(\omega) = 0$, then ω is identified with a complex number. Let $\mathcal{E}(\Omega)$ be the set of expressions over Ω where an expression E is a rooted DAG (directed acyclic graph) and each node with outdegree $n \in \mathbb{N}$ is labeled with an operator of Ω of arity n . There is a natural **evaluation function** $\text{val} : \mathcal{E}(\Omega) \rightarrow \mathbb{R}$. If Ω has partial functions, then $\text{val}()$ is also partial. If $\text{val}(E)$ is undefined, we write $\text{val}(E) = \uparrow$ and say E is **invalid**. When $\Omega = \Omega_2 = \{+, -, \times, \div, \sqrt{}\} \cup \mathbb{Z}$ we get the important class of **constructible expressions**, so called because their values are precisely the constructible reals.

Constant Zero Problem, ZERO(Ω): Given $E \in \mathcal{E}(\Omega)$, decide if $\text{val}(E) = \uparrow$; if not, decide if $\text{val}(E) = 0$.

Guaranteed Precision Evaluation Problem, GVAL(Ω): Given $E \in \mathcal{E}(\Omega)$ and $a, r \in \mathbb{Z} \cup \{\infty\}$, $(a, r) \neq (\infty, \infty)$, compute some approximate value in $\text{val}(E)[r, a]$,

Schanuel's Conjecture: If $z_1, \dots, z_n \in \mathbb{C}$ are linearly independent over \mathbb{Q} , then the set $\{z_1, \dots, z_n, e^{z_1}, \dots, e^{z_n}\}$ contains a subset $B = \{b_1, \dots, b_n\}$ that is algebraically independent, i.e., there is no polynomial $P(X_1, \dots, X_n) \in \mathbb{Q}[X_1, \dots, X_n]$ such that $P(b_1, \dots, b_n) = 0$. This conjecture generalizes several deep results in transcendental number theory, and implies many other conjectures.

NAIVE APPROACH

For lack of a better term, we call the approach to exact computation in which every numerical quantity is computed exactly (explicitly if possible) the *naive approach*. Thus an exact algorithm that relies solely on the use of a big number package is probably naive. This approach, even for rational problems, faces the “bugbear of exact computation,” namely, high numerical precision. Using an off-the-shelf big number package does not appear to be a practical option [FvW93a, KLN91, Yu92]. There is evidence (surveyed in [YD95]) that just improving current big number packages alone is unlikely to gain a factor of more than 10.

BIG EXPRESSION PACKAGES

The most common examples of expressions are determinants and the distance $\sqrt{\sum_{i=1}^n (p_i - q_i)^2}$ between two points p, q . A big expression package allows a user to construct and evaluate expressions with big number values. They represent the next logical step after big number packages, and are motivated by the observation that the numerical part of a geometric computation is invariably reduced to repeated evaluations of a few variable¹ expressions (each time with different constants substituted for the variables). When these expressions are test values, then it is sufficient to compute them to one bit of relative precision. Some implementation efforts are shown in Table 41.4.1.

TABLE 41.4.1 Expression packages.

SYSTEM	DESCRIPTION	REFERENCES
LN	Little Numbers	[FvW96]
LEA	Lazy ExAct Numbers	[BJMM93]
Real/Expr	Precision-driven exact expressions	[YD95]
LEDA Real	Exact numbers of Library of Efficient Data structures and Algorithms	[BFMS99, BKM ⁺ 95]
Core Library	Package with Numerical Accuracy API and C++ interface	[KLPY99]

¹These expressions involve variables, unlike the constant expressions in $\mathcal{E}(\Omega)$.

One of LN’s goals is to remove all overhead associated with function calls or dynamic allocation of space for numbers with unknown sizes. It incorporates an effective floating-point filter based on static error analysis. The experience in [CM93] suggests that LN’s approach is too aggressive as it leads to code bloat. The LEA system philosophy is to delay evaluating an expression until forced to, and to maintain intervals of uncertainty for values. Upon complete evaluation, the expression is discarded. It uses root bounds to achieve exactness and floating point filters for speed.

The **Real/Expr Package** is the first system to achieve guaranteed precision for a general class of nonrational expressions. Its introduces the “precision-driven mechanism” whereby a user-specified precision at the root of the expression is transformed and downward-propagated toward the leaves, while approximate values generated at the leaves are evaluated and error bounds upward-propagated up to the root. This upward-downward process may need to be iterated. LEDA **Real** is a number type with a similar mechanism. It is part of a much more ambitious system of data structures for combinatorial and geometric computing (see [Chapter 65](#)). The semantics of **Real/Expr** of expression assignment is akin to constraint propagation in the constraint programming paradigm. The **Core Library** (**CORE**) is derived from **Real/Expr** with the goal of making the system as easy to use as possible. The two pillars of this transformation are the adoption of conventional assignment semantics and the introduction of a simple **Numerical Accuracy API** [Yap98].

The CGAL Library (Chapter 65) is a major library of geometric algorithms which are designed according to the EGC principles. While it has some native number types supporting rational expressions, the current distribution relies on LEDA **Real** or **CORE** for more general algebraic expressions. Shewchuk [She96] implements an arithmetic package that uses adaptive-precision floating-point representations. While not a big expression package, it has been used to implement polynomial predicates and shown to be extremely efficient.

THEORY

The class of algebraic computational problems encompasses most problems in contemporary computational geometry. Such problems can be solved exactly in singly-exponential space [Yap97]. This general result is based on recent progress in the decision problem for Tarski’s language, on the associated cell decomposition problems, as well as cell adjacency computation ([Chapter 33](#)). However, general EGC libraries such as **Core Library** and LEDA **Real** depend directly on the algorithms for the guaranteed precision evaluation problem $GVAL(\Omega)$ (see [Glossary](#)), where Ω is the set of operators in the computation model. The possibility of such algorithms can be reduced to the recursiveness of a constellation of problems that might be called the **Fundamental Problems of EGC**. The first is the Constant Zero Problem $ZERO(\Omega)$. But there are two closely related problems. In the **Constant Validity Problem** $VALID(\Omega)$, we are to decide if a given $E \in \mathcal{E}(\Omega)$ is valid, i.e., $\text{val}(E) \neq \uparrow$. The **Constant Sign Problem** $SIGN(\Omega)$ is to compute $\text{sign}(E)$ for any given $E \in \mathcal{E}(\Omega)$, where $\text{sign}(E) \in \{\uparrow, -1, 0, +1\}$. In case $\text{val}(E)$ is complex, define $\text{sign}(E)$ to be the sign of the real part of $\text{val}(E)$.

There is a natural hierarchy of the expression classes, each corresponding to a class of complex numbers as shown in 41.4.2. In Ω_3 , $P(X)$ is any polynomial with integer coefficients and I is some means of identifying a unique root of $P(X)$: I may be an complex interval bounding a unique root of $P(X)$, or an integer i

TABLE 41.4.2 Expression hierarchy.

OPERATORS	NUMBER CLASS	EXTENSIONS
$\Omega_0 = \{+, -, \times\} \cup \mathbb{Z}$	Integers	
$\Omega_1 = \Omega_0 \cup \{\div\}$	Rational Numbers	$\Omega_1^+ = \Omega_1 \cup \mathbb{Q}$
$\Omega_2 = \Omega_1 \cup \{\sqrt{\cdot}\}$	Constructible Numbers	$\Omega_2^+ = \Omega_2 \cup \{\sqrt[k]{\cdot} : k \geq 3\}$
$\Omega_3 = \Omega_2 \cup \{\text{RootOf}(P(X), I)\}$	Algebraic Numbers	Use of $\diamond(E_1, \dots, E_d, i)$, [BFM ⁺ 01]
$\Omega_4 = \Omega_3 \cup \{\exp(\cdot), \ln(\cdot)\}$	Elementary Numbers (cf. [Cho99])	

to indicate the i th largest real root of $P(X)$. The operator $\text{RootOf}(P, I)$ can be generalized to allow allowing expressions as coefficients of $P(X)$ as in Burnikel et al. [BFM⁺01], or by introducing systems of polynomial equations as in Richardson [Ric97]. Although Ω_4 can be treated as a set of real operators, it is more natural to treat Ω_4 (and sometimes Ω_3) as complex operators. Thus the elementary functions $\sin x, \cos x, \arctan x$, etc., are available as expressions in Ω_4 .

It is clear $\text{ZERO}(\Omega)$ and $\text{VALID}(\Omega)$ is reducible to $\text{SIGN}(\Omega)$. For Ω_4 , all three problems are recursively equivalent. The fundamental problems related to Ω_i are decidable for $i \leq 3$. It is a major open question whether the fundamental problems for Ω_4 are decidable. These questions have been studied by Richardson and others [Ric97, Cho99, MW96]. The most general positive result is that $\text{SIGN}(\Omega_3)$ is decidable. An intriguing conditional result is that $\text{ZERO}(\Omega_4)$ is decidable if Schanuel's conjecture is true; this may be deduced from Richardson's work [Ric97].

CONSTRUCTIVE ROOT BOUNDS

In practice, algorithms for the guaranteed precision problem $\text{GVAL}(\Omega_3)$ can exploit the fact that algebraic numbers have computable root bounds. A **root bound** for Ω is a total function $\beta : \mathcal{E}(\Omega) \rightarrow \mathbb{R}_{\geq 0}$ such that for all $E \in \mathcal{E}(\Omega)$, if E is valid and $\text{val}(E) \neq 0$ then $|\text{val}(E)| \geq \beta(E)$. More precisely, β is called an **exclusion** root bound; it is an **inclusion root bound** when the inequality becomes " $|\text{val}(E)| \leq \beta(E)$." We use the (exclusion) root bound β to solve $\text{ZERO}(\Omega)$ as follows: to test if an expression E evaluates to zero, we compute an approximation α to $\text{val}(E)$ such that $|\alpha - \text{val}(E)| < \beta(E)/2$. While computing α , we can recursively verify the validity of E . If E is valid, we compare α with $\beta/2$. It is easy to conclude that $\text{val}(E) = 0$ if $|\alpha| \leq \beta/2$. Otherwise $|\alpha| > \beta/2$, and the sign of $\text{val}(E)$ is that of α . An important remark is that the root bound β determines the worst-case complexity. This is unavoidable if $\text{val}(E) = 0$. But if $\text{val}(E) \neq 0$, the worst case may be avoided by iteratively computing α_i with increasing absolute precision ε_i . If for any $i \geq 1$, $|\alpha_i| > \varepsilon_i$, we stop and conclude $\text{sign}(\text{val}(E)) = \text{sign}(\alpha_i) \neq 0$.

There is an extensive classical mathematical literature on root bounds, but they are usually not suitable for computation. Recently, new root bounds have been introduced that explicitly depend on the structure of expressions $E \in \mathcal{E}(E)$. In [LY01], such bounds are called **constructive** in the following sense: (i) There are easy-to-compute recursive rules for maintaining a set of numerical parameters $u_1(E), \dots, u_m(E)$ based on the structure of E , and (ii) $\beta(E)$ is given by an explicit formula in terms of these parameters. The first constructive bounds in EGC were

the degree-length and degree-height bounds of Yap and Dubé [YD95, Yap00] in their implementation of **Real/Expr**. The (Mahler) Measure Bound was introduced even earlier by Mignotte [Mig82, BFMS00] for the problem of “identifying algebraic numbers.” A major improvement was achieved with the introduction of the BFMS Bound [BFMS00]. Li-Yap [LY01] introduced another bound aimed at improving the BFMS Bound in the presence of division. Comparison of these bounds is not easy: but let us say a bound β **dominates** another bound β' if for every $E \in \mathcal{E}(\Omega_2)$, $\beta(E) \leq \beta'(E)$. Burnikel et al. [BFM⁺01] generalized the BFMS Bound to the BFMSS Bound. Yap noted that if we incorporate a symmetrizing trick for the $\sqrt{x/y}$ transformation, then BFMSS will dominate BFMS. Among current constructive root bounds, three are not dominated by other bounds: BFMSS, Measure, and Li-Yap Bounds. In general, BFMSS seems to be the best. Other root bounds include a multivariate root bound of Canny [Can88] (see extension in [Yap00, Chapter XI]) and an Eigenvalue Bound of Scheinerman [Sch00]. A recent factoring technique of Pion and Yap [PY03] can be used to improve the existing bounds (in particular, BFMSS). This technique can exploit the presence of k -ary input numbers, and is thus favorable for the majority of realistic inputs (which are binary or decimal).

FILTERS

An extremely effective technique for speeding up predicate evaluation is based on the filter concept. Since evaluating the predicate amounts to determining the sign of an expression E , we can first use machine arithmetic to quickly compute an approximate value α of E . For a small overhead, we can simultaneously determine an error bound ε where $|\text{val}(E) - \alpha| \leq \varepsilon$. If $|\alpha| > \varepsilon$, then the sign of α is the correct one and we are done. Otherwise, we evaluate the sign of E again, this time using a sure-fire if slow evaluation method. The algorithm used in the first evaluation is called a (floating-point) **filter**. The expected cost of the two-stage evaluation is small if the filter is efficient with a high probability of success. This idea was first used by Fortune and van Wyk [FvW96]. Floating-point filters can be classified along the static-to-dynamic dimension: **static filters** compute the bound ε solely from information that are known at compile time while **dynamic filters** depend on information available at run time. There is an **efficiency-efficacy tradeoff**: static filters (e.g., FvW Filter [FvW96]) are more efficient, but dynamic filters (e.g., BFS Filter [BFS98]) are more accurate (efficacious). Interval arithmetic has been shown to be an effective way to implement dynamic filters [BBP01]. Automatic tools for generating filter code are treated in [FvW93b, Fun97]. Filters can be elaborated in several ways. First, we can use a cascade of filters [BFS98]. The “steps” of an algorithm which are being filtered can be defined at different levels of granularity. One extreme is to consider an entire algorithm as one step [MNS⁺96, KW98]. A general formulation “structural filtering” is proposed in [FMN99]. Probabilistic analysis [DP99] shows the efficacy of arithmetic filters. The filtering of determinants is treated in several papers [Cla92, BBP01, PY01, BY00].

Filtering is related to program checking [BK95, BLR93]. View a computational problem P as an input-output relation, $P \subseteq I \times O$ where I, O is the input and output spaces, respectively. Let be A a (standard) **algorithm** for P which, viewed as a total function $A : I \rightarrow O \cup \{\text{NaN}\}$, has the property that for all $i \in I$, $(i, A(i)) \in P$ iff there is some $o \in O$ such that $(i, o) \in P$. Let $H : I \rightarrow O \cup \{\text{NaN}\}$ be another algorithm with no restrictions; call H a **heuristic algorithm** for P .

Let $F : I \times O \rightarrow \{\text{true}, \text{false}\}$. Then F is **checker** for P if F computes the characteristic function for P , $F(i, o) = \text{true}$ iff $(i, o) \in P$. Note that F is a checker for the problem P , and not for any purported program for P . Hence, unlike program checking, we do not require any special properties of P such as self-reducibility. We call F a **filter** for P if $F(i, o) = \text{true}$ implies $(i, o) \in P$. So filters are less restricted than checkers. A **filtered program** for P is therefore a triple (H, F, A) where H is heuristic algorithm, A a standard algorithm and F a filter. To run this program on input i , we first compute $H(i)$ and check if $F(i, H(i))$ is true. If so, we output $H(i)$; otherwise compute and output $A(i)$. Filtered programs can be extremely effective when H, F are both efficient and efficacious. Usually H is easy—it is just a machine arithmetic implementation of an exact algorithm. The filter F can be more subtle, but it is still more readily constructed than any checker. The problem P_{sdet} of computing the sign of determinants illustrates this: the only checker we know here is trivial, amounting to computing the determinant itself. On the other hand, effective filters for P_{sdet} are known [BBP01, PY01].

PRECISION COMPLEXITY

An important goal of EGC is to control the cost of high-precision computation. We describe two approaches based on modifying the algorithmic specification.

In predicate evaluation, there is an in-built precision of 1-relative bit (this precision guarantees the correct sign in the predicate evaluation). But in construction steps, any precision guarantees must be explicitly requested by the user. For optimization problems, a standard method to specify precision is to incorporate an extra input parameter $\epsilon > 0$. Assume the problem is to produce an output x to minimizes the function $\mu(x)$. An **ϵ -approximation algorithm** will output a solution x such that $\mu(x) \leq (1 + \epsilon)\mu(x^*)$ for some optimum x^* . An example is the **Euclidean Shortest-path Problem in 3-space** (3ESP). Since this problem is NP-hard (Section 27.5), we seek an ϵ -approximation algorithm. A simple way to implement an ϵ -approximation algorithm is to directly implement any *exact* algorithm in which the underlying arithmetic has guaranteed precision evaluation (using, e.g., Core Library). However, the bit complexity of such an algorithm may not be obvious. The more conventional approach is to explicitly build the necessary approximation scheme directly into the algorithm. One such scheme was given by Papadimitriou [Pap85] which is polynomial time in n and $1/\epsilon$. Choi et al. [CSY97] give an improved scheme, and perform a rare bit-complexity analysis.

Another way to control precision is to consider output complexity. In geometric problems, the input and output **sizes** are measured in two independent ways: combinatorial size and bit sizes. Let the input combinatorial and input bit sizes be n and L , respectively. By an L -bit input, we mean each of the numerical parameters in the description of the geometric object (see Section 41.2) is an L -bit number. Now an extremely fruitful concept in algorithmic design is this: an algorithm is said to be **output-sensitive** if the complexity of the algorithm can be made a function of the output size as well as of the input size parameters. In the usual view of output-sensitivity, only the output combinatorial size is exploited. Choi et al. [SCY00] introduced the concept of **precision-sensitivity** to remedy this gap. They presented the first precision-sensitive algorithm for 3ESP, and gave some experimental results. Using the framework of **pseudo-approximation algorithms**, Asano et al. [AKY04] gave new precision-sensitive algorithms for 3ESP, as well as for an optimal d_1 -motion for a rod.

GEOMETRIC ROUNDING

We saw rounded geometry as one of the fixed-precision approaches (Section 41.3) to robustness. But geometric rounding is also important in EGC, with a difference. The EGC problem is to “round” a geometric structure (Section 41.2) D to a geometric structure D' with lower precision. In fixed-precision computation, one is typically asked to construct D' from some input S that *implicitly* defines D . In EGC, D is explicitly given (e.g., D may be computed from S by an EGC algorithm). The EGC view should be more tractable since we have separated the two tasks: (a) computing D and (b) rounding D . We are only concerned with (b), the ***pure rounding problem***. For instance, if S is a set of lines that are specified by linear equations with L -bit coefficients, then the arrangement $D(S)$ of S would have vertices with $2L + O(1)$ -bit coordinates. We would like to round the arrangement, say, back to L bits. Such a situation, where the output bit precision is larger than the input bit precision, is typical. If we pipeline several of these computations in a sequence, the final result could have a very high bit precision unless we perform rounding.

If D rounds to D' , we could call D' a ***simplification*** of D . This viewpoint connects to a larger literature on simplification of geometry (e.g., simplifying geometric models in computer graphics and visualization ([Chapter 54](#))). Two distinct objectives goals in simplification are ***combinatorial*** versus ***precision simplification***. For example, a problem that has been studied in a variety of contexts (e.g., Douglas-Peucker algorithm in computational cartography) is that of simplifying a polygonal line P . We can use ***decimation*** to reduce the combinatorial complexity (i.e., number of vertices $\#(P)$), for example, by omitting every other vertex in P . Or we can use ***clustering*** to reduce the bit-complexity of P to L -bits, e.g., we collapse all vertices that lie within the same grid cell, assuming grid points are L -bit numbers. Let $d(P, P')$ be the Hausdorff distance between P and another polyline P' ; other similar measures of distance may be used. In any simplification P' of P , we want to keep $d(P, P')$ small. In [BCD⁺02], two optimization problems are studied: in the ***Min-# Problem***, given P and ε , find P' to minimize $\#(P)$, subject to $d(P, P') \leq \varepsilon$. In the ***Min- ε Problem***, the roles of $\#(P)$ and $d(P, P')$ are reversed. For EGC applications, optimality can often be relaxed to simple feasibility. Path simplification can be generalized to the simplification of any cell complexes.

BEYOND ALGEBRAIC

Non-algebraic computation over Ω_4 is important in practice. This includes the use of elementary functions such as $\exp x, \ln x, \sin x$, etc, which are found in standard libraries (`math.h` in C/C++). Elementary functions can be implemented via their representation as ***hypergeometric functions***, an approach taken by Du et al. [DEMY02]. They described solutions for fundamental issues such as automatic error analysis, hypergeometric parameter processing and argument reduction. If f is a hypergeometric function and x is an explicit number, one can compute $f(x)$ to any desired absolute accuracy. But in the absence of root bounds for Ω_4 , we cannot solve the guaranteed precision problem $\text{GVAL}(\Omega_4)$. One systematic way to get around this is to invoke the uniformity conjecture [Ric00]: this conjecture provides us with a bound. If this bound ever led to an error, we would have produced a

counterexample to the uniformity conjecture.

There are situations where we can either avoid the use of transcendental functions, or their apparent need turns out to be non-essential (e.g., in motion planning). For instance, rigid transformations are important in solid modeling, but they involve trigonometric functions. We can get arbitrarily good approximations by using *rational rigid transformations*. Solutions in 2 and 3 dimensions are given by Canny et al. [CDR92] and Milenkovic and Milenkovic [MM93], respectively.

APPLICATIONS

We now consider issues in implementing specific algorithms under the EGC paradigm. The rapid growth in the number of such algorithms means the following list is quite partial. We attempt to illustrate the range of activities in several groups: **(i)** The early EGC algorithms produced were those that are easily reduced to integer arithmetic and polynomial predicates, such convex hulls or Delaunay triangulations. The goal was to demonstrate that such algorithms are implementable and relatively efficient (e.g., [FvW96]). To treat irrational predicates, the careful analysis of root bounds were needed to ensure efficiency. Thus, Burnikel, Mehlhorn, and Schirra [BMS94, Bur96] gave sharp bounds in the case of Voronoi diagrams for line segments. Similarly, Dubé and Yap [DY93] analyzed the root bounds in Fortune's sweepline algorithm, and first identified the usefulness of floating point approximations in EGC. Another approach is to introduce algorithms that use new predicates with low algebraic degrees. This line of work was initiated by Liotta, Preparata, and Tamassia [LPT97, BS00]. **(ii)** Polyhedral modeling is a natural domain for EGC techniques. Two efforts are [CM93, For97]. The most general viewpoint here uses Nef polyhedra [See01] in which open, closed or half-open polyhedral sets are represented. This is a radical departure from the traditional solid modeling based on *regularized sets* and the associated *regularized operators*. The regularization of a set $S \subseteq \mathbb{R}^d$ is obtained as the closure of the interior of S ; regularized sets do not allow lower dimensional features, e.g., a line sticking out of a solid is not permitted. Treatment of Nef polyhedra was previously impossible outside the EGC framework. **(iii)** An interesting domain is optimization problems such as linear and quadratic programming [Gae99, GS00] and the smallest enclosing cylinder problem [SSTY00]. In linear programming, there is a tradition of using benchmark problems for evaluating algorithms and their implementations. But what is lacking in the benchmarks is *reference solutions* with guaranteed accuracy to (say) 16 digits. One application of EGC algorithms is to produce such solutions. **(iv)** An area of major challenge is computation of algebraic curves and surfaces. Krishnan et al. [KFC⁺01] implemented a library of algebraic primitives to support the manipulation of algebraic curves. Algorithms for low degree curves and surfaces are beginning to be addressed, e.g., [BEH⁺02, GHS01, Wei02]. **(v)** The development of general geometric libraries such as CGAL [HHK⁺01] or LEDA [MN95] exposes a range of issues peculiar to EGC. For instance, in EGC we want a framework where various number kernels and filters can be used for a single algorithm.

41.5 TREATMENT OF DEGENERACIES

Suppose the input to an algorithm is a set of planar points. Depending on the context, any of the following scenarios might be considered “degenerate”: two covetical points, three collinear points, four cocircular points. Intuitively, these are degenerate because arbitrarily small perturbations can result in qualitatively different geometric structures. Degeneracy is basically a discontinuity [Yap90b, Sei98]. Sedgewick [Sed83] calls degeneracies the “bugbear of geometric algorithms.” Degeneracy is a major cause of nonrobustness for two reasons. First, it presents severe difficulties for approximate arithmetic. Second, even under the EGC paradigm, implementors are faced with a large number of special degenerate cases that must be treated (this number grows exponentially in the dimension of the underlying space). Thus there is a need to develop general techniques for handling degeneracies.

GLOSSARY

Inherent and induced degeneracy: This is illustrated by the planar convex hull problem: an input set S with three collinear points p, q, r is inherently degenerate if it lies entirely in one halfplane determined by the line through p, q, r . If p, q, r are collinear but S does not lie on one side of the line through p, q, r , then we may have an induced degeneracy for a divide-and-conquer algorithm. This happens when the algorithm solves a subproblem $S' \subseteq S$ containing p, q, r with all the remaining points on one side. Induced degeneracy is algorithm-dependent. In this chapter, we simply say “degeneracy” for induced degeneracy. More precisely, an input is **degenerate** if it leads to a path containing a vanishing test value in the computation tree [Yap90b]. A nondegenerate input is also said to be **generic**.

Generic algorithm: One that is only guaranteed to be correct on generic inputs.

General algorithm: One that works correctly for all (legal) inputs. Note that “general” and “generic” are often used synonymously in other literature (e.g., “generic inputs” often means inputs in general position).

THE BASIC ISSUES

1. One basic goal of this field is to provide a *systematic transformation* of a generic algorithm A into a general algorithm A' . Since generic algorithms are widespread in the literature, the availability of general tools for this $A \mapsto A'$ transformation is useful for implementing robust algorithms.
2. Underlying any transformations $A \mapsto A'$ is some kind of perturbation of the inputs. This raises the issue of *controlled perturbations*. For example, if A is an algorithm for intersecting two convex polytopes, then we would like the perturbation to expand the input polytopes so that the incidence of a vertex in the relative interior of a face will be detected by A' .
3. There is a *postprocessing issue*: although A' is “correct” in some technical

sense, it need not necessarily produce the same outputs as an ideal algorithm A^* . For example, suppose A computes the Voronoi diagram of a set of points in the plane. Four cocircular points are a degeneracy and are not treated by A . The transformed A' can handle four cocircular points but it may output two Voronoi vertices that have identical coordinates and are connected by a Voronoi edge of length 0. This may arise if we use infinitesimal perturbations. The postprocessing problem amounts to cleaning up the output of A' (removing the length-0 edges in this example) so that it conforms to the ideal output of A^* .

CONVERTING GENERIC TO GENERAL ALGORITHMS

We have two general methods for converting a generic algorithm to a general one:

Blackbox sign evaluation schemes. We postulate a *sign blackbox* that takes as input a function $f(\mathbf{x}) = f(x_1, \dots, x_n)$ and parameters $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{R}^n$, and outputs a nonzero sign (either + or -). In case $f(\mathbf{a}) \neq 0$, this sign is guaranteed to be the sign of $f(\mathbf{a})$, but the interesting fact is that we get a nonzero sign even if $f(\mathbf{a}) = 0$. We can formulate a consistency property for the blackbox, both in an algebraic setting [Yap90b] or in a geometric setting [Yap90a]. The transformation $A \mapsto A'$ amounts to replacing all evaluations of test values by calls to this blackbox. In [Yap90b], a family of *admissible schemes* for blackboxes is given in case the functions $f(\mathbf{x})$ are polynomials.

Perturbation toward a nondegenerate instance. A fundamentally different approach is provided by Seidel [Sei98], based on the following idea. For any problem, if we know one nondegenerate input \mathbf{a}^* for the problem, then every other input \mathbf{a} can be made nondegenerate by perturbing it in the direction of \mathbf{a}^* . We can take the perturbed input to be $\mathbf{a} + \epsilon \mathbf{a}^*$ for some infinitesimal ϵ . For example, for the convex hull of points in \mathbb{R}^n , we can choose \mathbf{a}^* to be distinct points on the moment curve (t, t^2, \dots, t^n) .

We compare these two approaches. We currently only have blackbox schemes for rational functions, while Seidel's method would apply even in nonalgebraic settings. Blackbox schemes are independent of particular problems, while the nondegenerate instances \mathbf{a}^* depend on the problem (and on the input size); no systematic method to choose \mathbf{a}^* is known.

The first work in this area is the SoS (“simulation of simplicity”) technique of Edelsbrunner and Mücke [EM90]. The method amounts to adding powers of an indeterminate ϵ to each input parameter. Such ϵ -methods were first used in linear programming in the 1950s. The SoS scheme (for determinants) turns out to be an admissible scheme [Yap90b]. Intuitively, sign blackbox invocations should be almost as fast as the actual evaluations with high probability [Yap90b]. But the worst-case exponential behavior led Emiris and Canny to propose more efficient numerical approaches [EC95]. To each input parameter a_i in \mathbf{a} , they add a perturbation $b_i\epsilon$ (where $b_i \in \mathbb{Z}$ and ϵ is again an infinitesimal): these are called *linear perturbations*. In case the test values are determinants, they show that a simple choice of the b_i 's will ensure nondegeneracy and efficient computation. For general rational function tests, a lemma of Schwartz shows that a random choice of the b_i 's

is likely to yield nondegeneracy. Emiris, Canny, and Seidel [ECS97, Sei98] give a general result on the validity of linear perturbations, and apply it to common test polynomials.

APPLICATIONS AND PRACTICE

Michelucci [Mic95] describes implementations of blackbox schemes, based on the concept of “ ϵ -arithmetic.” One advantage of his approach is the possibility of controlling the perturbations. Experiences with the use of perturbation in the beneath-beyond convex hull algorithm in arbitrary dimensions are reported in [ECS97]. Neuhauser [Neu97] improved and implemented the rational blackbox scheme of Yap. He also considered controlled perturbation techniques. Comes and Ziegelmann [CZ99] implemented the linear perturbation ideas of Seidel in CGAL.

In solid modeling systems, it is very useful to systematically avoid degenerate cases (numerous in this setting). Fortune [For97] uses symbolic perturbation to allow an “exact manifold representation” of nonregularized polyhedral solids (see [Section 56.1](#)). The idea is that a dangling rectangular face (for instance) can be perturbed to look like a very flat rectangular solid, which has a manifold representation. Here, controlling the perturbation is clearly necessary.

Hertling and Weihrauch [HW94] define “levels of degeneracy” and use this to obtain lower bounds on the size of decision computation trees.

In contrast to our general goal of eliminating *explicit* handling of degeneracies, there are a few papers on “perturbation” that proposes to directly handle degeneracies. Burnikel, Mehlhorn, and Schirra [BMS95] describe the implementation of a line segment intersection algorithm and semidynamic convex hull maintenance in arbitrary dimensions. Based on this experience, they question the usefulness of perturbation methods using three observations: (i) perturbations may increase the running time of an algorithm by an arbitrary amount; (ii) the postprocessing problem can be significant; and (iii) it is not hard to handle degeneracies directly. But the probability of (i) occurring in a drastic way (e.g., for a degenerate input of n identical points) is so negligible that it may not deter most users when they have the option of writing a generic algorithm, especially when the general algorithm is very complex or not readily available. Other experiences suggest that property (iii) is the exception rather than the rule. In any case, users must weigh these considerations (cf. [Sch94]).

A weaker form of the [BMS95] approach is illustrated by work of Halperin and co-workers [HS98, Raa99]. Again, the algorithm must explicitly detect the presence of degeneracies, but now we explicitly perturb the input to remove all degeneracies. Their problem may be framed as follows: given a sequence $S = (O_1, \dots, O_n)$ of geometric objects, let A_i ($i = 1, \dots, n$) be the arrangement formed by $S_i = (O_1, \dots, O_i)$. The goal is to compute $A_n = A(S_n)$. For any object O and $\varepsilon > 0$, consider a predicate $P_1(O, \varepsilon)$ with this **monotonicity property**: if $\varepsilon' > \varepsilon$ and $P_1(O, \varepsilon')$ is true then $P_1(O, \varepsilon)$ is true. Call P_1 an **approximate degeneracy predicate**. If $P_1(O, \varepsilon)$ is true, we say O is ε -**degenerate**. Also, $P_1(O, 0^+)$ reduces to standard notions of degeneracy. Such predicates may be defined by a Boolean combination of polynomial inequalities. For instance, let O be a curve and $P_1(O, \varepsilon)$ is true iff there is a δ -ball B centered at a point of O , $\delta \leq \varepsilon$, such that $B \cap O$ is not connected. Thus $P_1(O, 0^+)$ is the property that O is self-intersecting. In general, let P_k denote an approximate degenerate predicate on $k \geq 1$ distinct objects. If P_k and

P'_k are two such predicates, then so is $P_k \vee P'_k$ and $P_k \wedge P'_k$. For instance, $P_2(O_1, O_1, \varepsilon)$ might say that O_1, O_2 are ε -close. Fix a collection \mathcal{P} of approximate degeneracy predicates. We say that S is ε -**degenerate** if for some $P_k \in \mathcal{P}$, $P_k(O_1, \dots, O_k, \varepsilon)$ is true for some choice of k distinct objects $O_1, \dots, O_k \in S$. The following ε - δ **perturbation estimation problem** is basic: given $\varepsilon > 0$, find $\delta = \delta(\varepsilon, S, O) > 0$ such that if S is non ε -degenerate, and O is any object, with probability $> 1/2$, a random δ -perturbation O' of O will form a non ε -degenerate configuration with S . By general principles, we know that δ exists; but we would like good bounds on δ (say polynomial in $|S|$, etc). Using this, we can solve the **perturbed arrangement problem**: given S and $\varepsilon > 0$, compute an arrangement $A(S')$ where S' is not ε -degenerate and S' is a δ -perturbation of S . The cited papers above solve the perturbed arrangement problem in two situations, when the objects are spheres and polyhedral surfaces, respectively. The idea is to use a form of randomized incremental construction.

41.6 OPEN PROBLEMS

1. The main theoretical question in EGC is whether the Constant Zero Problem for Ω_4 is decidable. A related, possibly simpler, question is whether $\text{ZERO}(\Omega_3 \cup \{\sin(\cdot), \pi\})$ is decidable.
2. In constructive root bounds, it is unknown if there exists a root bound $\beta : \mathcal{E}(\Omega_2) \rightarrow \mathbb{R}_{\geq 0}$ where $-\lg(\beta(E)) = O(D(E))$ and $D(E)$ is the degree of E . In current bounds, we only know a quadratic bound, $-\lg(\beta(E)) = O(D(E)^2)$. The Uniformity Conjecture of Richardson [Ric00], if true, would be a very deep result with practical applications.
3. Give a optimal algorithm for the guaranteed precision evaluation problem $\text{GVAL}(\Omega)$ for, say, $\Omega = \Omega_2$. The solution includes a reasonable cost model.
4. In geometric rounding, we pose two problems: (a) Extend the Greene-Yao rounding problem to non-uniform grids (e.g., the grid points are L -bit floating point numbers). (b) Round simplicial complexes. The preferred notion of rounding here should not increase combinatorial complexity (unlike Greene-Yao), but rather allow features to collapse (triangles can degenerate to a vertex), but disallow inversion (triangles cannot flip its orientation).
5. Give good bounds for the ε - δ perturbation estimation problem.
6. Give a systematic treatment of inexact (dirty) data. Held [Hel01a, Hel01b] describes the engineering of reliable algorithms to handle such inputs.

41.7 SOURCES AND RELATED MATERIAL

SURVEYS

Forrest [For87] is an influential overview of the field of computational geometry. He deplores the gap between theory and practice and describes the open problem of robust intersection of line segments (expressing a belief that robust solutions do not exist). Other surveys of robustness issues in geometric computation are Schirra [Sch99], Yap and Dubé [YD95] and Fortune [For93]. Robust geometric modelers are surveyed in [PCH⁺95].

RELATED CHAPTERS

- [Chapter 24: Arrangements](#)
- [Chapter 27: Shortest paths and networks](#)
- [Chapter 33: Computational real algebraic geometry](#)
- [Chapter 56: Solid modeling](#)
- [Chapter 64: Computational geometry software](#)
- [Chapter 65: Two computational geometry libraries: LEDA and CGAL](#)

REFERENCES

- [AKY04] Te. Asano, D.G. Kirkpatrick, and C.K. Yap. Pseudo approximation algorithms, with applications to optimal motion planning. *Discrete Comput. Geom.*, 31:131–171, 2004.
- [BBP01] H. Brönnimann, C. Burnikel, and S. Pion. Interval arithmetic yields efficient dynamic filters for computational geometry. *Discrete Appl. Math.*, 109(1–2):25–47, 2001.
- [BCD⁺02] G. Barequet, D.Z. Chen, O. Daescu, M.T. Goodrich, and J. Snoeyink. Efficiently approximating polygonal paths in three and higher dimensions. *Algorithmica*, 33:150–167, 2002.
- [BEH⁺02] E. Berberich, A. Eigenwillig, M. Hemmer, S. Hert, K. Mehlhorn, and E. Schömer. A computational basis for conic arcs and boolean operations on conic polygons. *Proc. ESA 2002, Lecture Notes Comput. Sci.*, volume 2461, pages 174–186, Springer-Verlag, Berlin, 2002.
- [BFM⁺01] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. *Lecture Notes Comput. Sci.*, volume 2161, pages 254–265, Springer-Verlag, Berlin, 2001.
- [BFMS99] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. Exact geometric computation made easy. *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 341–450, 1999.
- [BFMS00] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica*, 27:87–99, 2000.

- [BFS98] C. Burnikel, S. Funke, and M. Seel. Exact geometric predicates using cascaded computation. *Proc. 14th Annu. Sympos. Comput. Geom.*, pages 175–183, 1998.
- [BJMM93] M.O. Benouamer, P. Jaillon, D. Michelucci, and J-M. Moreau. A lazy arithmetic library. *Proc. IEEE 11th Sympos. Computer Arithmetic*, pages 242–269, Windsor, Ontario, 1993.
- [BK95] M. Blum and S. Kannan. Designing programs that check their work. *J. Assoc. Comput. Mach.*, 42:269–291, 1995.
- [BKM⁺95] C. Burnikel, J. Könnemann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Exact geometric computation in LEDA. *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C18–C19, 1995.
- [BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing and self-correcting programs, with applications to numerical programs. *J. Comput. Syst. Sci.*, 47:549–595, 1993.
- [BMS94] C. Burnikel, K. Mehlhorn, and S. Schirra. How to compute the Voronoi diagram of line segments: Theoretical and experimental results. *Lecture Notes Comput. Sci.*, volume 855, Springer-Verlag, Berlin, pages 227–239, 1994.
- [BMS95] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1995.
- [BS00] J.-D. Boissonnat and J. Snoeyink. Efficient algorithms for line and curve segment intersection using restricted predicates. *Comput. Geom. Theory Appl.*, 16(1), 2000.
- [Bur96] C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections*. Ph.D thesis, Universität des Saarlandes, 1996.
- [BY00] H. Brönnimann and M. Yvinec. Efficient exact evaluation of signs of determinants. *Algorithmica*, 27:21–56, 2000.
- [Can88] J.F. Canny. *The complexity of robot motion planning*. Ph.D. thesis, MIT. *ACM Doctoral Dissertation Award Series*. The MIT Press, Cambridge, 1988.
- [CDR92] J.F. Canny, B.R. Donald, and E.K. Ressler. A rational rotation method for robust geometric algorithms. *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 251–160, Berlin, 1992.
- [Cho99] T.Y. Chow. What is a closed-form number? *Amer. Math. Monthly*, 106:440–448, 1999.
- [Cla92] K.L. Clarkson. Safe and effective determinant evaluation. *Proc. 33th Annu. IEEE Sympos. Found. Comput. Sci.*, 387–395, 1992.
- [CM93] J.D. Chang and V.J. Milenkovic. An experiment using LN for exact geometric computations. *Proc. 5th Canad. Conf. Comput. Geom.*, pages 67–72, Univ. Waterloo, 1993.
- [CSY97] J. Choi, J. Sellen, and C.K. Yap. Approximate Euclidean shortest path in 3-space. *Internat. J. Comput. Geom. Appl.*, 7:271–295, 1997.
- [CZ99] J. Comes and M. Ziegelmann. An easy to use implementation of linear perturbations within cgal. *Proc. 3rd Workshop Algorithm Eng. (WAE99)*, Berlin, 1999. *Lecture Notes Comput. Sci.*, volume 1668, Springer-Verlag, Berlin, 1999.
- [DEM^Y02] Z. Du, M. Eleftheriou, J. Moreira, and C.K. Yap. Hypergeometric functions in exact geometric computation. In V. Brattka, M. Schoeder, and K. Weihrauch, editors, *Proc. 5th Workshop Comput. Complexity Anal.*, pages 55–66, 2002. Malaga, 2002. *Electr. Notes Theoret. Comput. Sci.*, 66:1 (2002), <http://www.elsevier.nl/locate/entcs/volume66.html>.

- [DP99] O. Devillers and F.P. Preparata. Further results on arithmetic filters for geometric predicates. *Comput. Geom. Theory Appl.*, 13:141–148, 1999.
- [DS88] D.P. Dobkin and D. Silver. Recipes for Geometry & Numerical Analysis—Part I: An empirical study. *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, 93–105, 1988.
- [DY93] T. Dubé and C.K. Yap. A basis for implementing exact geometric algorithms (extended abstract), 1993. Paper from URL <http://cs.nyu.edu/cs/faculty/yap>.
- [EC95] I.Z. Emiris and J.F. Canny. A general approach to removing degeneracies. *SIAM J. Computing*, 24:650–664, 1995.
- [ECS97] I.Z. Emiris, J.F. Canny, and R. Seidel. Efficient perturbations for handling geometric degeneracies. *Algorithmica*, 19:219–242, 1997.
- [EM90] H. Edelsbrunner and E.P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 1990.
- [FM91] S.J. Fortune and V.J. Milenkovic. Numerical stability of algorithms for line arrangements. *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, 334–341, 1991.
- [FMN99] S. Funke, K. Mehlhorn, and S. Näher. Structural filtering: A paradigm for efficient and exact geometric programs. *Proc. 11th Canad. Conf. Comput. Geom.*, pages 39–42, 1999.
- [For87] A.R. Forrest. Computational geometry and software engineering: Towards a geometric computing environment. In D.F. Rogers and R.A. Earnshaw, editors, *Techniques for Comput. Graph.*, pages 23–37. Springer-Verlag, Berlin, 1987.
- [For89] S.J. Fortune. Stable maintenance of point-set triangulations in two dimensions. *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, 494–499, 1989.
- [For92] S.J. Fortune. Numerical stability of algorithms for 2-d Delaunay triangulations. *Proc. 8th Annu. ACM Sympos. Computational Geom.*, pages 83–92, 1992.
- [For93] S.J. Fortune. *Progress in Computational Geometry*, chapter 3, pages 81–127, R. Martin, editor. Information Geometers, Winchester, 1993.
- [For95a] S.J. Fortune. Numerical stability of algorithms for 2-d Delaunay triangulations. *Internat. J. Comput. Geom. Appl.*, 5:193–213, 1995.
- [For97] S.J. Fortune. Polyhedral modeling with multiprecision integer arithmetic. *Comput. Aided Design*, pages 123–133, 1997.
- [For98] S.J. Fortune. Vertex-rounding a three-dimensional polyhedral subdivision. *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 116–125, 1998.
- [Fun97] S. Funke. *Exact arithmetic using cascaded computation*. Master’s thesis, Max Planck Institute for Computer Science, Saarbrücken, Germany, 1997.
- [FvW93a] S.J. Fortune and C.J. van Wyk. Efficient exact arithmetic for computational geometry. *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1993.
- [FvW93b] S.J. Fortune and C.J. van Wyk. LN User Manual, 1993. AT&T Bell Laboratories.
- [FvW96] S.J. Fortune and C.J. van Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. *ACM Trans. Graph.*, 15:223–248, 1996.
- [Gae99] B. Gärtner. Exact arithmetic at low cost—a case study in linear programming. *Comput. Geom. Theory Appl.*, 13:121–139, 1999.
- [GGHT97] M.T. Goodrich, L.J. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 284–293, 1997.

- [GHS01] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 264–273, 2001.
- [GM95] L.J. Guibas and D. Marimont. Rounding arrangements dynamically. *Proc. 11th Annu. ACM Sympos. Computational Geom.*, pages 190–199, 1995.
- [GS00] B. Gärtner and S. Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, 110–118, 2000.
- [GSS89] L.J. Guibas, D. Salesin, and J. Stolfi. Epsilon geometry: building robust algorithms from imprecise computations. *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, 208–217, 1989.
- [GY86] D.H. Greene and F.F. Yao. Finite-resolution computational geometry. *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci.*, 143–152, 1986.
- [Hel01a] M. Held. FIST: Fast industrial-strength triangulation of polygons. *Algorithmica*, 30:563–596, 2001.
- [Hel01b] M. Held. VRONI: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Comput. Geom. Theory Appl.*, 18:95–123, 2001.
- [HHK88] C. Hoffmann, J.E. Hopcroft, and M. Karasick. Towards implementing robust geometric computations. *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, 106–117, 1988.
- [HHK⁺01] S. Hert, M. Hoffmann, L. Kettner, S. Pion, and M. Seel. An adaptable and extensible geometry Kernel. *Proc. 5th Internat. Workshop Algorithm Eng. (WAE-01)*, Aarhus, pages 79–90, Springer-Verlag, Berlin, 2001.
- [Hob99] J.D. Hobby. Practical segment intersection with finite precision output. *Comput. Geom. Theory Appl.*, 13:199–214, 1999.
- [HS98] D. Halperin and C.R. Shelton. A perturbation scheme for spherical arrangements with applications to molecular modeling. *Comput. Geom. Theory Appl.*, 10:273–288, 1998.
- [HW94] P. Hertling and K. Weihrauch. Levels of degeneracy and exact lower complexity bounds for geometric algorithms. *Proc. 6th Canad. Conf. Comput. Geom.*, pages 237–242, 1994.
- [JW94] J.W. Jaromczyk and G.W. Wasilkowski. Computing convex hull in a floating point arithmetic. *Comput. Geom. Theory Appl.*, 4:283–292, 1994.
- [KFC⁺01] S. Krishnan, M. Foskey, T. Culver, J. Keyser, and D. Manocha. PRECISE: Efficient multiprecision evaluation of algebraic roots and predicates for reliable geometric computation. *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, 274–283, 2001.
- [KLN91] M. Karasick, D. Lieber, and L.R. Nackman. Efficient Delaunay triangulation using rational arithmetic. *ACM Trans. Graphics*, 10:71–91, 1991.
- [KLPY99] V. Karamcheti, C. Li, I. Pechtchanski, and C.K. Yap. A Core Library for robust numerical and geometric libraries. *15th Annu. ACM Sympos. Comput. Geom.*, pages 351–359, 1999.
- [KW98] L. Kettner and E. Welzl. One sided error predicates in geometric computing. In K. Mehlhorn, editor, *Proc. 15th IFIP World Computer Congress, Fundamentals—Foundations of Computer Science*, pages 13–26, 1998.
- [LPT97] G. Liotta, F.P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, 156–165, 1997.

- [LY01] C. Li and C.K. Yap. A new constructive root bound for algebraic expressions. *Proc. 12th ACM-SIAM Sympos. Discrete Algorithms*, pages 496–505, 2001.
- [Mic95] D. Michelucci. An epsilon-arithmetic for removing degeneracies. *Proc. IEEE 12th Sympos. Computer Arithmetic*, pages 230–237, Windsor, Ontario, 1995.
- [Mig82] M. Mignotte. Identification of algebraic numbers. *J. Algorithms*, 3:197–204, 1982.
- [MM93] V.J. Milenkovic and Ve. Milenkovic. Rational orthogonal approximations to orthogonal matrices. *Proc. 5th Canad. Conf. Comput. Geom.*, pages 485–490, Waterloo, 1993.
- [MN90] V.J. Milenkovic and L.R. Nackman. Finding compact coordinate representations for polygons and polyhedra. *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, 244–252, 1990.
- [MN95] K. Mehlhorn and S. Näher. LEDA: a platform for combinatorial and geometric computing. *Commun. ACM*, 38:96–102, 1995.
- [MNS⁺96] K. Mehlhorn, S. Näher, T. Schilz, R. Seidel, M. Seel, and C. Uhrig. Checking geometric programs or verification of geometric structures. *Proc. 12th ACM Symp. on Computational Geom.*, pages 159–165, 1996.
- [MW96] A. Macintyre and A. Wilkie. On the decidability of the real exponential field. *Kreiseliana, About and Around Georg Kreisel*, pages 441–467. A.K. Peters, Wellesley, 1996.
- [Neu97] M.A. Neuhauser. Symbolic perturbation and special arithmetics for controlled handling of geometric degeneracies. *Proc. 5th Internat. Conf. Central Europe Comput. Graphics Visualization (WSCG'97)*, pages 386–395, 1997.
- [OTU87] T.A. Ottmann, G. Thiemt, and C. Ullrich. Numerical stability of geometric algorithms. *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 119–125, 1987.
- [Pap85] C.H. Papadimitriou. An algorithm for shortest-path motion in three dimensions. *Inform. Process. Lett.*, 20:259–263, 1985.
- [PCH⁺95] N.M. Patrikalakis, W. Cho, C.-Y. Hu, T. Maekawa, E.C. Sherbrooke, and J. Zhou. Towards robust geometric modelers, 1994 progress report. *Proc. 1995 NSF Design Manufacturing Grantees Conf.*, pages 139–140, 1995.
- [PY01] V.Y. Pan and Y. Yu. Certification of numerical computation of the sign of the determinant of a matrix. *Algorithmica*, pages 708–724, 2001.
- [PY03] S. Pion and C.K. Yap. Constructive root bound method for k -ary rational input numbers. *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 256–263, 2003.
- [Raa99] S. Raab. Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes. *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1999.
- [Ric97] D. Richardson. How to recognize zero. *J. Symbolic Computation*, 24:627–645, 1997.
- [Ric00] D. Richardson. The uniformity conjecture. In J. Blank, V. Brattka, and P. Hertling, editors, *Computability and Complexity in Analysis. Lecture Notes Comput. Sci.*, volume 2064, pages 253–272, Springer-Verlag, Berlin, 2000.
- [Sch94] P. Schorn. Degeneracy in geometric computation and the perturbation approach. *Comput. J.*, 37:35–42, 1994.
- [Sch99] S. Schirra. Robustness and precision issues in geometric computation. In J.R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*. Elsevier Publishers, B.V. North-Holland, Amsterdam, 1999.
- [Sch00] E.R. Scheinerman. When close enough is close enough. *Amer. Math. Monthly*, 107:489–499, 2000.

- [SCY00] J. Sellen, J. Choi, and C. Yap. Precision-sensitive Euclidean shortest path in 3-Space. *SIAM J. Computing*, 29:1577–1595, 2000.
- [Sed83] R. Sedgewick. *Algorithms*. Addison-Wesley, Reading, 1983.
- [See01] M. Seel. *Planar Nef Polyhedra and Generic High-dimensional Geometry*. Ph.D. thesis, Universität des Saarlandes, 2001.
- [Seg90] M.G. Segal. Using tolerances to guarantee valid polyhedral modeling results. *Comput. Graph.*, 24:105–114, 1990.
- [Sei98] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998.
- [She96] J.R. Shewchuk. Robust adaptive floating-point geometric predicates. *Proc. 12th ACM Symp. on Computational Geom.*, pages 141–150, 1996.
- [SI89a] K. Sugihara and M. Iri. A solid modeling system free from topological inconsistency. *J. Inform. Proc., Inform. Proc. Soc. Japan*, 12:380–393, 1989.
- [SI89b] K. Sugihara and M. Iri. Two design principles of geometric algorithms in finite precision arithmetic. *Appl. Math. Lett.*, 2:203–206, 1989.
- [SI92] K. Sugihara and M. Iri. Construction of the Voronoi diagram for ‘one million’ generators in single-precision arithmetic. *Proc. IEEE*, 80:1471–1484, 1992.
- [SII00] K. Sugihara, M. Iri, H. Inagaki, and T. Imai. Topology-oriented implementation—an approach to robust geometric algorithms. *Algorithmica*, 27, 2000.
- [SS85] M.G. Segal and C.H. Séquin. Consistent calculations for solids modelling. *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 29–38, 1985.
- [SSTY00] E. Schömer, J. Sellen, M. Teichmann, and C. Yap. Smallest enclosing cylinders. *Algorithmica*, 27:170–186, 2000.
- [Sug89] K. Sugihara. On finite-precision representations of geometric objects. *J. Comput. Syst. Sci.*, 39:236–247, 1989.
- [Ull90] C. Ullrich, editor. *Computer Arithmetic and Self-validating Numerical Methods*. Academic Press, Boston, 1990.
- [Wei02] R. Wein. High level filtering for arrangements of conic arcs. *Proc. 10th European Sympos. Algorithms*, volume 2461 of *Lecture Notes Comput. Sci.*, pages 884–895. Springer-Verlag, Berlin, 2002.
- [Yap90a] C.K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40:2–18, 1990.
- [Yap90b] C.K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990.
- [Yap97] C.K. Yap. Towards exact geometric computation. *Comput. Geom. Theory Appl.*, 7:3–23, 1997.
- [Yap98] C.K. Yap. A new number core for robust numerical and geometric libraries. *Proc. 3rd CGC Workshop Geom. Comput.*, 1998. <http://www.cs.brown.edu/cgc/cgc98/home.html>.
- [Yap00] C.K. Yap. *Fundamental Problems in Algorithmic Algebra*. Oxford University Press, 2000.
- [YD95] C.K. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F.K. Hwang, editors, *Computing in Euclidean Geometry*, 2nd edition, pages 452–486. World Scientific Press, Singapore, 1995.
- [Yu92] J. Yu. *Exact arithmetic solid modeling*. Ph.D. thesis, Dept. of Comput. Sci., Purdue Univ., Tech. Rep. CSD-TR-92-037, 1992.

42 PARALLEL ALGORITHMS IN GEOMETRY

Michael T. Goodrich

INTRODUCTION

The goal of parallel algorithm design is to develop parallel computational methods that run very fast with as few processors as possible, and there is an extensive literature of such algorithms for computational geometry problems. There are several different parallel computing models, and in order to maintain a focus in this chapter, we will describe results in the Parallel Random Access Machine (PRAM) model, which is a synchronous parallel machine model in which processors share a common memory address space (and all inter-processor communication takes place through this shared memory). Although it does not capture all aspects of parallel computing, it does model the essential properties of parallelism. Moreover, it is a widely accepted model of parallel computation, and all other reasonable models of parallel computation can easily simulate a PRAM.

Interestingly, parallel algorithms can have a direct impact on efficient sequential algorithms, using a technique called *parametric search*. This technique, which is discussed in [Chapter 43](#), involves the use of a parallel algorithm to direct searches in a parameterized geometric space so as to find a critical location (e.g., where an important parameter changes sign or achieves a maximum or minimum value).

The PRAM model is subdivided into submodels based on how one wishes to handle concurrent memory access to the same location. The Exclusive-Read, Exclusive-Write (EREW) variant does not allow for concurrent access. The Concurrent-Read, Exclusive-Write (CREW) variant permits concurrent memory reads, but memory writes must be exclusive. Finally, the Concurrent-Read, Concurrent-Write (CRCW) variant allows for both concurrent memory reading and writing, with concurrent writes being resolved by some simple rule, such as having an arbitrary member of a collection of conflicting writes succeed. One can also define randomized versions of each of these models (e.g., an rCRCW PRAM), where in addition to the usual arithmetic and comparison operations, each processor can generate a random number from 1 to n in one step.

Early work in parallel computational geometry, in the way we define it here, began with the work of Chow [[Cho80](#)], who designed several parallel algorithms with polylogarithmic running times using a linear number of processors. Subsequent to this work, several researchers initiated a systematic study of work-efficient parallel algorithms for geometric problems, including Aggarwal *et al.* [[ACG⁺88](#)], Akl [[Akl82](#), [Akl84](#), [Akl85](#)], Amato and Preparata [[AP92](#), [AP95](#)], Atallah and Goodrich [[AG86](#), [Goo87](#)], and Reif and Sen [[RS92](#), [Sen89](#)].

In Section 42.1 we give a brief discussion of general techniques for parallel geometric algorithm design. We then partition the research in parallel computational geometry into problems dealing with convexity (Section 42.2), arrangements and decompositions (Section 42.3), proximity (Section 42.4), geometric searching (Section 42.5), and visibility, envelopes, and geometric optimization (Section 42.6).

42.1 SOME PARALLEL TECHNIQUES

The design of efficient parallel algorithms for computational geometry problems often depends upon the use of powerful general parallel techniques (e.g., see [AL93, Já92, KR90, Rei93]). We review some of these techniques below.

PARALLEL DIVIDE-AND-CONQUER

Possibly the most general technique is parallel divide-and-conquer. In applying this technique one divides a problem into two or more subproblems, solves the subproblems recursively in parallel, and then merges the subproblem solutions to solve the entire problem. As an example application of this technique, consider the problem of constructing the upper convex hull of a S set of n points in the plane presorted by x -coordinates. Divide the list S into $\lceil \sqrt{n} \rceil$ contiguous sublists of size $\lfloor \sqrt{n} \rfloor$ each and recursively construct the upper convex hull of the points in each list. Assign a processor to each pair of sublists and compute the common upper tangent line for the two upper convex hulls for these two lists, which can be done in $O(\log n)$ time using a well-known “binary search” computation [Ede87, O’R98, PS85]. By maximum computations on the left and right common tangents, respectively, for each subproblem S_i , one can determine which vertices on the upper convex hull of S_i belong to the upper convex hull of S . Compressing all the vertices identified to be on the upper convex hull of S constructs an array representation of this hull, completing the construction.

The running time of this method is characterized by the recurrence relation $T(n) \leq T(\sqrt{n}) + O(\log n)$, which implies that $T(n)$ is $O(\log n)$. It is important to note that the coefficient for the $T(\sqrt{n})$ term is 1 even though we had $\lceil \sqrt{n} \rceil$ subproblems, for all these subproblems were processed simultaneously in parallel. The number of processors needed for this computation can be characterized by the recurrence relation $P(n) \leq \max\{\lceil \sqrt{n} \rceil P(\sqrt{n}), n\}$, which implies that $P(n)$ is $O(n)$. Thus, the *work* needed for this computation is $O(n \log n)$, which is not quite optimal. Still, this method can be adapted to result in an optimal work bound [BSV96, Che95, GG97].

BUILD-AND-SEARCH

Another important technique in parallel computational geometry is the build-and-search technique. It is a paradigm that often yields efficient parallel adaptations of sequential algorithms designed using the powerful plane sweeping technique. In the build-and-search technique, the solution to a problem is partitioned into a *build* phase, where one constructs in parallel a data structure built from the geometric data present in the problem, and a *search* phase, where one searches this data structure in parallel to solve the problem at hand. An example of an application of this technique is for the trapezoidal decomposition problem: given a collection of nonintersecting line segments in the plane, determine the first segments intersected by vertical rays emanating from each segment endpoint (cf. Figure 40.0.1). The existing efficient parallel algorithm for this problem is based upon first building in parallel a data structure on the input set of segments that allows for such vertical

ray-shooting queries to be answered in $O(\log n)$ time by a single processor, and then querying this structure for each segment endpoint in parallel. This results in a parallel algorithm with an efficient $O(n \log n)$ work bound and fast $O(\log n)$ query time.

42.2 CONVEXITY

Results on the problem of constructing the convex hull of n points in \mathbb{R}^d are summarized in Table 42.2.1, for various fixed values of d , and, in the case of $d = 2$, under assumptions about whether the input is presorted. We restrict our attention to parallel algorithms with efficient work bounds, where we use the term **work** of an algorithm here to refer to the product of its running time and the number of processors used by the algorithm. A parallel algorithm has an **optimal** work bound if the work used asymptotically matches the sequential lower bound for the problem. In the table, h denotes the size of the hull, and c is some fixed constant. Also, we use (throughout this chapter) $\bar{O}(f(n))$ to denote an asymptotic bound that holds with high probability.

TABLE 42.2.1 Parallel convex hull algorithms.

PROBLEM	MODEL	TIME	WORK	REF
2D presorted	rand-CRCW	$\bar{O}(\log^* n)$	$\bar{O}(n)$	[GG91]
2D presorted	CRCW	$O(\log \log n)$	$O(n)$	[BSV96]
2D presorted	EREW	$O(\log n)$	$O(n)$	[Che95]
2D polygon	EREW	$O(\log n)$	$O(n)$	[Che95]
2D	rand-CRCW	$\bar{O}(\log n)$	$\bar{O}(n \log h)$	[GG91]
2D	EREW	$O(\log n)$	$O(n \log n)$	[MS88]
2D	EREW	$O(\log^2 n)$	$O(n \log h)$	[GG91]
3D	rand-CRCW	$\bar{O}(\log n)$	$\bar{O}(n \log n)$	[RS92]
3D	CREW	$O(\log n)$	$O(n^{1+1/c})$	[AP93]
3D	EREW	$O(\log^2 n)$	$O(n \log n)$	[AGR94]
3D	EREW	$O(\log^3 n)$	$O(n \log h)$	[AGR94]
Fixed $d \geq 4$	rand-EREW	$\bar{O}(\log^2 n)$	$\bar{O}(n^{\lfloor d/2 \rfloor})$	[AGR94]
Even $d \geq 4$	EREW	$O(\log^2 n)$	$O(n^{\lfloor d/2 \rfloor})$	[AGR94]
Odd $d > 4$	EREW	$O(\log^2 n)$	$O(n^{\lfloor d/2 \rfloor} \log^c n)$	[AGR94]

We discuss a few of these algorithms to illustrate their flavor.

2-DIMENSIONAL CONVEX HULLS

The two-dimensional convex hull algorithm of Miller and Stout [MS88] is based upon a parallel divide-and-conquer scheme where one presorts the input and then divides it into many subproblems ($O(n^{1/4})$ in their case), solves each subproblem independently in parallel, and then merges all the subproblem solutions together

in $O(\log n)$ parallel time. Of course, the difficult step is the merge of all the subproblems, with the principal difficulty being the computation of common tangents between hulls. The total running time is characterized by the recurrence

$$T(n) \leq T(n^{1/4}) + O(\log n),$$

which solves to $T(n) = O(\log n)$.

3-DIMENSIONAL CONVEX HULLS

All of the 3D convex hull algorithms listed in [Table 42.2.1](#) are also based upon this many-way, divide-and-conquer paradigm, except that there is no notion of presorting in three dimensions, so the subdivision step also becomes nontrivial. Reif and Sen [RS92] use a random sample to perform the division, and the methods of Amato, Goodrich, and Ramos [AGR94] derandomize this approach. Amato and Preparata [AP93] use parallel separating planes, an approach extended to higher dimension in [AGR94].

LINEAR PROGRAMMING

A problem strongly related to convex hull construction, which has also been addressed in a parallel setting, is d -dimensional linear programming, for fixed dimensions d (see [Chapter 45](#)). Of course, one could solve this problem by transforming it to its dual problem, constructing a convex hull in this dual space, and then evaluating each vertex in the simplex that is dual to this convex hull. This would be quite inefficient, however, for $d \geq 4$. The best parallel bounds for this problem are listed in [Table 42.2.2](#). See [Section 45.6](#) for a detailed discussion.

TABLE 42.2.2 Fixed d -dimensional parallel linear programming.

MODEL	TIME	WORK	REF
Rand-CRCW	$\bar{O}(1)$	$\bar{O}(n)$	[AM90]
CRCW	$O((\log \log n)^{d-1})$	$O(n)$	[GR97]
EREW	$O(\log n (\log \log n)^{d-1})$	$O(n)$	[Goo96]

OPEN PROBLEMS

There are a number of interesting open problems regarding convexity:

1. Can d -dimensional linear programming be solved (deterministically) in $O(\log n)$ time using $O(n)$ work in the CREW PRAM model?
2. Is there an efficient output-sensitive parallel convex hull algorithm for $d \geq 4$?
3. Is there an optimal-work $O(\log^2 n)$ -time CREW PRAM convex hull algorithm for odd dimensions greater than 4?

42.3 ARRANGEMENTS AND DECOMPOSITIONS

Another important class of geometric problems that has been addressed in the parallel setting are arrangement and decomposition problems, which deal with ways of partitioning space. We review the best parallel bounds for such problems in Table 42.3.1.

GLOSSARY

Arrangement: The partition of space determined by the intersections of a collection of geometric objects, such as lines, line segments, or (in higher dimensions) hyperplanes. In this chapter, algorithms for constructing arrangements produce the *incidence graph*, which stores all adjacency information between the various primitive topological entities determined by the partition, such as intersection points, edges, faces, etc. See [Section 24.3.1](#).

Red-blue arrangement: An arrangement defined by two sets of objects A and B such that the objects in A (resp. B) are nonintersecting.

Axis-parallel: All segments/lines are parallel to one of the coordinate axes.

Polygon triangulation: A decomposition of the interior of a polygon into triangles by adding diagonals between vertices. See [Section 26.2](#).

Trapezoidal decomposition: A decomposition of the plane into trapezoids (and possibly triangles) by adding appropriate vertical line segments incident to vertices. See [Section 34.3](#).

Star-shaped polygon: A (simple) polygon that is completely visible from a single point. A polygon with nonempty kernel. See [Section 26.1](#).

1/r-cutting: A partition of \mathbb{R}^d into $O(r^d)$ simplices such that each simplex intersects at most n/r hyperplanes. See [Sections 36.2](#) and [40.1](#).

TABLE 42.3.1 Parallel arrangement and decomposition algorithms.

PROBLEM	MODEL	TIME	WORK	REF
d -dim hyperplane arr	EREW	$O(\log n)$	$O(n^d)$	[AGR94]
2D seg arr	rand-CRCW	$\tilde{O}(\log n)$	$\tilde{O}(n \log n + k)$	[CCT92a, CCT92b]
2D axis-par seg arr	CREW	$O(\log n)$	$O(n \log n + k)$	[Goo91]
2D red-blue seg arr	CREW	$O(\log n)$	$O(n \log n + k)$	[GSG92, GSG93, Rüb92]
2D seg arr	EREW	$O(\log^2 n)$	$O(n \log n + k)$	[AGR95]
Polygon triangulation	CRCW	$O(\log n)$	$O(n)$	[Goo95]
Polygon triangulation	CREW	$O(\log n)$	$O(n \log n)$	[Goo89, Yap88]
2D nonint seg trap decomp	CREW	$O(\log n)$	$O(n \log n)$	[ACG89]
2D quadtree decomp	EREW	$O(\log n)$	$O(n \log n + k)$	[BET99]

We sketch the one randomized algorithm in Table 42.3.1 to illustrate how randomization and parallel computation can be mixed. Let S be a set of segments in

the plane with k intersecting pairs. The goal is to construct $\mathcal{A}(S)$, the arrangement induced by S . First, an estimate \hat{k} for k is obtained from a random sample. Then a random subset $R \subset S$ of a size r dependent on \hat{k} is selected. $\mathcal{A}(R)$ is constructed using a suboptimal parallel algorithm, and processed (in parallel) for point location. Next the segments intersecting each cell of $\mathcal{A}(R)$ are found using a parallel point-location algorithm, together with some ad hoc techniques. Visibility information among the segments meeting each cell is computed using another suboptimal parallel algorithm. Finally, the resulting cells are merged in parallel. Because various key parameters in the suboptimal algorithms are kept small by the sampling, optimal expected work is achieved.

All of the algorithms for computing segment arrangements are *output-sensitive*, in that their work bounds depend upon both the input size and the output size. In these cases we must slightly extend our computational model to allow for the machine to request additional processors if necessary. In all these algorithms, this request may originate only from a single “master” processor, however, so this modification is not that different from our assumption that the number of processors assigned to a problem can be a function of the input size. Of course, to solve a problem on a real parallel computer, one would simulate one of these efficient parallel algorithms to achieve an optimal speed-up over what would be possible using a sequential method.

A related class of intersection-related problems is the class of problems dealing with methods for detecting intersections. Testing if a collection of objects has at least one intersection is frequently easier than finding all such intersections, and Table 42.3.2 reviews such results in the parallel domain.

GLOSSARY

Star-shaped polygon: A (simple) polygon that is completely visible from a single point; a polygon with non-empty kernel. See [Chapter 26](#).

TABLE 42.3.2 Parallel intersection detection algorithms.

PROBLEM	MODEL	TIME	WORK	REF
2 convex polygons	CREW	$O(1)$	$O(n^{1/c})$	[DK89a]
2 star-shaped polygons	CREW	$O(\log n)$	$O(n)$	[GM91]
2 convex polyhedra	CREW	$O(\log n)$	$O(n)$	[DK89a]

Given a collection of n hyperplanes in \mathbb{R}^d , another important decomposition problem is the construction of a $(1/r)$ -cutting. Here an EREW algorithm running in $O(\log n \log r)$ time using $O(nr^{d-1})$ work has been obtained [Goo93].

OPEN PROBLEMS

1. Is there an optimal-work $O(\log n)$ -time polygon triangulation algorithm that does not use concurrent writes?
 2. Can a line segment arrangement be constructed in $O(\log n)$ time using $O(n \log n + k)$ work in the CREW PRAM model?
-

42.4 PROXIMITY

An important property of Euclidean space is that it is a metric space, and distance plays an important role in many computational geometry applications. For example, computing a closest pair of points can be used in collision detection, as can the more general problem of computing the nearest neighbor of each point in a set S , a problem we will call the ***all-nearest neighbors (ANN)*** problem. Perhaps the most fundamental problem in this domain is the subdivision of space into regions where each region $V(s)$ is defined by a *site* s in a set S of geometric objects such that each point in $V(s)$ is closer to s than to any other object in S . This subdivision is the ***Voronoi diagram*** (Chapter 23); its graph-theoretic dual, which is also an important geometric structure, is the ***Delaunay triangulation*** (Section 25.1). For a set of points S in \mathbb{R}^d , there is a simple “lifting” transformation that takes each point $(x_1, x_2, \dots, x_d) \in S$ to the point $(x_1, x_2, \dots, x_d, x_1^2 + x_2^2 + \dots + x_d^2)$, forming a set of points S' in \mathbb{R}^{d+1} (Section 23.1). Each simplex on the convex hull of S' with a negative $(d+1)$ -st component in its normal vector projects back to a simplex of the Delaunay triangulation in \mathbb{R}^d . Thus, any $(d+1)$ -dimensional convex hull algorithm immediately implies a d -dimensional Voronoi diagram (VD) algorithm. Table 42.4.1 summarizes the bounds of efficient parallel algorithms for constructing Voronoi diagrams in this way, as well as methods that are designed particularly for Voronoi diagram construction or other specific proximity problems. (In the table, the underlying objects are points unless stated otherwise.)

GLOSSARY

Convex position: A set of points that are all on the boundary of their convex hull.

Voronoi diagram for line segments: A Voronoi diagram that is defined by a set of nonintersecting line segments, with distance from a point p to a segment s being defined as the distance from p to a closest point on s . See Section 23.3.

OPEN PROBLEMS

1. Can a 2D Voronoi diagram be constructed in $O(\log n)$ time using $O(n \log n)$ work under either the CREW or EREW PRAM models?

TABLE 42.4.1 Parallel proximity algorithms.

PROBLEM	MODEL	TIME	WORK	REF
2D ANN in convex pos	EREW	$O(\log n)$	$O(n)$	[CG92]
2D ANN	EREW	$O(\log n)$	$O(n \log n)$	[CG92]
d -dim ANN	CREW	$O(\log n)$	$O(n \log n)$	[Cal93]
2D VD in L_1 metric	CREW	$O(\log n)$	$O(n \log n)$	[WC90]
2D VD	rand-CRCW	$\bar{O}(\log n)$	$\bar{O}(n \log n)$	[RS92]
2D VD	CRCW	$O(\log n \log \log n)$	$O(n \log n \log \log n)$	[CGÓ90]
2D VD	EREW	$O(\log^2 n)$	$O(n \log n)$	[AGR94]
2D VD for segments	CREW	$O(\log^2 n)$	$O(n \log^2 n)$	[GÓY93]
3D VD	EREW	$O(\log^2 n)$	$O(n^2)$	[AGR94]

2. Is there an efficient output-sensitive parallel algorithm for constructing 3D Voronoi diagrams?

42.5 GEOMETRIC SEARCHING

Given a subdivision of space by a collection S of geometric objects, such as line segments, the point location problem is to build a data structure for this set that can quickly answer *vertical ray-shooting queries*, where one is given a point p and asked to report the first object in S hit by a vertical ray from p . We summarize efficient parallel algorithms for planar point location in Table 42.5.1. The time and work bounds listed, as well as the computational model, are for building the data structure to achieve an $O(\log n)$ query time. We do not list the space bounds for any of these methods in the table since, in every case, they are equal to the preprocessing work bounds.

GLOSSARY

Arbitrary planar subdivision: A subdivision of the plane (not necessarily connected), defined by a set of line segments that intersect only at their endpoints.

Monotone subdivision: A connected subdivision of the plane in which each face is intersected by a vertical line in a single segment.

Triangulated subdivision: A connected subdivision of the plane into triangles whose corners are vertices of the subdivision (see Chapter 25).

Shortest path in a polygon: The shortest path between two points that does not go outside of the polygon (see Section 26.4).

Ray-shooting query: A query whose answer is the first object hit by a ray oriented in a specified direction from a specified point.

TABLE 42.5.1 Parallel geometric searching algorithms.

QUERY PROBLEM	MODEL	TIME	WORK	REF
Point loc in arb subdivision	CREW	$O(\log n)$	$O(n \log n)$	[ACG89]
Point loc in monotone subdivision	EREW	$O(\log n)$	$O(n)$	[TV91]
Point loc in triangulated subdivision	CREW	$O(\log n)$	$O(n)$	[CZ90]
Point loc in d -dim hyp arr	EREW	$O(\log n)$	$O(n^d)$	[AGR94]
Shortest path in triangulated polygon	CREW	$O(\log n)$	$O(n)$	[GSG92]
Ray shooting in triangulated polygon	CREW	$O(\log n)$	$O(n)$	[HS93]
Line & convex polyhedra intersection	CREW	$O(\log n)$	$O(n)$	[DK89b, CZ90]

OPEN PROBLEMS

1. Is there an efficient data structure that allows n simultaneous point locations to be performed in $O(\log n)$ time using $O(n)$ processors in the EREW PRAM model?
2. Is there an efficient data structure for 3-dimensional point location in convex subdivisions that can be constructed in $O(n \log n)$ work and at most $O(\log^2 n)$ time and which allows for a query time that is at most $O(\log^2 n)$?

42.6 VISIBILITY, ENVELOPES, AND OPTIMIZATION

We summarize efficient parallel methods for various visibility and lower envelope problems for a simple polygon in Table 42.6.1. In the table, m denotes the number of edges in a visibility graph. For definitions see [Chapter 28](#).

TABLE 42.6.1 Parallel visibility algorithms for a simple polygon.

PROBLEM	MODEL	TIME	WORK	REF
Kernel	EREW	$O(\log n)$	$O(n)$	[Che95]
Vis from a point	EREW	$O(\log n)$	$O(n)$	[ACW91]
Vis from an edge	CRCW	$O(\log n)$	$O(n)$	[Her92]
Vis from an edge	CREW	$O(\log n)$	$O(n \log n)$	[GSG92, GSG93]
Vis graph	CREW	$O(\log n)$	$O(n \log^2 n + m)$	[GSG92, GSG93]

We sketch the algorithm for computing the point visibility polygon [ACW91], which is notable for two reasons: first, it is employed as a subprogram in many other algorithms; and second, it requires much more intricate processing and analysis than the relatively simple optimal sequential algorithm ([Section 25.3](#)). The parallel algorithm is recursive, partitioning the boundary into $n^{1/4}$ subchains, and computing *visibility chains* from the source point of visibility x . Each of these chains

is star-shaped with respect to x , i.e., effectively “monotone” (see [Section 26.1](#)). This monotonicity property is, however, insufficient to intersect the visibility chains quickly enough in the merge step to obtain optimal bounds. Rather, the fact that the chains are subchains of the boundary of a simple polygon must be exploited to achieve logarithmic-time computation of the intersection of two chains. This then leads to the optimal bounds quoted in [Table 42.6.1](#).

The bounds of efficient parallel methods for visibility problems on general sets of segments and curves in the plane are summarized in [Table 42.6.2](#).

GLOSSARY

Lower envelope: The function $F(x)$ defined as the pointwise minimum of a collection of functions $\{f_1, f_2, \dots, f_n\}$: $F(x) = \min_i f_i(x)$ (see [Section 21.2](#)).

k -intersecting curves: A set of curves every two of which intersect at most k times (where they cross).

$\lambda_s(n)$: The maximum length of a Davenport-Schinzel sequence [SA95, AS00] of order s on n symbols. If s is a constant, $\lambda_s(n)$ is $o(n \log^* n)$. See [Section 40.4](#).

TABLE 42.6.2 General parallel visibility and enveloping algorithms.

PROBLEM	MODEL	TIME	WORK	REF
Lower env for segments	EREW	$O(\log^2 n)$	$O(n \log n)$	[Her89]
Lower env for k -int curves	EREW	$O(\log^2 n)$	$O(\lambda_{k+2}(n) \log n)$	[BM87]

Finally, we summarize some efficient parallel algorithms for solving several geometric optimization problems in [Table 42.6.3](#).

GLOSSARY

Largest-area empty rectangle: For a collection S of n points in the plane, the largest-area rectangle that does not contain any point of S in its interior.

All-farthest neighbors problem in a simple polygon: Determine for each vertex p of a simple polygon the vertex q such that the shortest path from p is longest.

Closest visible-pair between polygons: A closest pair of mutually-visible vertices between two nonintersecting simple polygons in the plane.

Minimum circular-arc cover: For a collection of n arcs of a given circle C , a minimum-cardinality subset that covers C .

Optimal-area inscribed/circumscribed triangle: For a convex polygon P , the largest-area triangle inscribed in P , or, respectively, the smallest-area triangle circumscribing P .

Min-link path in a polygon: A piecewise-linear path of fewest “links” inside a simple polygon between two given points p and q ; see [Sections 23.4](#) and [24.3](#).

TABLE 42.6.3 Parallel geometric optimization algorithms.

PROBLEM	MODEL	TIME	WORK	REF
Largest-area empty rectangle	CREW	$O(\log^2 n)$	$O(n \log^3 n)$	[AKPS90]
All-farthest neighbors in polygon	CREW	$O(\log^2 n)$	$O(n \log^2 n)$	[Guh92]
Closest visible-pair btw polygons	CREW	$O(\log n)$	$O(n \log n)$	[HCL92]
Min circular-arc cover	EREW	$O(\log n)$	$O(n \log n)$	[AC89]
Opt-area inscr/circum triangle	CRCW	$O(\log \log n)$	$O(n)$	[CM92]
Opt-area inscr/circum triangle	CREW	$O(\log n)$	$O(n)$	[CM92]
Min-link path in a polygon	CREW	$O(\log n \log \log n)$	$O(n \log n \log \log n)$	[CGM ⁺ 90]

OPEN PROBLEMS

1. Can the visibility graph of a set of n nonintersecting line segments be constructed using $O(n \log n + m)$ work in time at most $O(\log^2 n)$ in the CREW model, where m is the size of the graph?
2. Can the visibility graph of a triangulated polygon be computed in $O(\log n)$ time using $O(n + m)$ work in the CREW model?

42.7 SOURCES AND RELATED MATERIAL

FURTHER READING

Our presentation has been results-oriented and has not provide much problem intuition or algorithmic techniques. There are several excellent surveys available in the literature [Ata92, AC94, AC00, AG93, RS93, RS00] that are more techniques-oriented. Another good location for related material is the book by Akl and Lyons [AL93].

RELATED CHAPTERS

- [Chapter 22: Convex hull computations](#)
- [Chapter 23: Voronoi diagrams and Delaunay triangulations](#)
- [Chapter 24: Arrangements](#)
- [Chapter 26: Polygons](#)
- [Chapter 34: Point location](#)
- [Chapter 38: Geometric intersection](#)
- [Chapter 40: Randomization and derandomization](#)
- [Chapter 45: Linear programming](#)

REFERENCES

- [AS00] P.K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47. Elsevier North-Holland, Amsterdam, 2000.
- [ACG⁺88] A. Aggarwal, B. Chazelle, L.J. Guibas, C. Ó'Dúnlaing, and C.K. Yap. Parallel computational geometry. *Algorithmica*, 3:293–327, 1988.
- [AKPS90] A. Aggarwal, D. Kravets, J.K. Park, and S. Sen. Parallel searching in generalized Monge arrays with applications. In *Proc. 2nd Annu. ACM Sympos. Parallel Algorithms Architect.*, pages 259–268, 1990.
- [Akl82] S.G. Akl. A constant-time parallel algorithm for computing convex hulls. *BIT*, 22:130–134, 1982.
- [Akl84] S.G. Akl. Optimal algorithms for computing convex hulls and for sorting. *Computing*, 33:1–11, 1984.
- [Akl85] S.G. Akl. Optimal parallel algorithms for selection, sorting and computing convex hulls. In G.T. Toussaint, editor, *Computational Geometry*, pages 1–22. North-Holland, Amsterdam, 1985.
- [AL93] S.G. Akl and K.A. Lyons. *Parallel Computational Geometry*. Prentice-Hall, Englewood Cliffs, 1993.
- [AM90] N. Alon and N. Megiddo. Parallel linear programming in fixed dimension almost surely in constant time. In *Proc. 31st Annu. IEEE Sympos. Found. Comput. Sci.*, pages 574–582, 1990.
- [AGR94] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Parallel algorithms for higher-dimensional convex hulls. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 683–694, 1994.
- [AGR95] N.M. Amato, M.T. Goodrich, and E.A. Ramos. Computing faces in segment and simplex arrangements. In *Proc. 27th Annu. ACM Sympos. Theory Comput.*, pages 672–682, 1995.
- [AP92] N.M. Amato and F.P. Preparata. The parallel 3D convex hull problem revisited. *Internat. J. Comput. Geom. Appl.*, 2:163–173, 1992.
- [AP93] N.M. Amato and F.P. Preparata. An NC¹ parallel 3D convex hull algorithm. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 289–297, 1993.
- [AP95] N.M. Amato and F.P. Preparata. A time-optimal parallel algorithm for three-dimensional convex hulls. *Algorithmica*, 14:169–182, 1995.
- [Ata92] M.J. Atallah. Parallel techniques for computational geometry. *Proc. IEEE*, 80:1435–1448, 1992.
- [AC89] M.J. Atallah and D.Z. Chen. An optimal parallel algorithm for the minimum circle-cover problem. *Inform. Process. Lett.*, 34:159–165, 1989.
- [AC94] M.J. Atallah and D.Z. Chen. Parallel computational geometry. In A.Y. Zomaya, editor, *Parallel Computations: Paradigms and Applications*. World Scientific, Singapore, 1994.
- [ACW91] M.J. Atallah, D.Z. Chen, and H. Wagener. Optimal parallel algorithm for visibility of a simple polygon from a point. *J. Assoc. Comput. Mach.*, 38:516–553, 1991.
- [ACG89] M.J. Atallah, R. Cole, and M.T. Goodrich. Cascading divide-and-conquer: A technique for designing parallel algorithms. *SIAM J. Comput.*, 18:499–532, 1989.

- [AG86] M.J. Atallah and M.T. Goodrich. Efficient parallel solutions to some geometric problems. *J. Parallel Distrib. Comput.*, 3:492–507, 1986.
- [AG93] M.J. Atallah and M.T. Goodrich. Deterministic parallel computational geometry. In J.H. Reif, editor, *Synthesis of Parallel Algorithms*, pages 497–536. Morgan Kaufmann, San Mateo, 1993.
- [AC00] M.J. Atallah and D.Z. Chen. Deterministic parallel computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 155–200. Elsevier North-Holland, Amsterdam, 2000.
- [BSV96] O. Berkman, B. Schieber, and U. Vishkin. A fast parallel algorithm for finding the convex hull of a sorted point set. *Internat. J. Comput. Geom. Appl.*, 6:231–242, 1996.
- [BET99] M. Bern, D. Eppstein, and S.-H. Teng. Parallel construction of quadtrees and quality triangulations. *Internat. J. Comput. Geom. Appl.*, 9:517–532, 1999.
- [BM87] L. Boxer and R. Miller. Parallel dynamic computational geometry. Report 87-11, Dept. Comput. Sci., SUNY-Buffalo, 1987.
- [Cal93] P.B. Callahan. Optimal parallel all-nearest-neighbors using the well-seated pair decomposition. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 332–340, 1993.
- [CM92] S. Chandran and D.M. Mount. A parallel algorithm for enclosed and enclosing triangles. *Internat. J. Comput. Geom. Appl.*, 2:191–214, 1992.
- [CGM⁺90] V. Chandru, S.K. Ghosh, A. Maheshwari, V.T. Rajan, and S. Saluja. *NC*-algorithms for minimum link path and related problems. Technical Report CS-90/3, Tata Inst., Bombay, India, 1990.
- [Che95] D.Z. Chen. Efficient geometric algorithms on the EREW PRAM. *IEEE Trans. Parallel Distrib. Syst.*, 6:41–47, 1995.
- [Cho80] A.L. Chow. *Parallel algorithms for geometric problems*. Ph.D. thesis, Dept. Comput. Sci., Univ. Illinois, Urbana, 1980.
- [CCT92a] K.L. Clarkson, R. Cole, and R.E. Tarjan. Erratum: Randomized parallel algorithms for trapezoidal diagrams. *Internat. J. Comput. Geom. Appl.*, 2:341–343, 1992.
- [CCT92b] K.L. Clarkson, R. Cole, and R.E. Tarjan. Randomized parallel algorithms for trapezoidal diagrams. *Internat. J. Comput. Geom. Appl.*, 2:117–133, 1992.
- [CG92] R. Cole and M.T. Goodrich. Optimal parallel algorithms for polygon and point-set problems. *Algorithmica*, 7:3–23, 1992.
- [CGÓ90] R. Cole, M.T. Goodrich, and C. Ó'Dúnlaing. Merging free trees in parallel for efficient Voronoi diagram construction. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, volume 443 of *Lecture Notes Comput. Sci.*, pages 432–445. Springer-Verlag, Berlin, 1990.
- [CZ90] R. Cole and O. Zajicek. An optimal parallel algorithm for building a data structure for planar point location. *J. Parallel Distrib. Comput.*, 8:280–285, 1990.
- [DK89a] N. Dadoun and D.G. Kirkpatrick. Cooperative subdivision search algorithms with applications. In *Proc. 27th Allerton Conf. Commun. Control Comput.*, pages 538–547, 1989.
- [DK89b] N. Dadoun and D.G. Kirkpatrick. Parallel construction of subdivision hierarchies. *J. Comput. Syst. Sci.*, 39:153–165, 1989.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monogr. Theoret. Comput. Sci.* Springer-Verlag, Heidelberg, 1987.

- [GM91] S.K. Ghosh and A. Maheshwari. An optimal parallel algorithm for determining the intersection type of two star-shaped polygons. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 2–6, 1991.
- [GG97] M. Ghouse and M.T. Goodrich. Fast randomized parallel methods for planar convex hull construction. *Comput. Geom. Theory Appl.*, 7:219–236, 1997.
- [GG91] M. Ghouse and M.T. Goodrich. In-place techniques for parallel convex hull algorithms. In *Proc. 3rd Annu. ACM Sympos. Parallel Algorithms Architect.*, pages 192–203, 1991.
- [Goo87] M.T. Goodrich. *Efficient parallel techniques for computational geometry*. Ph.D. thesis, Dept. Comput. Sci., Purdue Univ., West Lafayette, 1987.
- [Goo89] M.T. Goodrich. Triangulating a polygon in parallel. *J. Algorithms*, 10:327–351, 1989.
- [Goo91] M.T. Goodrich. Intersecting line segments in parallel with an output-sensitive number of processors. *SIAM J. Comput.*, 20:737–755, 1991.
- [Goo93] M.T. Goodrich. Geometric partitioning made easier, even in parallel. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 73–82, 1993.
- [Goo95] M.T. Goodrich. Planar separators and parallel polygon triangulation. *J. Comput. Syst. Sci.*, 51:374–389, 1995.
- [GÓY93] M.T. Goodrich, C. Ó'Dúnlaing, and C.K. Yap. Constructing the Voronoi diagram of a set of line segments in parallel. *Algorithmica*, 9:128–141, 1993.
- [GR97] M.T. Goodrich and E.A. Ramos. Bounded-independence derandomization of geometric partitioning with applications to parallel fixed-dimensional linear programming. *Discrete Comput. Geom.*, 18:397–420, 1997.
- [GSG92] M.T. Goodrich, S. Shauck, and S. Guha. Parallel methods for visibility and shortest path problems in simple polygons. *Algorithmica*, 8:461–486, 1992.
- [GSG93] M.T. Goodrich, S. Shauck, and S. Guha. Addendum to “parallel methods for visibility and shortest path problems in simple polygons.” *Algorithmica*, 9:515–516, 1993.
- [Goo96] M.T. Goodrich. Fixed-dimensional parallel linear programming via relative epsilon-approximations. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 132–141, 1996.
- [Guh92] S. Guha. Parallel computation of internal and external farthest neighbours in simple polygons. *Internat. J. Comput. Geom. Appl.*, 2:175–190, 1992.
- [Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.
- [Her92] J. Hershberger. Optimal parallel algorithms for triangulated simple polygons. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 33–42, 1992.
- [HS93] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 54–63, January 1993.
- [HCL92] F.R. Hsu, R.C. Chang, and R.C.T. Lee. Parallel algorithms for computing the closest visible vertex pair between two polygons. *Internat. J. Comput. Geom. Appl.*, 2:135–162, 1992.
- [Já92] J. JáJá. *An Introduction to Parallel Algorithms*. Addison-Wesley, Reading, 1992.
- [KR90] R.M. Karp and V. Ramachandran. Parallel algorithms for shared memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 869–941. Elsevier/The MIT Press, Amsterdam, 1990.
- [MS88] R. Miller and Q.F. Stout. Efficient parallel convex hull algorithms. *IEEE Trans. Comput.*, C-37:1605–1618, 1988.

- [O'R98] J. O'Rourke. *Computational Geometry in C*, 2nd edition. Cambridge University Press, 1998.
- [PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [RS93] S. Rajasekaran and S. Sen. Random sampling techniques and parallel algorithms design. In J.H. Reif, editor, *Synthesis of Parallel Algorithms*, pages 411–452. Morgan Kaufmann, San Mateo, 1993.
- [Rei93] J.H. Reif. *Synthesis of Parallel Algorithms*. Morgan Kaufmann, San Mateo, 1993.
- [RS92] J.H. Reif and S. Sen. Optimal parallel randomized algorithms for three-dimensional convex hulls and related problems. *SIAM J. Comput.*, 21:466–485, 1992.
- [RS00] J.H. Reif and S. Sen. Parallel computational geometry: An approach using randomization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 765–828. Elsevier North-Holland, Amsterdam, 2000.
- [Rüb92] C. Rüb. Computing intersections and arrangements for red-blue curve segments in parallel. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 115–120, 1992.
- [Sen89] S. Sen. *Random Sampling Techniques for Efficient Parallel Algorithms in Computational Geometry*. Ph.D. thesis, Dept. Comput. Sci., Duke Univ., 1989.
- [SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [TV91] R. Tamassia and J.S. Vitter. Parallel transitive closure and point location in planar structures. *SIAM J. Comput.*, 20:708–725, 1991.
- [WC90] Y.C. Wee and S. Chaiken. An optimal parallel L_1 -metric Voronoi diagram algorithm. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 60–65, 1990.
- [Yap88] C.K. Yap. Parallel triangulation of a polygon in two calls to the trapezoidal map. *Algorithmica*, 3:279–288, 1988.

43 PARAMETRIC SEARCH

Jeffrey S. Salowe

INTRODUCTION

Parametric search is a technique that can sometimes be used to solve an optimization problem when there is an efficient algorithm for the related decision problem. If successful, one creates an optimization algorithm that makes only a small number of calls to the decision algorithm. We provide a general description (Section 43.1) and four examples (Sections 43.2–43.5) to illustrate the technique.

43.1 PARAMETRIC SEARCH OVERVIEW

GLOSSARY

Monotonic function: A function $f(x)$ having the property that $f(y) \geq f(x)$ if $y > x$.

Root-finding problem: Determining the largest value θ^* of θ with the property that $f(\theta^*) = 0$.

Monotonic root-finding problem: A root-finding problem where $f(\theta)$ is monotonically increasing in θ .

Fixed-value problem: Evaluating $f(\theta)$ for a given value of θ .

Parametric search: A technique to solve efficiently suitable monotonic root-finding problems.

WHAT IS PARAMETRIC SEARCH?

The parametric search technique was invented by Megiddo [Meg79, Meg83] as a technique to solve certain optimization problems. Parametric search is particularly effective if the optimization problem can be phrased as a monotonic root-finding problem and if an efficient algorithm for the fixed-value problem can be constructed.

More specifically, let $f(\theta)$ be a monotonic function with a root, and suppose our optimization problem is to determine $\theta^* = \sup\{\theta \mid f(\theta) = 0\}$. (Our notation emphasizes the dependence on the parameter θ , but it obscures the dependence of certain functions on the problem inputs.) Let $A(\theta)$ be an algorithm that computes $f(\theta)$, written in the form of a binary *decision tree* whose nodes s correspond to inequalities $g_s(\theta) \geq 0$. The parametric search technique evaluates $f(\theta^*)$, and in the process discovers θ^* , by evaluating the sign of $f(\theta)$ at some of the roots of $g_s(\theta)$.

The technique works best when each $g_s(\theta)$ has at most a constant number of roots and when $A(\theta)$ is an efficient parallel algorithm.

WHAT IS ITS EFFECT?

Parametric search generally yields the following results. Suppose that the optimization problem has n inputs and the decision problem has $n+1$ inputs, the additional input being for the parameter θ . If $A(\theta)$ computes $f(\theta)$ sequentially in $S(n)$ time, θ^* can be found in $O((S(n))^2)$ time. If $B(\theta)$ is an efficient parallel algorithm to compute $f(\theta)$ that runs in $T(n)$ time using $P(n)$ processors, θ^* can be found in $O(S(n)T(n) \log P(n) + T(n)P(n))$ time. Under favorable conditions, parametric search solves an optimization problem in $O(\log^c n)$ $f(\theta)$ evaluations, where c is a small constant.

HOW IS IT APPLIED?

It is sometimes difficult to determine whether a given problem can be phrased as a root-finding problem suitable for parametric search. As a guideline, we illustrate the parametric search technique through a series of examples. The examples are picked for their illustrative value, and we do not necessarily derive the most efficient results known. Instead, we demonstrate the efficacy of the technique by obtaining surprisingly efficient solutions. Parametric search was used on the problems mentioned in Sections 43.3–43.5 to substantially improve the time complexity over previous techniques.

43.2 EXAMPLE 1: QUARTERING THE PLANE

GLOSSARY

Planar ham-sandwich cut: A line that simultaneously bisects two planar sets. (See [Sections 11.2](#) and [31.2](#).)

Median: A number $x \in A$ with the property that at most half of the numbers in A are less than x , and at most half of the numbers in A are greater than x .

General position: A condition on a set of points that forbids certain configurations. A typical general position assumption is that no three points in the plane are collinear.

PROBLEM STATEMENT

- Input: Set $U = \{u_1, \dots, u_n\}$, consisting of n points in the plane, each point satisfying $y(u_i) > 0$, where $y(u)$ is the y -coordinate of point u . Set $L = \{l_1, \dots, l_n\}$, a set of n points in the plane, each point satisfying $y(l_i) < 0$.

- Output: A planar ham-sandwich cut for U and L .

We assume that the points are in general position (no three points collinear and no two points with the same y -coordinates), that the input values are rational, and that n is an odd positive integer. In this case, the ham-sandwich cut is unique. These conditions simplify the explanation of the algorithm.

CHOICE OF MONOTONIC FUNCTION

The quartering problem is not immediately in the form of a monotonic root-finding problem, but it can be converted to one in the following manner. Let θ be an angle with respect to the x -axis, $0 < \theta < \pi$, measured in the usual way, and let $m(\theta, U)$ denote the intersection of the x -axis with the line at angle θ that bisects U . Let $m(\theta, L)$ be the analogous quantity for set L . We seek an angle θ such that $f(\theta) = m(\theta, U) - m(\theta, L) = 0$. Because a ham-sandwich cut exists, there is a value θ^* of θ for which $f(\theta^*) = 0$; our assumptions above ensure that θ^* is unique.

With this choice of the $f(\theta)$ function, the quartering problem seems to be a good candidate for parametric search. The function $f(\theta)$ is monotonic in θ , the quartering problem is solved if and only if $f(\theta^*) = 0$, and the value of $f(\theta)$ is easily computed, as described below.

FIXED-VALUE EVALUATION

To compute $f(\theta)$, first consider the U points. If these points are projected onto the x -axis along lines at an angle of θ , we have n one-dimensional points. The median of these projected points is precisely $m(\theta, U)$. Similarly, the median of the projected L points is $m(\theta, L)$. The evaluation of $f(\theta)$ amounts to:

1. Determining $m(\theta, U)$ by a median-find procedure.
2. Determining $m(\theta, L)$ by a median-find procedure.
3. Calculating $f(\theta) = m(\theta, U) - m(\theta, L)$.

The median-find procedure is a comparison-based algorithm that runs sequentially in $O(n)$ time and in parallel in $O(\log n)$ time using $O(n/\log n)$ processors. This is our algorithm $A(\theta)$.

THE DECISION-TREE ALGORITHM

We now rewrite $A(\theta)$ as a decision-tree algorithm and examine its comparisons. The median-find algorithm is central to $A(\theta)$. The generic step $s(i, j)$ of the median-find algorithm is to compare α_i and α_j , where α_i and α_j are two of the inputs; here the input values $\alpha_i(\theta)$ and $\alpha_j(\theta)$ are the projections of points $u_i = (x_i, y_i)$ and $u_j = (x_j, y_j)$ along a line with angle θ . It is apparent that $\alpha_i(\theta) = x_i - y_i \cot \theta$ and $\alpha_j(\theta) = x_j - y_j \cot \theta$. The decision tree node $s(i, j)$ corresponds to $g_{s(i,j)}(\theta) = x_i - x_j + (y_j - y_i) \cot \theta \geq 0$. There are no other branch points in the algorithm that depend on θ .

The function $g_{s(i,j)}$ has one root, $\theta_{s(i,j)} = \tan^{-1} \frac{y_j - y_i}{x_j - x_i}$. This is because the function $\cot(\theta)$ is monotonically decreasing in the range $0 < \theta < \pi$ and takes on all

values. Although the exact numerical value of $\theta_{s(i,j)}$ is generally unavailable, the sign of $f(\theta_{s(i,j)})$ can be evaluated. Consider comparison $s(m,n)$ in the computation of $f(\theta_{s(i,j)})$. The value of the function

$$g_{s(m,n)}(\theta_{s(i,j)}) = x_m - x_n + (y_n - y_m) \frac{x_j - x_i}{y_j - y_i}$$

is rational if the inputs are rational. Furthermore, the truth value of $\theta_{s(i,j)} < \theta_{s(i',j')}$ can be determined without the actual numerical values of $\theta_{s(i,j)}$ and $\theta_{s(i',j')}$: the truth value of $\theta_{s(i,j)} < \theta_{s(i',j')}$ is the same as the truth value of

$$\frac{y_j - y_i}{x_j - x_i} < \frac{y_{j'} - y_{i'}}{x_{j'} - x_{i'}}.$$

These two observations are needed below.

EVALUATING $f(\theta^*)$

Recall that we seek θ^* , the value of θ for which $f(\theta^*) = 0$. Suppose we try to run the algorithm $A(\theta^*)$ for $f(\theta^*)$, even though we do not know θ^* . Our main difficulty is resolving comparisons that depend on the value of θ^* .

Algorithm $A(\theta^*)$ is in the form of a decision tree, where each node s is labeled with inequality $g_s(\theta^*) \geq 0$. In order to resolve these decisions, we must determine the truth values of $g_s(\theta^*) \geq 0$.

These truth values are determined as follows. (This is the crucial step in parametric search.) The function $g_s(\theta)$ has one root, θ_s . Furthermore, $g_s(\theta)$ is monotonically decreasing in θ , so we can therefore determine the truth value of $g_s(\theta^*) \geq 0$ by determining the relative values of θ^* and θ_s . The relative values of θ^* and θ_s can be inferred by evaluating the sign of the fixed-value problem $f(\theta_s)$. Because $f(\theta)$ is monotonic, $f(\theta_s) < 0$ implies that $\theta_s < \theta^*$, and $f(\theta_s) > 0$ implies that $\theta_s > \theta^*$. If $f(\theta_s) = 0$, then $\theta^* = \theta_s$, and we have the value we seek. As stated above, the sign of $f(\theta_s)$ can be determined at the roots of $g_s(\theta)$.

Let $A(\theta^*)$ be based on a sequential median-find algorithm. Algorithm $A(\theta^*)$ runs in $O(n)$ time, but each comparison s evaluates the truth value of inequality $g_s(\theta^*) \geq 0$ by computing the sign of $f(\theta_s)$. The sign of $f(\theta_s)$ can be found in $O(n)$ time, so $A(\theta^*)$ runs in $O(n^2)$ time, even though the exact value of θ^* is unknown until the end of the computation.

IMPROVEMENTS USING PARALLELISM

We can decrease the time complexity of the algorithm by replacing the usual median-find procedure with a sequentialized version of a parallel algorithm. It is possible to devise a median-find procedure that uses $O(n/\log n)$ processors, completes in $O(\log n)$ time, and can be simulated in $O(n)$ sequential time. (Note that there are algorithms with better bounds that cannot be simulated in $O(n)$ sequential time.)

The advantage of a parallel algorithm is that the comparisons on a particular time step can be evaluated in an arbitrary order. Let

$$g_{s_1}(\theta^*) \geq 0, g_{s_2}(\theta^*) \geq 0, \dots, g_{s_{n/\log n}}(\theta^*) \geq 0$$

be the comparisons on time step j . Rather than evaluating each of them by computing $f(\theta_{s_i})$, $1 \leq i \leq n/\log n$, we evaluate the one with median θ_s value. (This is where we need to order the θ_s values.) This comparison can be used to infer the truth value of half of the remaining comparisons. That is, we evaluate the comparisons by performing a binary search for θ^* among the θ_{s_i} values.

The time complexity of the new algorithm is as follows. A total of $O(n)$ comparisons must be evaluated, organized so that $O(n/\log n)$ comparisons are made per time step for a duration of $O(\log n)$ time steps. During each time step, binary search resolves $O(\log n)$ comparisons by actually computing the sign of $f(\theta_s)$, and the rest of the comparisons are decided by transitivity. There are consequently $O(n \log n)$ operations per time step, multiplied by $O(\log n)$ time steps, giving a total of $O(n \log^2 n)$ operations.

FURTHER IMPROVEMENTS

This problem can be attacked with the related “prune-and-search” technique. If the proper comparisons are done, it is possible to reduce the size of the original problem and solve a substantially-smaller subproblem. The resulting time complexity is $O(n)$.

43.3 EXAMPLE 2: SELECTING VERTICES IN ARRANGEMENTS

GLOSSARY

Selection problem: Given a totally ordered set S and an integer k , $1 \leq k \leq |S|$, the selection problem is to find θ^* , the k th smallest item in S .

Ranking problem: Given a totally ordered set S and a number θ , the ranking problem is to return the number of items $\text{rank}(\theta, S)$ in S whose value is less than or equal to θ .

Arrangement: The subdivision of space induced by a set of hyperplanes. (See Chapter 24.)

Permutation: A sequence of n distinct integers in the range 1 through n .

Inversion: A pair (i, j) occurring in a permutation where $i < j$ but j precedes i in the permuted sequence.

PROBLEM STATEMENT

- Input: Set $H = \{h_1, \dots, h_n\}$ of lines in the plane, where h_i has equation $y = m_i x + b_i$, and the lines are indexed in order of increasing slope. Integer k , $1 \leq k \leq \binom{n}{2}$.
- Output: Let V be the intersection points (vertices) of the arrangement formed by H . The output is the vertex v^* whose x -coordinate has rank k among the x -coordinates in V .

We assume that m_i and b_i are rational, and that H is in general position, so that no three lines intersect in a single vertex, no two vertices have the same x -coordinate, no line is vertical, and no two lines are parallel.

CHOICE OF MONOTONIC FUNCTION

Consider the function $f(\theta) = \text{rank}(\theta, V) - k$. This function is monotonically non-decreasing in θ , and it has the property that θ^* , the x -coordinate of v^* , satisfies $\theta^* = \sup\{\theta \mid f(\theta) = 0\}$.

FIXED-VALUE EVALUATION

Evaluating $f(\theta) = \text{rank}(\theta, V)$ amounts to counting the number of vertices in V whose x -coordinates are less than input θ . This can be done in the following way. The y -intercepts of the intersections of H with the line $x = \theta$ are the numbers $m_i\theta + b_i$, $1 \leq i \leq n$. If these numbers are sorted in decreasing order and value $m_i\theta + b_i$ is replaced by index i , the result is a permutation $\pi(\theta)$. The key insight is that the number of inversions in $\pi(\theta)$ equals $\text{rank}(\theta, V)$.

Algorithm $A(\theta)$, the algorithm to determine $f(\theta)$, consists of:

1. Computing the permutation $\pi(\theta)$.
2. Counting the number of inversions in $\pi(\theta)$.
3. Subtracting k from this result.

The first step is essentially a sorting step, which can be done sequentially in $O(n \log n)$ time and in parallel in $O(\log n)$ time with $O(n)$ processors. The second step can be done by a mergesort-like procedure.

THE DECISION-TREE ALGORITHM

The first step of algorithm $A(\theta)$ depends on the value of θ . Once the permutation $\pi(\theta)$ is computed, the control flow of the second and third steps does not depend on θ .

The comparisons $s(i, j)$ in $A(\theta)$ ask whether i precedes j in the permutation: Is $m_i\theta + b_i \geq m_j\theta + b_j$? We rewrite this inequality as

$$g_{s(i,j)}(\theta) = (m_i - m_j)\theta + (b_i - b_j) \geq 0.$$

It is clear that $g_{s(i,j)}(\theta)$ has a root $\theta_{s(i,j)}$ at $\frac{b_j - b_i}{m_i - m_j}$ (recall that no two lines have the same slope). The sign of $m_i - m_j$ is negative, implying that the functions $g_{s(i,j)}(\theta)$ are monotonically nonincreasing. The root $\theta_{s(i,j)}$ is rational, so evaluating the sign of $f(\theta_{s(i,j)})$ or comparing $\theta_{s(i,j)}$ values poses no difficulty.

EVALUATING $f(\theta^*)$

Suppose we attempt to evaluate $f(\theta^*)$ at the unknown x -coordinate θ^* . The chief difficulty is resolving comparisons involving θ^* . These comparisons correspond to inequalities of the form $g_{s(i,j)}(\theta^*) \geq 0$.

The inequality $g_{s(i,j)}(\theta^*) \geq 0$ is the same as the inequality $\theta^* \geq \theta_{s(i,j)}$. We can determine the truth value of this inequality by evaluating $f(\theta_{s(i,j)})$. Because $f(\theta)$ is monotonic, $f(\theta_{s(i,j)}) < 0$ implies that $\theta_{s(i,j)} < \theta^*$, and $f(\theta_{s(i,j)}) > 0$ implies that $\theta_{s(i,j)} > \theta^*$. Otherwise, $f(\theta_{s(i,j)}) = 0$, and $\theta^* = \theta_{s(i,j)}$.

A sequential implementation of algorithm $A(\theta^*)$ evaluates $O(n \log n)$ comparisons. Each comparison at node $s(i, j)$ determines the sign of $f(\theta_{s(i,j)})$, an operation that takes $O(n \log n)$ time. Step one therefore takes $O(n^2 \log^2 n)$ time to simulate. The rest of the work, steps two and three, takes additional $O(n \log n)$ time steps. The total work is $O(n^2 \log^2 n)$.

IMPROVEMENTS USING PARALLELISM

There are efficient parallel sorting algorithms; it is possible to sort n numbers in $O(\log n)$ time using n processors. If we perform a binary search on the n comparisons per level, only $O(\log n)$ $f(\theta)$ -evaluations are done, and the remaining comparisons are resolved by transitivity. The work per level is $O(n \log^2 n)$. There are $O(\log n)$ levels, so the time complexity of this algorithm is $O(n \log^3 n)$.

FURTHER IMPROVEMENTS

Cole [Col87b] gave a general technique that can be used to remove a log factor from the time complexity. If a parallel algorithm can be described by a circuit with constant fan-out gates (say fan-out two), then the following trick can be applied. Suppose that $\frac{c-1}{c}$ of the comparisons on the first time step have been resolved; then the inputs of at least $\frac{c-2}{c}$ of the comparisons on the second time step are available, and these comparisons are also ready to be resolved. Cole's idea is to combine these newly-ready comparisons with the unresolved comparisons. The total number of comparisons that need to be resolved by actually evaluating $f(\theta)$ becomes $O(\log P(n) + T(n))$. With respect to the sorting problem, the parallel sorting algorithm can be written as a circuit with fan-out two, so a total of $O(\log n)$ function evaluations need to be performed.

A second log factor can be removed by approximate ranking. Rather than computing the number of inversions exactly, the number is approximated. This approximation is sufficiently precise to determine the relative values of θ_s and θ^* . The resulting time complexity is $O(n \log n)$.

It has recently been established experimentally [OV02] that, under realistic assumptions about the input, Cole's improvement may be unnecessary (here and elsewhere): QuickSort is superior to parallel sorting in many practical situations. For example, if the roots being sorted are uniformly distributed over the comparison batches, then QuickSort is provably better. Although this assumption is often unwarranted, it seems to hold in many situations, as evidenced by successful application to the Fréchet-distance algorithm of Alt and Godau [AG95].

43.4 EXAMPLE 3: SELECTING INTERDISTANCES

GLOSSARY

L_p interdistance: Given points $a = (a_1, a_2, \dots, a_d)$ and $b = (b_1, b_2, \dots, b_d)$, $1 \leq p < \infty$, the L_p interdistance between a and b is given by

$$\|a - b\|_p = \left(\sum_{i=1}^d |a_i - b_i|^p \right)^{1/p}.$$

L_∞ interdistance: Given points a and b as above, the L_∞ interdistance between a and b is given by

$$\|a - b\|_\infty = \max_{1 \leq i \leq d} \{|a_i - b_i|\}.$$

$\tilde{O}(f(n))$: The set of functions that are $O(f(n)^{1+\epsilon})$, for any $\epsilon > 0$.

PROBLEM STATEMENT

- Input: Set P of n points in the plane. Integer k , $1 \leq k \leq \binom{n}{2}$.
- Output: Let D be the L_p interdistances formed by the points in P . The output is the interdistance θ^* with rank k in D .

We assume that all interdistances are unique.

CHOICE OF MONOTONIC FUNCTION

As in the vertex selection problem, the function $f(\theta) = \text{rank}(\theta, D) - k$.

FIXED-VALUE EVALUATION

The ranking problem in either metric can be viewed as a problem involving balls and points. Place a ball of radius θ around each point in P ; then $\text{rank}(\theta, D)$ is one-half times the number of point-ball containments. Do not include the center point-ball containments in this total. In the L_∞ metric, the unit ball is a square, and in the L_2 metric, the unit ball is a circle.

We deal with the L_∞ problem first. Ranking can be done efficiently by merging the x -coordinates of the vertical box sides of radius θ with the x -coordinates $\{x_1, \dots, x_n\}$ of P , and then repeating this process with the y -coordinates. (Assume that x_1, \dots, x_n are presorted.) Given these sorted orders, we can simulate a sweep-line algorithm that counts the number of point-square containments.

The L_2 ranking problem is somewhat harder, but the basic strategy is identical to the L_∞ case. To rank θ , we form an arrangement of circles, each circle of radius θ and centered about a distinct point in P . Assume that this arrangement can be

built and preprocessed for planar point location, and assume that each region of the arrangement is labeled with the number of circles that contain it. For each point in P , perform a point location query to determine how many circles contain it.

Suppose there are s circles and t points. The arrangement can be built in $O(s^2)$ time, and each point location query can be answered in $O(\log s)$ time. The total processing time is $O(s^2 + t \log s)$.

Our ranking problem consists of n circles and points. If we divide the set of circles into $O(\sqrt{n})$ groups of size $O(\sqrt{n})$ and perform the procedure above, ranking can be performed in $\tilde{O}(n^{3/2})$ time.

THE DECISION-TREE ALGORITHM

The first step in the L_∞ ranking algorithm is to sort the values $\{x_1, \dots, x_n, x_1 - \theta, \dots, x_n - \theta\}$ and to sort the analogous y -coordinates. Some of these comparisons $s(i, j)$ depend on θ ; they are of the form $x_i \geq x_j - \theta$. This implies that $g_{s(i,j)}(\theta) = \theta + x_i - x_j$, and the root of $g_{s(i,j)}(\theta)$ is $\theta_{s(i,j)} = x_j - x_i$. After these two sorted orders are known, the remainder of the algorithm does not depend on θ .

The L_2 algorithm is more complicated. The construction of the circular arrangement contains some steps that depend on θ . A typical such step $s(z, C)$ involves the comparison of a point with a circle: Does point $z = (z_1, z_2)$ lie inside circle C ? Let the center of circle C be (c_1, c_2) . Deciding if z lies on or inside circle C of radius θ is equivalent to determining the truth value of the inequality $(z_1 - c_1)^2 + (z_2 - c_2)^2 \leq \theta^2$, so $g_{s(z,C)}(\theta) = \theta^2 - (z_1 - c_1)^2 - (z_2 - c_2)^2$. Function $g_s(z, C)$ has roots at $\pm\theta_{s(z,C)} = \pm\sqrt{(z_1 - c_1)^2 + (z_2 - c_2)^2}$.

EVALUATING $f(\theta^*)$

As in vertex selection, we perform interdistance selection by ranking unknown interdistance θ^* . For the L_∞ problem, the only step that needs the value of θ^* is the merging step; here, comparisons of the form $x_i \geq x_j - \theta^*$ must be resolved. This comparison is precisely $\theta_{s(i,j)} \leq \theta^*$, which we can resolve by evaluating $f(\theta_{s(i,j)})$.

The cost of presorting the data is $O(n \log n)$, and there are $O(n)$ comparisons in the merging steps, each comparison taking $O(n)$ time. Parametric search takes $O(n \log n + n^2) = O(n^2)$ time.

For the L_2 problem, comparisons of the form $(z_1 - c_1)^2 + (z_2 - c_2)^2 \leq (\theta^*)^2$ must be resolved. This comparison is precisely $(\theta_{s(z,C)})^2 \leq (\theta^*)^2$. Since $f(-\theta_{s(z,C)}) = -k$, this comparison can be resolved by evaluating the sign of $f(+\theta_{s(z,C)})$. Note that the square root is not needed in this evaluation because θ is squared in the functions $g_{s(z,C)}$.

The description of the L_2 ranking problem included an analysis of its time complexity. The ranking algorithm makes $\tilde{O}(n^{3/2})$ comparisons, each taking $\tilde{O}(n^{3/2})$ time, for a total of $\tilde{O}(n^3)$ time.

IMPROVEMENTS USING PARALLELISM

In the L_∞ algorithm, only the merging step needs to be parallelized. This can be done in $O(\log n)$ time using $O(n/\log n)$ processors. A straightforward application

of parametric search gives an $O(n \log^3 n)$ time algorithm.

With respect to the L_2 algorithm, it is possible to devise a parallel algorithm that uses $O(n^{3/2})$ processors and $O(\log n)$ time. Consequently, only $O(\log^2 n)$ comparisons need to be resolved by ranking. The total time is only $\tilde{O}(n^{3/2})$.

FURTHER IMPROVEMENTS

Cole's trick removes one log factor from the L_∞ algorithm, giving an $O(n \log^2 n)$ time algorithm. A different ranking scheme, one based on epsilon nets (Sections 31.2 and 34.4), is used to obtain better ranking results for the L_2 problem. The resulting time complexity is $\tilde{O}(n^{4/3})$.

43.5 EXAMPLE 4: RAY SHOOTING

GLOSSARY

Ray shooting: Determining the first object intersected by a ray ([Chapter 37](#)).

Partition tree: A data structure for simplex range queries (Section 31.2).

PROBLEM STATEMENT

- Input: A set H of n hyperplanes in k -dimensional space. A query ray ρ with origin o .
- Output: The first hyperplane of H that ρ intersects.

It is intended that the queries be repeated many times, so we want a data structure with small query time. We assume that o is not contained in a hyperplane of h and that $\rho \cap H \neq \emptyset$.

CHOICE OF MONOTONIC FUNCTION

Let ray ρ be given by its origin o and an arbitrary point $o(1)$ on ρ . For nonnegative θ , let $\rho(\theta)$ be the open subsegment of ρ given by $(1 - \lambda)o + \lambda o(1)$, $0 < \lambda < \theta$. (We will call the nonorigin endpoint $o(\theta)$). Let

$$f(\theta) = \begin{cases} [|\{h \in H \mid \rho(\theta) \cap h \neq \emptyset\}| \geq 1], & \theta \geq 0 \\ 0, & \theta < 0. \end{cases}$$

Here, $[P(x)] = 1$ if predicate $P(x)$ is true and 0 if $P(x)$ is false. The set $\{h \in H \mid \rho(\theta) \cap h \neq \emptyset\}$ consists of the hyperplanes in H that $\rho(\theta)$ intersects.

FIXED-VALUE EVALUATION

We now address the issue of efficiently computing $f(\theta)$. A reasonable data structure for such a task is a partition tree, described in [Chapter 36](#).

Let X be a set of points. Each node r in the partition tree corresponds to a set $X(r) \subseteq X$; the root corresponds to X . Furthermore, each node r is associated with a region of space $J(r)$, usually a simplex.

The partition tree can be used to compute $f(\theta)$ by determining whether the endpoints of $\rho(\theta)$, o and $o(\theta)$, lie in the same cell of arrangement $\mathcal{A}(H)$. To do this, the hyperplanes in H are dualized to a set of points $\mathcal{D}(H)$, and a partition tree is constructed for $\mathcal{D}(H)$. Points o and $o(\theta)$ lie in the same cell of $\mathcal{A}(H)$ if the double-wedge, the dual of the segment connecting o and $o(\theta)$, does not contain any points of $\mathcal{D}(H)$.

THE DECISION-TREE ALGORITHM

The basic step of the algorithm above compares the position of hyperplane $\mathcal{D}(o(\theta))$ to a point p , one of the vertices of $J(r)$. This is tantamount to deciding whether point $o(\theta)$ and o are on the same side of a particular hyperplane $h = \mathcal{D}(p)$.

Let h be given by $n_h \cdot x = \alpha_h$, where n_h is the unit normal of h in the direction of o . Then $o(\theta)$ and o are on the same side of h if $g_h(\theta) = n_h \cdot o(\theta) - \alpha_h \geq 0$.

The function $g_h(\theta)$ has a single root at

$$\theta_h = \frac{\alpha_h - (n_h \cdot o)}{n \cdot (o(1) - o)}.$$

The sign of $f(\theta_h)$ can be evaluated when the components of n and $o(1)$ are rational.

EVALUATING $f(\theta^*)$

Given ray ρ , we seek the value of $o(\theta^*)$, the location of the first intersection of ρ with a hyperplane in h . The number of points inside the double-wedge for $\rho(\theta^*)$ can be computed by resolving comparisons of the form

$$g_h(\theta^*) = n_h \cdot \rho(\theta^*) - \alpha_h \geq 0 = g_h(\theta_h).$$

This comparison is the same as determining the truth value of $\theta^* \geq \theta_h$, which is decided by evaluating the sign of $f(\theta_h)$. As stated above, the partition tree is used to evaluate $f(\theta_h)$.

Let $B(n)$ be the cost of constructing a partition tree on H , let $C(n)$ be the amount of storage needed, and let $Q(n)$ be the cost of querying the partition tree. Preprocessing does not depend on θ , and it takes $B(n)$ time. After preprocessing, the number of operations necessary to evaluate each $f(\theta)$ is $Q(n)$, and there are $Q(n)$ such evaluations in computing θ^* . The total number of operations in the parametric search is $O(B(n) + Q(n)^2)$.

PARALLEL ALGORITHM

Suppose that the query algorithm can be parallelized so that it runs in $T(n)$ time on $P(n)$ processors. In this case, $B(n)$ time is spent preprocessing the data structure, and $T(n)P(n)$ comparisons are made. Of these comparisons, $T(n) \log P(n)$ are resolved by computing the sign of $f(\theta_h)$, and the rest are resolved by transitivity. This version of the parametric ray-shooting algorithm takes $D(n) = O(Q(n)T(n) \log P(n) + T(n)P(n) + B(n))$ time.

Using results on partition trees, one can construct a family of parametric search algorithms parameterized by m , $n \leq m \leq n^d$, whose preprocessing, storage, and query requirements are $B(n) = \tilde{O}(m)$, $C(n) = \tilde{O}(m)$, and $D(n) = \tilde{O}(\frac{n}{m^{1/d}})$, respectively.

43.6 OTHER RESULTS

We summarize in Table 43.6.1 some of the results obtained with the parametric search technique on computational geometry problems. Parametric search has been successfully applied in other domains as well.

TABLE 43.6.1 Selected parametric search results.

PROBLEM NAME	INPUT	COMPLEXITY	SOURCE
3-dim set diameter	n points in 3-dim	$O(n \log^3 n)$	[BCM93]
Minimum-width annulus	n points in plane	$\tilde{O}(n^{8/5})$	[AST94]
Collision btw two polyhedra	two polyhedra, n vertices total	$\tilde{O}(n^{8/5})$	[ST95]
Biggest stick	n -sided simple polygon	$\tilde{O}(n^{8/5})$	[AST94]
L_p interdistance selection	n points in plane, k	$\tilde{O}(n^{4/3})$	[AASS93]
L_∞ interdistance selection	n points in plane, k	$O(n \log^2 n)$	[Sal89]
Min Hausdorff dist btw polygons	n - and m -sided simple poly	$\tilde{O}((mn)^2)$	[AST94]
2-center	n points in plane	$O(n^2 \log n)$	[JK94]
Center point in plane	n points in plane	$O(n)$	[JM94]
Segment center in plane	n segments in plane	$\tilde{O}(n)$	[ES96]
Selecting verts in arrangements	n lines in plane, k	$O(n \log n)$	[CSSS89]

Parametric search is not limited to monotonic functions of a single parameter—there is also a multidimensional version of parametric search. An instance of its application in computational geometry appears in Matoušek [Mat93].

43.7 SOURCES AND RELATED MATERIAL

FURTHER READING

The example in Section 43.2 was drawn from Cole [Col87a], where it was used as the basis of a multidimensional partitioning algorithm. Megiddo [Meg85] discovered the linear-time algorithm for quartering the plane. The example in Section 43.3, usually known as *slope selection*, is from Cole et al. [CSSS89]. The interdistance examples in Section 43.4 are from Agarwal et al. [AASS93] and Salowe [Sal89]. Agarwal et al. discovered the L_2 algorithm, and Salowe described the L_∞ algorithm. Finally, the ray-shooting example in Section 43.5 is from Agarwal and Matoušek [AM93].

A good bibliography of parametric search in computational geometry appears in Agarwal et al. [AST94].

Parametric searching can be viewed as one among several techniques for geometric optimization. Agarwal and Sharir review the shortcomings of parametric search, and survey the alternatives in [AS98], including randomization ([Chapter 40](#)), expander graphs, cuttings, matrix searching, and the prune-and-search technique, which has been applied so successfully to linear programming ([Chapter 45](#)).

RELATED CHAPTERS

- [Chapter 34: Point location](#)
- [Chapter 36: Range searching](#)
- [Chapter 45: Linear programming](#)

REFERENCES

- [AASS93] P.K. Agarwal, B. Aronov, M. Sharir, and S. Suri. Selecting distances in the plane. *Algorithmica*, 9:495–514, 1993.
- [AG95] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5:75–91, 1995.
- [AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22:794–806, 1993.
- [AS98] P.K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30:412–458, 1998.
- [AST94] P.K. Agarwal, M. Sharir, and S. Toledo. Applications of parametric searching in geometric optimization. *J. Algorithms*, 17:292–318, 1994.
- [BCM93] H. Brönnimann, B. Chazelle, and J. Matoušek. Product range spaces, sensitive sampling, and derandomization. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 400–409, 1993.
- [Col87a] R. Cole. Partitioning point sets in arbitrary dimensions. *Theoret. Comput. Sci.*, 49:239–265, 1987.

- [Col87b] R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. Assoc. Comput. Mach.*, 34:200–208, 1987.
- [CSSS89] R. Cole, J. Salowe, W. Steiger, and E. Szemerédi. An optimal-time algorithm for slope selection. *SIAM J. Comput.*, 18:792–810, 1989.
- [ES96] A. Efrat and M. Sharir. A near-linear algorithm for the planar segment center problem. *Discrete Comput. Geom.*, 16:239–258, 1996.
- [JK94] J.W. Jaromczyk and M. Kowaluk. An efficient algorithm for the Euclidean two-center problem. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 303–311, 1994.
- [JM94] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discrete Comput. Geom.*, 12:291–312, 1994.
- [Mat93] J. Matoušek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.
- [Meg79] N. Megiddo. Combinatorial optimization with rational objective functions. *Math. Oper. Res.*, 4:414–424, 1979.
- [Meg83] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. Assoc. Comput. Mach.*, 30:852–865, 1983.
- [Meg85] N. Megiddo. Partitioning with two lines in the plane. *J. Algorithms*, 6:430–433, 1985.
- [OV02] R. Oostrum and R.C. Veltkamp. Parametric search made practical. In *Proc. 18th Annu. Sympos. Comput. Geom.*, pages 1–9, 2002.
- [Sal89] J. Salowe. L_∞ interdistance selection by parametric search. *Inform. Process. Lett.*, 30:9–14, 1989.
- [ST95] E. Schömer and C. Thiel. Efficient collision detection for moving polyhedra. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 51–60, 1995.

44 THE DISCREPANCY METHOD IN COMPUTATIONAL GEOMETRY

Bernard Chazelle

INTRODUCTION

Discrepancy theory investigates how uniform nonrandom structures can be. For example, given n points in the plane, how should we color them red and blue so as to minimize the difference between the number of red points and the number of blue ones within any disk? Or, how should we place n points in the unit square so that the number of points that lie within any given triangle in the square is as close as possible to n times the area of the triangle? Questions of this nature have direct relevance to computational geometry for two reasons. One of them is their close association with the problem of derandomizing probabilistic algorithms. Such algorithms are often based on random sampling, and discrepancy theory provides tools for carrying out the sampling deterministically. This has led to the intriguing fact that virtually all of the important problems in low-dimensional computational geometry can be solved as efficiently deterministically as probabilistically. The second application of discrepancy theory to computational geometry is in the area of lower bounds for multidimensional searching. The complexity of these problems is often tied to spectral properties of geometric set systems, which themselves lie at the heart of geometric discrepancy theory.

44.1 VC-DIMENSION THEORY

GLOSSARY

Set system: A pair $\Sigma = (X, \mathcal{R})$, where X is a set and \mathcal{R} is a collection of subsets of X , is called a set system. The term **geometric set system** refers to the case where $X \subset \mathbb{R}^d$ and each $R \in \mathcal{R}$ is of the form $X \cap f(C)$, where C is a fixed region of \mathbb{R}^d (e.g., a simplex) and f is any member of a fixed group F of transformations (e.g., a rotation).

VC-dimension: Given $Y \subseteq X$, the set system *induced* by Y is of the form $(Y, \mathcal{R}|_Y)$, where $\mathcal{R}|_Y = \{Y \cap R \mid R \in \mathcal{R}\}$. The **VC-dimension** of Σ is the maximum size of any Y such that $\mathcal{R}|_Y = 2^Y$. For example, the VC-dimension of the infinite set system formed by points in \mathbb{R}^2 and halfplanes is 3. The **shatter function** $\pi_{\mathcal{R}}(m)$ of a (usually infinite) set system $\Sigma = (X, \mathcal{R})$ is the maximum number of subsets in the set system $(Y, \mathcal{R}|_Y)$ induced by any $Y \subseteq X$ of size m . If $\pi_{\mathcal{R}}(m)$ is bounded by cm^d for some constants $c, d > 0$, then the set system is said to have a **shatter function exponent** of at most d .

Dual set system: The set system $\Sigma^* = (X^*, \mathcal{R}^*)$, where $X^* = \mathcal{R}$, $\mathcal{R}^* = \{R_x \mid x \in X\}$, and $R_x = \{R \in \mathcal{R} \mid x \in R\}$, is called the **dual set system** of Σ .

The shatter function of Σ^* is called the *dual shatter function* of Σ .

Discrepancy: The *incidence matrix* A of a finite set system $\Sigma = (X, \mathcal{R})$ is the matrix whose $|X|$ columns (resp. $|\mathcal{R}|$ rows) are indexed by the elements of X (resp. \mathcal{R}): A_{ij} is 1 if the i th set of \mathcal{R} contains the j th element of X , and 0 otherwise. The (red-blue) *discrepancy* of Σ is $\min_{x \in \{-1, 1\}^{|X|}} \|Ax\|_\infty$.

The concept of VC-dimension was introduced by Vapnik and Chervonenkis [VC71]. The relation between VC-dimension and shatter function is a key component of the theory.

LEMMA 44.1.1 [VC71, Sau72, She72]

If the shatter function exponent is $O(1)$, then so is the VC-dimension. Conversely, if the VC-dimension is $d \geq 1$ then, for any $m \geq d$, $\pi_{\mathcal{R}(m)} < (em/d)^d$.

LEMMA 44.1.2 [Ass83]

If a set system has VC-dimension d , then its dual has VC-dimension less than 2^{d+1} .

Any set system of n elements and n sets has discrepancy $O(\sqrt{n})$, and this bound is sometimes tight. If the VC-dimension is bounded, the discrepancy falls below the \sqrt{n} barrier. The bounds below are stated in terms of the shatter function exponent. In view of Lemma 44.1.1, we can replace the exponent by the VC-dimension if we wish. Matoušek, Welzl, and Wernisch [MWW93] established a bound of $O(n^{1/2 - 1/(2d)} (\log n)^{1 + 1/(2d)})$ on the discrepancy of set systems with shatter function exponent d . This was improved to $O(n^{1/2 - 1/(2d)})$ by Matoušek:

THEOREM 44.1.3 [Mat95b]

The discrepancy of a set system of n elements with shatter function exponent $d > 1$ is $O(n^{1/2 - 1/(2d)})$, which is optimal for $d \geq 2$.

Similar bounds can be obtained in terms of the dual shatter function. Matoušek, Welzl, and Wernisch proved a bound of $O(n^{1/2 - 1/(2d)} \sqrt{\log n})$ on the discrepancy of set systems with dual shatter function exponent d . It is surprising that an extra $\sqrt{\log n}$ should be needed. Optimality was shown by Matoušek for the cases $d = 2, 3$, and by Alon, Rónyai, and Szabó for $d > 3$.

THEOREM 44.1.4 [MWW93, Mat97, ARS99]

The discrepancy of a set system of n elements with dual shatter function exponent $d > 1$ is $O(n^{1/2 - 1/(2d)} \sqrt{\log n})$, which is optimal for $d \geq 2$.

44.2 SAMPLING IN BOUNDED VC-DIMENSION

GLOSSARY

ϵ -Net: Given a finite set system (X, \mathcal{R}) and any $0 < \epsilon < 1$, a set $N \subseteq X$ is called an ϵ -net for (X, \mathcal{R}) if $N \cap R \neq \emptyset$ for any $R \in \mathcal{R}$ with $|R|/|X| > \epsilon$.

ϵ -Approximation: Given a finite set system (X, \mathcal{R}) and any $0 < \epsilon < 1$, a set $A \subseteq X$ is called an ϵ -approximation for (X, \mathcal{R}) if, for any $R \in \mathcal{R}$,

$$\left| \frac{|R|}{|X|} - \frac{|A \cap R|}{|A|} \right| \leq \epsilon.$$

Product set system: Given two finite set systems $\Sigma_1 = (X_1, \mathcal{R}_1)$ and $\Sigma_2 = (X_2, \mathcal{R}_2)$, the *product set system* $\Sigma_1 \otimes \Sigma_2$ is defined as $(X_1 \times X_2, \mathcal{T})$, where \mathcal{T} consists of all subsets $T \subseteq X_1 \times X_2$ such that each set of the form $T_{x_2}^1 = \{x \in X_1 \mid (x, x_2) \in T\}$ belongs to \mathcal{R}_1 and, similarly, $T_{x_1}^2 = \{x \in X_2 \mid (x_1, x) \in T\}$ belongs to \mathcal{R}_2 .

To sample a set system Σ is to extract a (small) subset of the elements whose intersection with any set R of Σ is a good predictor of the size of R . This is the idea behind an ϵ -approximation. A weaker version of sampling, the ϵ -net, requires only that large enough sets R be intersected by the sample. The key result in VC-dimension theory is that if Σ has bounded VC-dimension, then for any given level of accuracy, the sample size need not depend on the size of the set system. This is rather counterintuitive. It says, for example, that if we want to estimate how many people live within 1 mile of a post office and, to go about it, we opt to pick a sample of the population, we should simply solve the problem for the sample, and then scale up the answer appropriately; the same sample size will work just as well whether the country is France or India!

LEMMA 44.2.1

Let X_1, X_2 be disjoint subsets of X of the same size, and let A_i be an ϵ -approximation for the subsystem induced by X_i . If $|A_1| = |A_2|$, then $A_1 \cup A_2$ is an ϵ -approximation for the subsystem induced by $X_1 \cup X_2$.

LEMMA 44.2.2

If A is an ϵ -approximation for (X, \mathcal{R}) , then any ϵ' -approximation (resp. -net) for $(A, \mathcal{R}|_A)$ is also an $(\epsilon + \epsilon')$ -approximation (resp. -net) for (X, \mathcal{R}) .

A greedy approach to sampling yields an effective algorithm for arbitrary set systems. Writing $\epsilon = 1/r$, choose some $1 \leq r \leq n$. First, remove all sets $R \in \mathcal{R}$ of size at most n/r . Second, initialize the set N to \emptyset . Next, find the element $x \in X$ that belongs to the most sets of \mathcal{R} (in case of a tie, any one will do) and add it to N . Remove from \mathcal{R} every set that contains x , discard x , and iterate in this fashion until \mathcal{R} is empty. An elementary analysis shows that this produces a $(1/r)$ -net for (X, \mathcal{R}) of size $O(r \log |\mathcal{R}|)$. This was proven independently by Johnson [Joh74] and Lovász [Lov75]. A slightly more complicated “weighted” version of the greedy algorithm, due to Chazelle [Cha00], gives an analogous result for $(1/r)$ -approximations.

THEOREM 44.2.3

Given a set system (X, \mathcal{R}) , where $|X| = n$ and $|\mathcal{R}| = m$, for any $1 \leq r \leq n$, it is possible to find, in time $O(nm)$, a $(1/r)$ -net for (X, \mathcal{R}) of size $O(r \log m)$ and a $(1/r)$ -approximation for (X, \mathcal{R}) of size $O(r^2 \log m)$.

The size of the sample depends (albeit weakly) on the size of the set system. In the presence of bounded VC-dimension, however, this dependency magically disappears. Again, we will base our results not on the VC-dimension but on the shatter function exponent d (but the same results hold if d denotes the VC-dimension). Geometric set systems often are defined implicitly and are accessible via an *oracle* function that takes any $Y \subseteq X$ as input and returns the list of sets in $\mathcal{R}|_Y$

(each set represented explicitly). We assume that the time to complete this task is $O(|Y|^{d+1})$, which is linear in the maximum possible size of the oracle's output. The existence of such an oracle is quite realistic: For example, in the case of points and disks in the plane, we have $d = 3$, and so this assumes that, given n points, we can enumerate all subsets enclosed by a disk in time $O(n^4)$. To do this, enumerate all k -tuples of points ($k \leq 3$) and, for each tuple, find which points lie inside the smallest disk enclosing the k points.

THEOREM 44.2.4

Given a set system (X, \mathcal{R}) of shatter function exponent d , for any $r \geq 2$, a $(1/r)$ -approximation for (X, \mathcal{R}) of size $O(dr^2 \log dr)$ and a $(1/r)$ -net for (X, \mathcal{R}) of size $O(dr \log dr)$ can be computed in time $O(d)^{3d}(r^2 \log dr)^d |X|$.

A randomized construction of ϵ -approximations in bounded VC-dimension was given by Vapnik and Chervonenkis [VC71]. The deterministic construction cited above is due to Chazelle and Matoušek [CM96]. Earlier influential work can be found in [CF90, Mat90, Mat91, Mat95a]. The bound on the size of ϵ -nets was established by Haussler and Welzl [HW87]. The running time for computing a $(1/r)$ -net was improved to $O(d)^{3d}(r \log dr)^d |X|$ by Brönnimann, Chazelle, and Matoušek [BCM99], using the concept of a *sensitive ϵ -approximation*.

For fixed d , Komlós, Pach, and Woeginger [KPW92] showed that the bound of $O(r \log r)$ for $(1/r)$ -nets cannot be improved in general (see a nice discussion in [PA95]). The situation is different with ϵ -approximations, however, for which Theorems 44.1.3 and 44.1.4 can be put to use. Matoušek, Welzl, and Wernisch proved the following:

THEOREM 44.2.5 [MWW93]

Let (X, \mathcal{R}) be a set system of VC-dimension $d > 1$. There exists a $(1/r)$ -approximation for (X, \mathcal{R}) of size $O(r^{2-2/(d+1)}(\log r)^{2-1/(d+1)})$, for any $r \geq 2$.

The log factor can be removed by appealing to Theorem 44.1.3.

THEOREM 44.2.6 [MWW93]

Let (X, \mathcal{R}) be a set system with dual shatter function exponent $d > 1$. There exists a $(1/r)$ -approximation for (X, \mathcal{R}) of size $O(r^{2-2/(d+1)}(\log r)^{1-1/(d+1)})$, for any $r \geq 2$.

Given n lines in the plane, we can use an ϵ -approximation to estimate how many lines cut through an arbitrary line segment. Suppose that, instead, we wish to estimate the number of vertices in the induced arrangement that fall within an arbitrary triangle. Product set systems allow us to do that. Let Σ_1 be the set system induced by n blue lines in the plane and the set of all line segments: a set of the system is the subset of blue lines intersected by a given segment. We define Σ_2 similarly with n red lines. The product $\Sigma_1 \otimes \Sigma_2$ is a set system (Z, \mathcal{T}) , where Z is the set of red-blue vertices of the induced arrangement (assuming general position). A set of $\Sigma_1 \otimes \Sigma_2$ is any subset T of Z such that, along any (blue or red) line ℓ , the vertices of T incident to ℓ (if any) appear consecutively among the red-blue vertices of ℓ . This suggests we can use ϵ -approximations to, say, estimate how many red-blue vertices fall in an arbitrary triangle, or even in an arbitrary convex region. One must be careful, however. The product of two set systems with bounded VC-dimension might not itself have bounded VC-dimension. Indeed, any bichromatic

matching of the lines gives a collection of n vertices, any of whose 2^n subsets is a valid set of \mathcal{T} . Although the product does not have bounded VC-dimension, it so happens that sampling in it is still possible: that is the beauty of product set systems.

LEMMA 44.2.7

Given any $0 \leq \epsilon_i \leq 1$, let A_i be an ϵ_i -approximation for a set system Σ_i , for $i = 1, 2$. Then the Cartesian product $A_1 \times A_2$ is an $(\epsilon_1 + \epsilon_2)$ -approximation for $\Sigma_1 \otimes \Sigma_2$.

The product operation is associative, and the theorem can be extended to multiple products of set systems. The notion of product set system was introduced by Brönnimann, Chazelle, and Matoušek, who also proved:

LEMMA 44.2.8 [BCM99]

Given an ϵ -approximation A for a set system Σ , the d -fold Cartesian product $A \times \cdots \times A$ is a $(d\epsilon)$ -approximation for the d -fold product $\Sigma \otimes \cdots \otimes \Sigma$.

One of the most important applications of the theorem above is to counting vertices in an arrangement of hyperplanes in \mathbb{R}^d . We consider the set system $\Sigma = (H, \mathcal{R})$ formed by a set H of hyperplanes in \mathbb{R}^d , where each $R \in \mathcal{R}$ is the subset of H intersected by an arbitrary line segment. Given a convex body σ (not necessarily full-dimensional), consider the arrangement formed by H within the affine span of σ , i.e., the lowest-dimensional flat that contains σ , and let $V(H, \sigma)$ be the set of vertices of this arrangement that lie inside σ .

THEOREM 44.2.9 [Cha93a, BCM99]

Given a set H of hyperplanes in \mathbb{R}^d in general position, along with an ϵ -approximation A for $\Sigma = (H, \mathcal{R})$; for any convex body σ of dimension $k \leq d$, we have

$$\left| \frac{|V(H, \sigma)|}{|H|^k} - \frac{|V(A, \sigma)|}{|A|^k} \right| \leq \epsilon.$$

44.3 GEOMETRIC ALGORITHMS

GLOSSARY

ϵ -Cutting: Given a set H of n hyperplanes in \mathbb{R}^d and $\epsilon > 0$, a collection \mathcal{C} of closed full-dimensional simplices (some of them unbounded) is called an ϵ -cutting if: (i) their interiors are pairwise disjoint, and together they cover \mathbb{R}^d ; (ii) the interior of any simplex of \mathcal{C} is intersected by at most ϵn hyperplanes of H .

Simplicial partition: Given a finite set $P \subset \mathbb{R}^d$, a collection $\{(P_i, R_i)\}$ is a *simplicial partition*, if (i) the P_i 's partition P and (ii) each R_i is a relatively open simplex enclosing P_i . The R_i 's can be of any dimension and need not be disjoint, and P_i need not be equal to $P \cap R_i$. We say that a hyperplane *cuts* R_i if it intersects, but does not contain, R_i . The maximum number of R_i 's that a single hyperplane can cut is the *cutting number* of the simplicial partition.

Partition tree: Given a finite set $P \subset \mathbb{R}^d$, a *partition tree* for P is a rooted tree \mathcal{T} whose root is associated with the point set P . The set P is partitioned into

subsets P_1, \dots, P_m , and each P_i is associated with a distinct child v_i of the root. There is a convex open set R_i , called the *region* of v_i , that contains P_i . The regions R_i are not necessarily disjoint. If $|P_i| > 1$, the subtree rooted at v_i is defined recursively with respect to P_i .

Point location: Preprocess an arrangement of n hyperplanes in \mathbb{R}^d so that, given a query point, one can quickly find the face of the arrangement that contains the point. Note that the face need not be d -dimensional. The complexity of a point location algorithm is measured by the query time and the amount of storage needed for the data structure. The time it takes to do the preprocessing is also of importance.

Simplex range searching: Preprocess a set P of n points in \mathbb{R}^d so that, given a query (closed) simplex σ , the size of $P \cap \sigma$ can be quickly evaluated. Simplex range searching refers to a slight generalization of the problem, in which weights in an additive group or semigroup are assigned to the points and the answer to a query is the sum of all of the weights within σ . This framework allows us to model both the counting and reporting versions of the problem, the latter requiring an explicit enumeration of the points in σ .

CUTTINGS

Clarkson [Cla87] and Haussler and Welzl [HW87] were among the first to introduce the notion of sparsely intersected space partitions for divide and conquer. The definition of an ϵ -cutting is due to Matoušek [Mat91]. Near-optimal ϵ -cutting constructions were given in two dimensions [Aga90, Aga91] and in arbitrary dimension [Mat90, Mat91, Mat95a]. The optimal ϵ -cutting construction cited below is due to Chazelle. It simplified an earlier design by Chazelle and Friedman in [CF90].

THEOREM 44.3.1 [Cha93a]

Given a set H of n hyperplanes in \mathbb{R}^d , for any $r > 0$ there exists a $(1/r)$ -cutting for H of size $O(r^d)$, which is optimal. The cutting, together with the list of hyperplanes intersecting the interior of each simplex, can be found deterministically in $O(nr^{d-1})$ time.

The standard proof of the theorem is based on a hierarchical construction of independent interest. Roughly, the cutting sought is the last one in a sequence of cuttings $\mathcal{C}_0, \dots, \mathcal{C}_m$ such that (i) \mathcal{C}_0 is of constant size; (ii) for $k > 0$, each simplex of \mathcal{C}_k is enclosed in a unique simplex of \mathcal{C}_{k-1} , which itself contains at most a constant number of simplices of \mathcal{C}_k ; and (iii) for some constant $c > 0$, \mathcal{C}_k is a $(1/c^k)$ -cutting of size $O(c^{dk})$.

The simplest application of cuttings is point location in an arrangement of hyperplanes. Consider n hyperplanes in \mathbb{R}^d . Given a query point, how fast can we find the cell (or lower-dimensional face) of the arrangement that contains the point? Assuming general position for simplicity, we set $r = n$ in the theorem. From the nesting structure of $\mathcal{C}_0, \mathcal{C}_1$, etc, we can locate the query point in \mathcal{C}_k (i.e., find the simplex that contains it) in constant time once we know its location within \mathcal{C}_{k-1} .

THEOREM 44.3.2 [Cha93a]

Point location among n hyperplanes can be done in $O(\log n)$ query time, using $O(n^d)$ preprocessing.

A nice application of cuttings is to the problem of deciding whether there exists any point/line incidence among n lines and n points in the plane. This is often called ***Hopcroft's problem***. A well-known construction of Erdős provides an arrangement of n lines such that at least n of its vertices are each incident to $\Omega(n^{1/3})$ edges. Choosing these n lines as input to Hopcroft's problem and placing the n points very near the high-degree vertices suggests that to solve the problem should require checking each point against the $\Omega(n^{1/3})$ lines incident to the nearby vertex, for a total of $\Omega(n^{4/3})$ time. This argument can be made rigorous [Eri96]; it offers a strong hint that to beat $\Omega(n^{4/3})$ might not be easy. The bound itself has not been achieved, although an algorithm by Matoušek, based on a subtle use of cuttings, comes near.

THEOREM 44.3.3 [Mat93]

To decide whether n points and n lines in the plane are free of any incidence can be done in time $n^{4/3} 2^{O(\log^ n)}$.*

SIMPLEX RANGE SEARCHING

Two essential tools in designing data structures for simplex range searching are the *simplicial partition* and the *spanning path*. We mention the key results about these constructions. As a matter of terminology, we say that a hyperplane cuts a line segment if it intersects it but neither of its endpoints. The n points of a square grid can easily be connected by a path so that no line cuts more than roughly \sqrt{n} edges. The optimal low-cutting spanning path construction of Chazelle and Welzl generalizes this result to any set of points in any dimension.

LEMMA 44.3.4 [CW89]

Any set of n points in \mathbb{R}^d can be ordered as p_1, \dots, p_n , in such a way that no hyperplane cuts more than $cn^{1-1/d}$ segments of the form $p_i p_{i+1}$, for some constant $c > 0$.

Simplicial partitions generalize the notion of a spanning path by considering not just edges, i.e., pairs of points, but larger subsets of them. Again, we wish to minimize the cutting number, i.e., the number of subsets a hyperplane can “cut through.” An optimal construction based on cuttings was discovered by Matoušek.

LEMMA 44.3.5 [Mat92]

Given a set P of n points in \mathbb{R}^d ($d > 1$), for any integer $1 < r \leq n/2$ there exists a simplicial partition with cutting number $O(r^{1-1/d})$ such that $n/r \leq |P_i| < 2n/r$ for each (P_i, R_i) in the partition.

The partition tree offers a simple solution to simplex range searching. At each node, store the sum of the weights of the points associated with the corresponding region. Given a query simplex σ , we proceed to explore all children v_i of the root and check whether σ intersects the region R_i of v_i : (i) if the answer is yes, but σ does not completely enclose the region R_i of v_i , then we visit v_i and recurse; (ii) if the answer is yes, but σ completely encloses R_i , we simply add to our current weight count the sum of the weights within P_i , which happens to be stored at v_i ; (iii) if the answer is no, we do not recurse at v_i .

The application of Lemma 44.3.5 for a large enough constant r yields a partition

tree construction that allows us to perform simplex range searching in $O(n^{1-1/d+\epsilon})$ query time, for any fixed $\epsilon > 0$, using $O(n)$ storage. A more complex argument by Matoušek gets rid of the ϵ term in the exponent.

THEOREM 44.3.6 [Mat92]

Given n points in \mathbb{R}^d , there exists a linear size data structure with which simplex range searching can be performed in time $O(n^{1-1/d})$ per query.

If superlinear storage is available, then space-time tradeoffs are possible. Chazelle, Sharir, and Welzl [CSW92] proved that simplex range searching with respect to n points in \mathbb{R}^d can be done in $O(n^{1+\epsilon}/m^{1/d})$ query time, using a data structure of size m , for any $n \leq m \leq n^d$. Matoušek [Mat93] slightly improved the query time to $O(n(\log m/n)^{d+1}/m^{1/d})$, for m/n large enough.

POLYHEDRAL ALGORITHMS

We discuss applications of the discrepancy method to convex hulls, Voronoi diagrams, halfspace intersection, linear programming, and other forms of convex programming.

The problem of computing the convex hull of n points in \mathbb{R}^d reduces by duality to that of computing the intersection of n halfspaces. In addition, computing the Voronoi diagram (or, equivalently, the Delaunay triangulation) of a finite set of points in Euclidean d -space can be reduced in linear time to a convex hull problem in $(d+1)$ -space. An optimal halfspace intersection algorithm can then be used for both the convex hull and the Voronoi diagram problems. The intersection of n halfspaces is a convex polyhedron with $O(n^{\lfloor d/2 \rfloor})$ faces (and possibly as many as that).

A simple approach to the halfspace intersection problem is to insert each halfspace one after the other and maintain the current intersection as we go. A simple data structure consisting of a triangulation of the current intersection polyhedron, together with a bipartite graph indicating which hyperplane intersects which cell of the triangulation, is sufficient to make this process efficient. In fact, if the order of insertion is random, then it follows from the work of Clarkson and Shor [CS89] that, with the right supporting data structure, the expected complexity of the algorithm can be made to be optimal. By combining the use of ϵ -nets, ϵ -approximations, ϵ -cuttings, and product set systems, Chazelle [Cha93b] showed how to compute the intersection deterministically in optimal time (Theorem 44.3.7); his algorithm was subsequently simplified by Brönnimann, Chazelle, and Matoušek [BCM99].

THEOREM 44.3.7

The polyhedron formed by the intersection of n halfspaces in \mathbb{R}^d can be computed in $O(n \log n + n^{\lfloor d/2 \rfloor})$ time.

As indicated earlier, this result has two important consequences: optimal algorithms for convex hulls and for Voronoi diagrams.

THEOREM 44.3.8

The convex hull of a set of n points in \mathbb{R}^d can be computed deterministically in $O(n \log n + n^{\lfloor d/2 \rfloor})$ time.

THEOREM 44.3.9

The Voronoi diagram (or Delaunay triangulation) of a set of n points in \mathbf{E}^d can be computed deterministically in $O(n \log n + n^{\lceil d/2 \rceil})$ time.

Linear programming is the problem of minimizing a linear function $c^T x$, subject to the constraints $Ax \leq b$ and $x \geq 0$, where A is an n -by- d matrix, $b \in \mathbb{R}^n$, and $c, x \in \mathbb{R}^d$. The discrepancy method can be used to derive a deterministic algorithm for linear programming that is linear in n and singly exponential in d .

The best route to this result is via an abstract formalism, called **LP-type programming**, due to Sharir and Welzl [SW92] (see also [MSW96]) that places the method in a much more general context and allows for even more surprising applications. For example, it can be used to prove that, given n points in, say, \mathbb{R}^{99} , the smallest enclosing ellipsoid can be found in $O(n)$ time.

An **LP-type problem** is specified by a pair (H, w) , where H is a finite set whose elements are the “constraints” of the problem, and w is a function mapping certain subsets of H to a totally ordered universe (W, \leq) . An element $h \in H$ is said to **violate** a subset $G \subseteq H$ if $w(G) < w(G \cup \{h\})$. A **basis** B of $G \subseteq H$ is a minimal set of constraints with the same cost as G , i.e., $w(B) = w(G)$ and $w(C) < w(B)$ for any $C \subset B$. The **combinatorial dimension** of (H, w) , denoted by δ , is the maximum size of any basis (of any subset of H). To solve the problem (H, w) is to find a basis of H . We need a few specific assumptions to make computational sense of this framework.

1. **MONOTONICITY.** Given any $F \subseteq G \subseteq H$, $w(F) \leq w(G)$.
2. **LOCALITY.** If $h \in H$ violates $G \subseteq H$, then it violates any basis of G .
3. **ORACLE.** Given a basis B of some subset of H , let $V(B)$ denote the set of violating constraints. Consider the set system (H, \mathcal{R}) , where \mathcal{R} is the collection of sets $V(B)$, for all bases B . It is assumed that (H, \mathcal{R}) has bounded VC-dimension, and let γ be its shatter function exponent. In practice, γ is either equal to or larger than δ . Given any subset $Y \subseteq H$, the oracle computes the set $\mathcal{R}|_Y$ in time $O(|Y|^{\gamma+1})$.

How does linear programming fit into the LP-type framework? For simplicity of explanation, we assume that the optimization function is of the form $(1, 0, \dots, 0)^T x$, and that the system is feasible: (i) H is the set of n closed halfspaces formed by the inequalities $Ax \leq b$; (ii) $W = \mathbb{R}^d$, ordered lexicographically; (iii) given $G \subseteq H$, $w(G)$ is the unique (lexicographically) minimal point with nonnegative coordinates in the halfspaces of G . A halfspace $h \in H$ violates $G \subseteq H$ if $w(G) < w(G \cup \{h\})$, which means that adding h to G would strictly increase the cost of the optimal solution: Geometrically, the hyperplane corresponding to h cuts off the old solution from the new feasible set. A basis consists of at most d halfspaces, and its combinatorial dimension is d . MONOTONICITY says that throwing in additional constraints cannot improve the optimal solution. LOCALITY means that the violation of a set of constraints can always be witnessed locally by focusing on any one of its bases. The oracle can be implemented easily so as to run in time $O(|Y|^{d+1})$.

Solving an LP-Type Problem

STEP 1. Let $D = \max\{\delta, \gamma\}$. If $|H| \leq cD^4 \log D$ for some suitably large constant c , compute a basis of H by checking all possible j -tuples of

constraints in the order $j = 1, \dots, \delta$, and picking the first one that is not violated by any constraint of H .

STEP 2. Compute a $(1/4D^2)$ -net N for (H, \mathcal{R}) .

STEP 3. Find a basis B of N recursively. Let V be the set of constraints of H that violate B . If $V = \emptyset$, then return B and stop; else add all of the violating constraints to the set N and repeat Step 3.

Assuming that, given any basis B and a constraint $h \in H$, to test whether h violates B or not can be done in time $D^{O(D)}$, which is the case in typical applications, LP-type problems can be solved in time linear in the number of constraints and exponential in the number of variables. Chazelle and Matoušek proved the following:

THEOREM 44.3.10 [CM96]

An LP-type problem (H, w) can be solved in time $|H| \cdot O(D^7 \log D)^D$, where D is the combinatorial dimension or the exponent in the complexity of the oracle, whichever is larger.

We mention two applications of this result, also taken from [CM96]. The first one is a linear deterministic algorithm for linear programming with any fixed number of variables. The second one addresses the complexity of finding the **Löwner-John ellipsoid** of n points in d -space, i.e., the smallest ellipsoid enclosing the n points (which is known to be unique).

THEOREM 44.3.11

Linear programming with n constraints and d variables can be solved in $d^{O(d)}n$ time.

THEOREM 44.3.12

The ellipsoid of minimum volume that encloses a set of n points in \mathbb{R}^d can be computed in time $d^{O(d^2)}n$.

LOWER BOUNDS FOR RANGE SEARCHING

An **off-line range searching** problem is specified by n points and m regions in \mathbb{R}^d . Each point p_i is assigned a **weight** x_i chosen in an additive group or semigroup. The output should be the sum of the weights of the points within each of the m regions. In the **on-line** version of the problem, the points and weights are preprocessed into a data structure and a query is a region whose weight sum constitutes the output.

From the algebraic perspective of adding weights, off-line range searching can be regarded as the problem of multiplying a fixed matrix by an arbitrary vector. The n points and m ranges form a set system Σ , whose incidence matrix we denote by A . The problem is to compute the map $x \in \mathbb{R}^n \mapsto Ax \in \mathbb{R}^m$. We use a linear circuit model with bounded coefficients. This is a directed acyclic graph whose nodes, the *gates*, have indegree 2. With each gate g is associated two complex numbers α_g, β_g of modulus $O(1)$. The gate g takes two complex numbers a, b as input and outputs $\alpha_g a + \beta_g b$. The size of the circuit is the number of edges. The **complexity** of the matrix A is the size of the smallest circuit for computing $x \mapsto Ax$. We note that the circuit depends only on A and must “work” for any input $x \in \mathbb{R}^n$.

It is not hard to prove that the size of the circuit is $\Omega(\log |\det B|)$, where B

is the square submatrix of A whose determinant is largest in absolute value: this is the classical **Morgenstern bound** [Mor73]. A stronger result, due to Chazelle, relates the size of the circuit to the singular values of the matrix A .

LEMMA 44.3.13 *Spectral Lemma* [Cha98]

Given an $n \times n$ $(0, 1)$ -matrix A , any circuit for computing Ax has size at least $\Omega(k \log \lambda_k)$, where λ_k is the k th largest eigenvalue of $A^T A$.

A slightly weaker formulation of the spectral bound was given by Chazelle and Lvov [CL01b]. It involves only the traces of $A^T A$ and its square. This has the huge advantage that every component of the formula has a simple combinatorial interpretation: the trace of $A^T A$ (of its square) counts the number of ones (resp. rectangles of ones) in A .

LEMMA 44.3.14 *Trace Lemma* [CL01b]

Given an $n \times n$ $(0, 1)$ -matrix A , any circuit for computing Ax has size

$$\Omega_\epsilon \left(n \log \left(\text{tr } M/n - \epsilon \sqrt{\text{tr } M^2/n} \right) \right),$$

where $M = A^T A$ and $\epsilon > 0$ is an arbitrarily small constant.

The bounds can be made more general to accommodate a few “help” gates, i.e., gates that can compute any function whatsoever [Cha98]. The spectral and trace lemmas have been used to derive bounds for a number of classical range searching problems. The monotone model of computation, where essentially subtractions are disallowed, has also been investigated. We mention the main results below and explain their meaning. The proofs rely heavily on tools from discrepancy theory, in particular, on constructions of low-discrepancy point sets and techniques from harmonic analysis to analyze the spectrum of geometric incidence matrices.

TABLE 44.3.1 Circuit lower bounds for range searching.

TYPE	GENERAL	MONOTONE
Axis-parallel boxes	$\Omega(n \log \log n)$	$\Omega(n(\log n / \log \log n)^{d-1})$
Simplices	$\Omega(n \log n)$	$\tilde{\Omega}(n^{2-2/(d+1)})$
Lines	$\Omega(n \log n)$	$\Omega(n^{4/3})$

The table indicates some of the lower bounds known to date. In all cases, the problem consists of n points in \mathbb{R}^d and n regions whose type is indicated in the first column. The GENERAL column refers to the circuit model discussed above. The proofs were given for $\alpha_g, \beta_g \in \{-1, 0, 1\}$, but extend trivially to any complex numbers with bounded modulus. The bounds for axis-parallel boxes [Cha97] and simplices [Cha98] were proven in dimension 2 and, hence, in any higher dimension. In the case of axis-parallel boxes, the lower bound jumps to $\Omega(n \log n)$ in dimension $\Theta(\log n)$ [CL01a]. The bound for lines was proven in [CL01b].

The MONOTONE column assumes that $\alpha_g, \beta_g \in \{0, 1\}$ at each gate of the circuit. The notation $\tilde{\Omega}(f(n))$ refers to $\Omega(f(n)/(\log n)^{O(1)})$. The bounds for axis-parallel boxes and simplices were given in [Cha97]. The bound for lines is mentioned in [Cha00]. All three bounds are essentially optimal in that model.

It is a fascinating open question how the wide gap between general and nonmonotone complexity is to be resolved. For example, is line range searching in $O(n \log n)$ or $O(n^{4/3})$? Most of the lower bounds for the monotone case have nearly matching upper bounds; in other words, to make effective use of nonmonotone computation seems very difficult. This has led to the widely held belief that the monotone bounds are the true answers. Recent work by Chazelle [Cha02] casts doubt on this conjecture. By means of grid points and line queries that bounce off the grid boundary, the general complexity of the problem is shown to be $\Theta(n \log n)$, while the monotone complexity is $\Theta(n^{3/2})$.

In the on-line version of range searching, the n points are preprocessed so that, given a query region, the sum of the weights of the points in the region can be computed quickly. The following bounds, established by Chazelle in the monotone model, are essentially optimal. Both of them make heavy use of low-discrepancy constructions for point sets in bounded-dimensional space, as well as of related constructions arising from Heilbronn's problem [Rot51].

THEOREM 44.3.15 [Cha89]

Given n points in \mathbb{R}^d , on-line simplex range searching requires $\tilde{\Omega}(n/m^{1/d})$ query time, using a data structure of size m .

THEOREM 44.3.16 [Cha90]

Given n points in \mathbb{R}^d , on-line range searching with axis-parallel box queries requires $\Omega(\log n / \log(2m/n))^{d-1}$ per query, using a data structure of size m .

FURTHER READING

Many aspects of the discrepancy method, including nongeometric ones, are covered in [Cha00]. The related topic of derandomization is surveyed in [Mat96]. The main texts on discrepancy theory are [BC87, Nie92, DT97, Mat99]; see also [BS95].

RELATED CHAPTERS

- [Chapter 13: Geometric discrepancy theory and uniform distribution](#)
- [Chapter 22: Convex hull computations](#)
- [Chapter 23: Voronoi diagrams and Delaunay triangulations](#)
- [Chapter 36: Range searching](#)
- [Chapter 40: Randomization and derandomization](#)
- [Chapter 45: Linear programming](#)

REFERENCES

- [Aga90] P.K. Agarwal. Partitioning arrangements of lines II: Applications. *Discrete Comput. Geom.*, 5:533–573, 1990.
- [Aga91] P.K. Agarwal. Geometric partitioning and its applications. In J.E. Goodman, R. Pollack, and R. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, pages 1–37, Amer. Math. Soc., Providence, 1991.
- [ARS99] N. Alon, L. Rónyai, and L. Szabó. Norm-graphs: variations and applications. *J. Combin. Theory Ser. B*, 76:280–290, 1999.

- [Ass83] P. Assouad. Densité et dimension. *Ann. Inst. Fourier*, 33:233–282, 1983.
- [BC87] J. Beck and W.W.L. Chen. *Irregularities of Distribution*. Volume 89 of Cambridge Tracts in Math., Cambridge University Press, 1987.
- [BS95] J. Beck and V.T. Sós. Discrepancy theory. In R.L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, Chapter 17, pages 1405–1446. North-Holland, Amsterdam, 1995.
- [BCM99] H. Brönnimann, B. Chazelle, and J. Matoušek. Product range spaces, sensitive sampling, and derandomization. *SIAM J. Comput.*, 28:1552–1575, 1999.
- [Cha89] B. Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2:637–666, 1989.
- [Cha90] B. Chazelle. Lower bounds for orthogonal range searching: II. The arithmetic model, *J. Assoc. Comput. Mach.*, 37:439–463, 1990.
- [Cha93a] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9:145–158, 1993.
- [Cha93b] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.
- [Cha97] B. Chazelle. Lower bounds for off-line range searching. *Discrete Comput. Geom.*, 17:53–65, 1997.
- [Cha98] B. Chazelle. A spectral approach to lower bounds with applications to geometric searching. *SIAM J. Comput.*, 27:545–556, 1998.
- [Cha00] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, hardcover 2000, paperback 2001.
- [Cha02] B. Chazelle. The power of nonmonotonicity in geometric searching. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 2002.
- [CF90] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10:229–249, 1990.
- [CL01a] B. Chazelle and A. Lvov. The discrepancy of boxes in higher dimension. *Discrete Comput. Geom.*, 25:519–524, 2001.
- [CL01b] B. Chazelle and A. Lvov. A trace bound for the hereditary discrepancy. *Discrete Comput. Geom.*, 26:221–231, 2001.
- [CM96] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:579–597, 1996.
- [CSW92] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8:407–429, 1992.
- [CW89] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489, 1989.
- [Cla87] K.L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.
- [CS89] K.L. Clarkson and P.W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- [DT97] M. Drmota and R.F. Tichy. *Sequences, Discrepancies and Applications*. Volume 1651 of *Lecture Notes in Math.*, Springer-Verlag, Berlin, 1997.
- [Eri96] J. Erickson. New lower bounds for Hopcroft’s problem. *Discrete Comput. Geom.*, 16:389–418, 1996.

- [Fre81] M.L. Fredman. Lower bounds on the complexity of some optimal data structures. *SIAM J. Comput.*, 10:1–10, 1981.
- [HW87] D. Haussler and E. Welzl. ϵ -nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.
- [Joh74] D.S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278, 1974.
- [KPW92] J. Komlós, J. Pach, and G. Woeginger. Almost tight bounds for ϵ -nets. *Discrete Comput. Geom.*, 7:163–173, 1992.
- [Lov75] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390, 1975.
- [Mat90] J. Matoušek. Construction of ϵ -nets. *Discrete Comput. Geom.*, 5:427–448, 1990.
- [Mat91] J. Matoušek. Cutting hyperplane arrangements. *Discrete Comput. Geom.*, 6:385–406, 1991.
- [Mat92] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.
- [Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993.
- [Mat95a] J. Matoušek. Approximations and optimal geometric divide-and-conquer. *J. Comput. System Sci.*, 50:203–208, 1995.
- [Mat95b] J. Matoušek. Tight upper bounds for the discrepancy of halfspaces. *Discrete Comput. Geom.*, 13:593–601, 1995.
- [Mat96] J. Matoušek. Derandomization in computational geometry. *J. Algorithms*, 20:545–580, 1996.
- [Mat97] J. Matoušek. On discrepancy bounds via dual shatter functions. *Mathematika*, 44:42–49, 1997.
- [Mat99] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide*. Volume 18 of *Algorithms Combin.*, Springer-Verlag, Berlin, 1999.
- [MSW96] J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.
- [MWW93] J. Matoušek, E. Welzl, and L. Wernisch. Discrepancy and ϵ -approximations for bounded VC-dimension. *Combinatorica*, 13:455–466, 1993.
- [Mor73] J. Morgenstern. Note on a lower bound of the linear complexity of the fast Fourier transform. *J. Assoc. Comput. Mach.*, 20:305–306, 1973.
- [Nie92] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-NSF, SIAM, Philadelphia, 1992.
- [PA95] J. Pach and P.K. Agarwal. *Combinatorial Geometry*. Wiley-Intersci. Ser. Discrete Math. Optim., Wiley, New York, 1995.
- [Rot51] K.F. Roth. On a problem of Heilbronn. *J. London Math. Soc.*, 26:198–204, 1951.
- [Sau72] N. Sauer. On the density of families of sets. *J. Combin. Theory Ser. A*, 13:145–147, 1972.
- [SW92] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, volume 577 of *Lecture Notes in Comput. Sci.*, pages 569–579. Springer-Verlag, Berlin, 1992.
- [She72] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific J. Math.*, 41:247–261, 1972.
- [VC71] V.N. Vapnik and A.Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16:264–280, 1971.

45 LINEAR PROGRAMMING

Martin Dyer, Nimrod Megiddo, and Emo Welzl

INTRODUCTION

Linear programming has many important practical applications, and has also given rise to a wide body of theory. See Section 45.9 for recommended sources. Here we consider the linear programming problem in the form of maximizing a linear function of d variables subject to n linear inequalities. We focus on the relationship of the problem to computational geometry, i.e., we consider the problem in small dimension. More precisely, we concentrate on the case where $d \ll n$, i.e., $d = d(n)$ is a function that grows very slowly with n . By linear programming duality, this also includes the case $n \ll d$. This has been called *fixed-dimensional* linear programming, though our viewpoint here will not treat d as constant. In this case there are strongly polynomial algorithms, provided the rate of growth of d with n is small enough.

The plan of the chapter is as follows. In Section 45.2 we consider the simplex method, in Section 45.3 we review deterministic linear time algorithms, in Section 45.4 randomized algorithms, and in Section 45.5 we consider the derandomization of the latter. Section 45.6 discusses the combinatorial framework of LP-type problems, which underlie most current combinatorial algorithms and allows their application to a host of optimization problems. In Section 45.7 we examine parallel algorithms for this problem, and finally in Section 45.8 we briefly discuss related issues. The emphasis throughout is on complexity-theoretic bounds for the linear programming problem in the form 45.1.1.

45.1 THE BASIC PROBLEM

Any linear program (LP) may be expressed in the inequality form

$$\begin{aligned} & \text{maximize} && z = c.x \\ & \text{subject to} && Ax \leq b, \end{aligned} \tag{45.1.1}$$

where $c \in \mathbb{R}^d$, $b \in \mathbb{R}^n$, and $A \in \mathbb{R}^{n \times d}$ are the input data and $x \in \mathbb{R}^d$ the variables. Without loss of generality, the columns of A are assumed to be linearly independent. The vector inequality in (45.1.1) is with respect to the componentwise partial order on \mathbb{R}^n . We will write a_i for the i th row of A , so the constraint may also be expressed in the form

$$a_i \cdot x = \sum_{j=1}^d a_{ij} x_j \leq b_i \quad (i = 1, \dots, n). \tag{45.1.2}$$

GLOSSARY

Constraint: A condition that must be satisfied by a solution.

Inequality form: The formulation of the linear programming problem where all the constraints are weak inequalities $a_i \cdot x \leq b_i$.

Feasible set: The set of points that satisfy all the constraints. In the case of linear programming, it is a convex polyhedron in \mathbb{R}^d .

Defining hyperplanes: The hyperplanes described by the equalities $a_i \cdot x = b_i$.

Tight constraint: An inequality constraint is tight at a certain point if the point lies on the corresponding hyperplane.

Infeasible problem: A problem with an empty feasible set.

Unbounded problem: A problem with no finite maximum.

Vertex: A feasible point where at least d linearly independent constraints are tight.

Nondegenerate problem: A problem where at each vertex precisely d constraints are tight.

Strongly polynomial-time algorithm: An algorithm for which the total number of arithmetic operations and comparisons (on numbers whose size is polynomial in the input length) is bounded by a polynomial in n and d alone.

We observe that (45.1.1) may be infeasible or unbounded, or have multiple optima. A complete algorithm for linear programming must take account of these possibilities. In the case of multiple optima, we assume that we have merely to identify *some* optimum solution. (The task of identifying *all* optima is considerably more complex; see [Dye83, AF92].) An optimum of (45.1.1) will be denoted by x^0 . At least one such solution (assuming one exists) is known to lie at a vertex of the feasible set. There is little loss in assuming nondegeneracy for theoretical purposes, since we may “infinitesimally perturb” the problem to ensure this using well-known methods [Sch86]. However, a complete algorithm must recognize and deal with this possibility.

It is well known that linear programs can be solved in time polynomial in the total length of the input data. However, it is not known in general if there is a *strongly* polynomial-time algorithm. This is true even if randomization is permitted. (Algorithms mentioned below may be assumed deterministic unless otherwise stated.) The “weakly” polynomial algorithms make crucial use of the size of the numbers, so seem unlikely to lead to strongly polynomial methods. However, strong bounds are known in some special cases. For example, if all a_{ij} are bounded by a constant, then É. Tardos [Tar86] has given a strongly polynomial algorithm.

45.2 THE SIMPLEX METHOD

GLOSSARY

Simplex method: For a nondegenerate problem in inequality form, this method seeks an optimal vertex by iteratively moving from one vertex to a better neighboring vertex.

Pivot rule: The rule by which a neighboring vertex is chosen.

Random-edge simplex algorithm: A randomized variant of the simplex method where the neighboring vertex is chosen uniformly at random.

Dantzig's simplex method is probably still the most commonly used method for solving large linear programs in practice, but (with standard pivot rules) Klee and Minty showed that Dantzig's pivot rule may require an exponential number of iterations in the worst case. For example, it may require $2^d - 1$ iterations when $n = 2d$. Other variants were subsequently shown to have similar behavior. While it is not known for certain that all suggested variants of the simplex method have this bad worst case, there seems to be no reason to believe otherwise. In our case $d \ll n$, the simplex method may require $\Omega(n^{d/2})$ iterations [KM72, AZ99], and thus it is polynomial only for $d = O(1)$. This is asymptotically no better than enumerating all vertices of the feasible region.

By contrast, Kalai [Kal92] gave a randomized simplex-like algorithm that requires only $2^{O(\sqrt{d \log n})}$ iterations. (An identical bound was also given by Matoušek, Sharir, and Welzl [MSW96] for a closely related algorithm; see Section 45.4.) Combined with Clarkson's methods [Cla95], this results in a bound of $O(d^2n) + e^{O(\sqrt{d \log d})}$ (cf. [MSW96]). This is the best “strong” bound known, other than for various special problems, and it is evidently polynomial provided $d = O(\log^2 n / \log \log n)$. No complete derandomization of these algorithms is known, and it is possible that randomization may genuinely help here. In this respect, the complexity of the so-called random-edge simplex method (where the pivot is chosen uniformly at random) is an open question. See [BDF⁺95, GHZ98, GST⁺01] for some limited information.

45.3 LINEAR-TIME LINEAR PROGRAMMING

The study of linear programming within computational geometry was initiated by Shamos [Sha78] as an application of an $O(n \log n)$ convex hull algorithm for the intersection of halfplanes. Muller and Preparata [MP78] gave an $O(n \log n)$ algorithm for the intersection of halfspaces in \mathbb{R}^3 . Dyer [Dye84] and Megiddo [Meg83] found, independently, an $O(n)$ time algorithm for the linear programming problem in the cases $d = 2, 3$.

Megiddo [Meg84] generalized the approach of these algorithms to arbitrary d , arriving at an algorithm of complexity $O(2^{2^d} n)$, which is polynomial for $d \leq \log \log n + O(1)$. This was subsequently improved by Clarkson [Cla86b] and Dyer [Dye86] to $O(3^{d^2} n)$, which is polynomial for $d = O(\sqrt{\log n})$. Megiddo [Meg84, Meg89] and Dyer [Dye86, Dye92] showed that Megiddo's idea could be used for many related problems: Euclidean one-center, minimum ball containing balls, minimum volume ellipsoid, etc.; see also the derandomized methods and LP-type problems in the sections below.

GLOSSARY

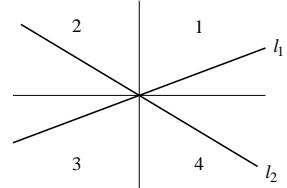
Multidimensional search: Given a set of hyperplanes and an oracle for locating a point relative to any hyperplane, locate the point relative to all the input hyperplanes.

MEGIDDO'S ALGORITHMS

The basic idea in these algorithms is as follows. It follows from convexity considerations that either the constraints in a linear program are tight (i.e., satisfied with equality) at x^0 , or they are irrelevant. We need identify only d tight constraints to identify x^0 . We do this by discarding a fixed proportion of the irrelevant constraints at each iteration. Determining whether the i th constraint is tight amounts to determining which case holds in $a_i \cdot x^0 \geq b_i$. This is embedded in a multidimensional search problem. Given *any* hyperplane $\alpha \cdot x = \beta$, we can determine which case of $\alpha \cdot x^0 \geq \beta$ holds by (recursively) solving three linear programs in $d - 1$ variables. These are (45.1.1) plus $\alpha \cdot x = \gamma$, where $\gamma \in \{\beta - \epsilon, \beta, \beta + \epsilon\}$ for “small” $\epsilon > 0$. (We need not define ϵ explicitly; it can be handled symbolically.) In each of the three linear programs we eliminate one variable to get $d - 1$. The largest of the three objective functions tells us where x^0 lies with respect to the hyperplane. We call this an *inquiry* about $\alpha \cdot x = \beta$. The problem now reduces to locating x^0 with respect to a proportion $P(d)$ of the n hyperplanes using only $N(d)$ inquiries.

The method recursively uses the following observation in \mathbb{R}^2 . Given two lines through the origin with slopes of opposite sign, knowing which quadrant a point lies in allows us to locate it with respect to *at least one* of the lines (see Figure 45.3.1).

FIGURE 45.3.1
Quadrants 1, 3 locate for l_2 ; quadrants 2, 4 locate for l_1 .



We use this on the first two coordinates of the problem in \mathbb{R}^d . First rotate until $\frac{1}{2}n$ defining hyperplanes have positive and $\frac{1}{2}n$ negative “slopes” on these coordinates. This can be done in $O(n)$ time using median-finding. Then arbitrarily pair a positive with a negative to get $\frac{1}{2}n$ pairs of the form

$$\begin{aligned} ax_1 + bx_2 + \cdots &= \cdots \\ cx_1 - dx_2 + \cdots &= \cdots, \end{aligned}$$

where a, b, c, d represent nonnegative numbers, and the \cdots represent linear functions of x_3, \dots, x_d on the left and arbitrary numbers on the right. Eliminating x_2 and x_1 in each pair gives two families S_1, S_2 of $\frac{1}{2}n$ hyperplanes each in $d - 1$ dimensions of the form

$$\begin{aligned} S_1 : \quad x_1 &+ \cdots = \cdots \\ S_2 : \quad x_2 &+ \cdots = \cdots. \end{aligned}$$

We recursively locate with respect to $\frac{1}{2}P(d-1)n$ hyperplanes with $N(d-1)$ inquiries in S_1 , and then locate with respect to a $P(d-1)$ -fraction of the corresponding paired hyperplanes in S_2 . We have then located $\frac{1}{2}P(d-1)^2n$ pairs with $2N(d-1)$ inquiries. Using the observation above, each pair gives us location with respect to at least

one hyperplane in d dimensions, i.e.,

$$P(d) = \frac{1}{2}P(d-1)^2, \quad N(d) = 2N(d-1). \quad (45.3.1)$$

Since $P(1) = \frac{1}{2}$, $N(1) = 1$ (by locating with respect to the median in \mathbb{R}^1), (45.3.1) yields

$$P(d) = 2^{-(2^d-1)}, \quad N(d) = 2^{d-1},$$

giving the following time bound $T(n, d)$ for solving (45.1.1).

$$T(n, d) \leq 3 \cdot 2^{d-1} T(n, d-1) + T((1 - 2^{-(2^d-1)})n, d) + O(nd),$$

with solution $T(n, d) = O(2^{2^d} n)$.

THE CLARKSON-DYER IMPROVEMENT

The Clarkson/Dyer improvement comes from repeatedly locating in S_1 and S_2 to increase $P(d)$ at the expense of $N(d)$.

45.4 RANDOMIZED ALGORITHMS

Dyer and Frieze [DF89] showed that, by applying an idea of Clarkson [Cla86a] to give a *randomized* solution of the multidimensional search in Megiddo's algorithm [Meg84], an algorithm of complexity $O(d^{3d+o(d)} n)$ was possible. Clarkson [Cla88, Cla95] improved this dramatically. We describe this below, but first outline a simpler algorithm subsequently given by Seidel [Sei91].

Suppose we order the constraints randomly. At stage k , we have solved the linear program subject to constraints $i = 1, \dots, k-1$. We now wish to add constraint k . If it is satisfied by the current optimum we finish stage k and move to $k+1$. Otherwise, the new constraint is clearly tight at the optimum over constraints $i = 1, \dots, k-1$. Thus, recursively solve the linear program subject to this equality (i.e., in dimension $d-1$) to get the optimum over constraints $i = 1, \dots, k$, and move on to $k+1$. Repeat until $k = n$.

The analysis hinges on the following observation. When constraint k is added, the probability it is not satisfied is exactly d/k (assuming, without loss, nondegeneracy). This is because only d constraints are tight at the optimum and this is the probability of writing one of these *last* in a random ordering of $1, 2, \dots, k$. This leads to an expected time of $O(d!n)$ for (45.1.1). Welzl [Wel91] extended Seidel's algorithm to solve other problems such as smallest enclosing ball or ellipsoid, and described variants that perform favorably in practice.

Sharir and Welzl [SW92] modified Seidel's algorithm, resulting in an improved running time of $O(d^3 2^d n)$. They put their algorithm in a general framework of solving "LP-type" problems (see Section 45.6 below). Matousěk, Sharir, and Welzl [MSW96] improved the analysis further, essentially obtaining the same bound as for Kalai's "primal simplex" algorithm. The algorithm was extended to LP-type problems by Gärtner [Gär95], with a similar time bound.

CLARKSON'S ALGORITHM

The basic idea is to choose a random set of r constraints, and solve the linear program subject to these. The solution will violate “few” constraints among the remaining $n - r$, and, moreover, one of these must be tight at x^0 . We solve a new linear program subject to the violated constraints and a new random subset of the remainder. We repeat this procedure (aggregating the old violated constraints) until there are no new violated constraints, in which case we have found x^0 . Each repetition gives an extra tight constraint for x^0 , so we cannot perform more than d iterations.

Clarkson [Cla88] gave a different analysis, but using Seidel’s idea we can easily bound the expected number of violated constraints (see also [GW01] for further simplifications of the algorithm). Imagine all the constraints ordered randomly, our sample consisting of the first r . For $i > r$, let $I_i = 1$ if constraint i is violated, $I_i = 0$ otherwise. Now $\Pr(I_i = 1) = \Pr(I_{r+1} = 1)$ for all $i > r$ by symmetry, and $\Pr(I_{r+1} = 1) = d/(r + 1)$ from above. Thus the expected number of violated constraints is

$$\mathbf{E} \left(\sum_{i=r+1}^n I_i \right) = \sum_{i=r+1}^n \Pr(I_i = 1) = (n - r)d/(r + 1) < nd/r.$$

(In the case of degeneracy, this will be an upper bound by a simple perturbation argument.)

Thus, if $r = \sqrt{n}$, say, there will be at most $d\sqrt{n}$ violated constraints in expectation. Hence, by Markov’s inequality, with probability $\frac{1}{2}$ there will be at most $2d\sqrt{n}$ violated constraints in actuality. We must therefore recursively solve about $2(d + 1)$ linear programs with at most $(2d^2 + 1)\sqrt{n}$ constraints. The “small” base cases can be solved by the simplex method in $d^{O(d)}$ time. This can now be applied recursively, as in [Cla88], to give a bound for (45.1.1) of

$$O(d^2 n) + (\log n)^{\log d + 2} d^{O(d)}.$$

Clarkson [Cla95] subsequently modified his algorithm using a different “iterative reweighting” algorithm to solve the $d + 1$ small linear programs, obtaining a better bound on the execution time.

Each constraint receives an initial weight of 1. Random samples of total weight $10d^2$ (say) are chosen at each iteration, and solved by the simplex method. If W is the current total weight of all constraints, and W' the weight of the unsatisfied constraints, then $W' \leq 2Wd/10d^2 = W/5d$ with probability at least $\frac{1}{2}$ by the discussion above, regarding the weighted constraints as a multiset. We now double the weights of all violated constraints and repeat until there are no violated constraints. This terminates in $O(d \log n)$ iterations by the following argument. After k iterations we have

$$W \leq \left(1 + \frac{1}{5d}\right)^k n \leq ne^{k/5d},$$

and W^* , the total weight of the d optimal constraints, satisfies $W^* \geq 2^{k/d}$, since at least one is violated at each iteration. Now it is clear that $W^* < W$ only while $k < Cd \ln n$, for some constant C . Applying this to the $d + 1$ small linear programs gives overall complexity

$$O(d^2 n + d^4 \sqrt{n} \log n) + d^{O(d)} \log n.$$

This is almost the best time known for linear programming, except that Kalai's algorithm (or [MSW96]) can be used to solve the base cases rather than the simplex method. Then we get the improved bound (cf. [GW96])

$$O(d^2 n) + e^{O(\sqrt{d \log d})}.$$

This is polynomial for $d = O(\log^2 n / \log \log n)$, and is the best bound to date.

45.5 DERANDOMIZED METHODS

Somewhat surprisingly, the randomized methods of Section 45.4 can also lead to the best *deterministic* algorithms for (45.1.1). Matoušek and Chazelle [CM96] produced a derandomized version of Clarkson's algorithm.

The idea, which has wider application, is based on finding (in linear time) *approximations* to the constraint set. If N is a constraint set, then for each $x \in \mathbb{R}^d$ let $V(x, N)$ be the set of constraints violated at x . A set $S \subseteq N$ is an ϵ -approximation to N if, for all x ,

$$\left| \frac{|V(x, S)|}{|S|} - \frac{|V(x, N)|}{|N|} \right| < \epsilon.$$

(See also [Chapters 36](#) and 40.) Since $n = |N|$ hyperplanes partition \mathbb{R}^d into only $O(n^d)$ regions, there is essentially only this number of possible cases for x , i.e., only this number of different sets $V(x, N)$. It follows from the work of Vapnik and Chervonenkis that a (d/r) -approximation of size $O(r^2 \log r)$ always exists, since a random subset of this size has the property with nonzero probability. If we can find such an approximation deterministically, then we can use it in Clarkson's algorithm in place of random sampling. If we use a (d/r) -approximation, then, if x^* is the linear programming optimum for the subset S , $|V(x^*, S)| = 0$, so that

$$|V(x^*, N)| < |N|d/r = nd/r,$$

as occurs in expectation in the randomized version. The implementation involves a refinement based on two elegant observations about approximations, which both follow directly from the definition.

- (i) An ϵ -approximation of a δ -approximation is an $(\epsilon + \delta)$ -approximation of the original set.
- (ii) If we partition N into q equal sized subsets N_1, \dots, N_q and take an (equal sized) ϵ -approximation S_i in each N_i ($i = 1, \dots, q$), then $S_1 \cup \dots \cup S_q$ is an ϵ -approximation of N .

We then *recursively* partition N into q equal sized subsets, to give a “partition tree” of height k , say, as in Figure 45.5.1 (cf. Section 36.2). The sets at level 0 in the partition tree are “small.” We calculate an ϵ_0 -approximation in each. We now take the union of these approximations at level 1 and calculate an ϵ_1 -approximation of this union. This is an $(\epsilon_0 + \epsilon_1)$ -approximation of the whole level

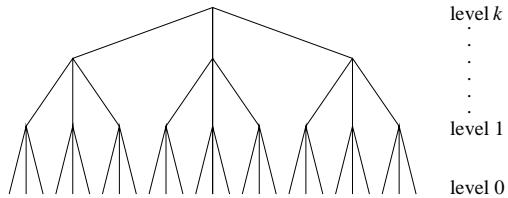


FIGURE 45.5.1

A partition tree of height k , with $q = 3$.

1 set, by the above observations. Continuing up the tree, we obtain an overall $(\sum_{i=0}^k \epsilon_i)$ -approximation of the entire set. At each stage, the sets on which we have to find the approximations remain “small” if the ϵ_i are suitably chosen. Therefore we can use a relatively inefficient method of finding an approximation. A suitable method is the *method of conditional probabilities* due to Raghavan and Spencer. It is (relatively) straightforward to implement this on a set of size m to run in $O(m^{d+1})$ time. However, since this has to be applied only to small sets (in comparison with n), the total work can be bounded by a linear function of n . Chazelle and Matoušek [CM96] used $q = 2$, and an ϵ_i that corresponds to roughly halving the union at each level $i = 1, \dots, k$.

The algorithm cannot completely mimic Clarkson’s, however, since we can no longer use $r = \sqrt{n}$. Such a large approximation cannot be determined in linear time by the above methods. But much smaller values of r suffice (e.g., $r = 10d^3$) simply to get *linear-time* behavior in the recursive version of Clarkson’s algorithm. Using this observation, Chazelle and Matoušek [CM96] obtained a deterministic algorithm with time-complexity $d^{O(d)}n$. This is currently the best time bound known for solving (45.1.1), and remains polynomial for $d = O(\log n / \log \log n)$.

45.6 LP-TYPE PROBLEMS

The randomized algorithms above by Clarkson and in [MSW96, Gär95] can be formulated in an abstract framework called *LP-type problems*. With an extra condition (involving VC-dimension of certain set-systems) this extends to the derandomization in [CM96]. In this way, the algorithms are applicable to a host of problems including smallest enclosing ball, polytope distance, smallest enclosing ellipsoid, largest ellipsoid in polytope, smallest ball intersecting a set of convex objects, angle-optimal placement in polygon, rectilinear 3-centers in the plane, spherical separability, width of thin point sets in the plane, and integer linear programming (see [MSW96, GW96] for descriptions of these problems and of the reductions needed). A different abstraction called *abstract objective functions* is described by Kalai in [Kal97], and for the even more general setting of *abstract optimization problems* see [Gär95].

For the definitions below, the reader should think of optimization problems in which we are given some set H of constraints and we want to *minimize* some given function under those constraints. For every subset G of H , let $w(G)$ denote the optimum value of this function when all constraints in G are satisfied. The function w is only given implicitly via some basic operations to be specified below. The goal is to compute an inclusion-minimal subset B_H of H with the same value as H (from which, in general, the value is easy to determine).

GLOSSARY

LP-type problem: A pair (H, w) , where H is a finite set and $w : 2^H \rightarrow \mathcal{W}$ for a linearly ordered set (\mathcal{W}, \leq) with a minimal element $-\infty$, so that the monotonicity and locality axioms below are satisfied.

Monotonicity axiom: For any F, G with $F \subseteq G \subseteq H$, we have $w(F) \leq w(G)$.

Locality axiom: For any $F \subseteq G \subseteq H$ with $-\infty \neq w(F) = w(G)$ and for any $h \in H$, $w(G) < w(G \cup \{h\})$ implies $w(F) < w(F \cup \{h\})$.

Constraints of LP-type problem: Given an LP-type problem (H, w) , the elements of H are called constraints.

Basis: A set B of constraints is called a basis if $w(B') < w(B)$ for every proper subset of B .

Basis of set of constraints: Given a set G of constraints, a subset $B \subseteq G$ is called a basis of G if it is a basis and $w(B) = w(G)$ (i.e., an inclusion-minimal subset of G with equal w -value).

Combinatorial dimension: The maximum cardinality of any basis in an LP-type problem (H, w) , denoted by $\delta = \delta_{(H, w)}$.

Basis regularity: An LP-type problem (H, w) is basis-regular if, for every basis B with $|B| = \delta$ and for every constraint h , all bases of $B \cup \{h\}$ have exactly δ elements.

Violation test: Decides whether or not $w(B) < w(B \cup \{h\})$, for a basis B and a constraint h .

Basis computation: Delivers a basis of $B \cup \{h\}$, for a basis B and a constraint h .

A simple example of an LP-type problem is the smallest enclosing ball problem (this problem traces back to J.J. Sylvester [Syl57]): Let S be a finite set of points in \mathbb{R}^d , and for $G \subseteq S$, let $\rho(G)$ be the radius of the ball of smallest volume containing G (with $\rho(\emptyset) = -\infty$). Then (S, ρ) is an LP-type problem with combinatorial dimension at most $d + 1$. A violation test amounts to a test deciding whether a point lies in a given ball, while an efficient implementation of basis computations is not obvious (cf. [Gär95]).

Many more examples have been indicated above. As the name suggests, linear programming can be formulated as an LP-type problem, although some care is needed in the presence of degeneracies. Let us assume that we want to maximize the objective function $-x_d$ in (45.1.1), i.e., we are looking for a point in \mathbb{R}^d of smallest x_d -coordinate. In the underlying LP-type problem, the set H of constraints is given by the halfspaces as defined by (45.1.2). For a subset G of these constraints, let $v(G)$ be the backwards lexicographically smallest point satisfying these constraints, with $v(G) := -\infty$ if G gives rise to an unbounded problem, and with $v(G) := \infty$ in case of infeasibility. We assume the backwards lexicographical ordering on \mathbb{R}^d to be extended to $\mathbb{R}^d \cup \{-\infty, \infty\}$ by letting $-\infty$ and ∞ be the minimal and maximal element, resp. The resulting pair (H, v) is LP-type of combinatorial dimension at most $d + 1$. In fact, if the problem is feasible and bounded, then the LP-type problem is basis-regular of combinatorial dimension d . The violation test and basis computation (this amounts to a dual pivot step) are easy to implement.

Matoušek, Sharir, and Welzl [MSW96] showed that a basis-regular LP-type problem (H, w) of combinatorial dimension δ with n constraints can be solved (i.e.,

a basis of H can be determined) with an expected number of at most

$$\min\{e^{2\sqrt{\delta \ln((n-\delta)/\sqrt{\delta})} + O(\sqrt{\delta} + \ln n)}, 2^{\delta+2}(n-\delta)\} \quad (45.6.1)$$

violations tests and basis computations, provided an initial basis B_0 with $|B_0| = \delta$ is available. (For linear programming one can easily generate such an initial basis by adding d symbolic constraints at “infinity”.) Then Gärtner [Gär95] was able to generalize this bound to all LP-type problems. Combining this with Clarkson’s methods, one gets a bound (cf. [GW96]) of

$$O(\delta n) + e^{O(\sqrt{\delta \log \delta})},$$

the best bound known up to now.

Matoušek [Mat94] provided a family of LP-type problems, for which the bound (45.6.1) is tight for the algorithm provided in [MSW96]. It is an open problem, though, whether the algorithm performs faster when applied to linear programming instances. In fact, Gärtner [Gär02] showed that the algorithm is quadratic for the instances in Matousek’s lower bound family which are realizable as linear programming problems as in (45.1.1).

Amenta [Ame94] considers the following extension of the abstract framework: Suppose we are given a family of LP-type problems (H, w_λ) , parameterized by a real parameter λ ; the underlying ordered value set \mathcal{W} has a maximum element ∞ representing *infeasibility*. The goal is to find the smallest λ for which (H, w_λ) is feasible, i.e., $w_\lambda(H) < \infty$. [Ame94] provides conditions under which such a problem can be transformed into a single LP-type problem, and she gives bounds on the resulting combinatorial dimension. This work exhibits interesting relations between LP-type problems and Helly-type theorems (see also [Ame96]).

45.7 PARALLEL ALGORITHMS

GLOSSARY

PRAM: Parallel Random Access Machine. (See [Section 42.1](#) for more information on this and the next two terms.)

EREW: Exclusive Read Exclusive Write.

CRCW: Concurrent Read Concurrent Write.

P: The class of polynomial time problems.

NC: The class of problems that have poly-logarithmic parallel time algorithms running a polynomial number of processors.

P-complete problem: A problem in P whose membership in NC implies $P = NC$.

Expander: A graph in which, for every set of nodes, the set of the neighbors of the nodes is relatively large.

We will consider only PRAM algorithms. (See also [Section 42.2](#).)

The general linear programming problem has long been known to be P-complete, so there is little hope of very fast parallel algorithms. However, the situation is different in the case $d \ll n$, where the problem is in NC if d grows slowly enough.

First, we note that there is a straightforward parallel implementation of Megiddo's algorithm [Meg83] that runs in $O((\log n)^d)$ time on an EREW PRAM. However, this algorithm is rather inefficient in terms of processor utilization, since at the later stages, when there are few constraints remaining, most processors are idle. However, Deng [Den90] gave an “optimal” $O(n)$ work implementation in the plane running in $O(\log n)$ time on a CRCW PRAM with $O(n/\log n)$ processors. Deng's method does not seem to generalize to higher dimensions.

Alon and Megiddo [AM94] gave a *randomized* parallel version of Clarkson's algorithm which, with high probability, runs in constant time on a CREW PRAM in fixed dimension. Here the “constant” is a function of dimension only, and the probability of failure to meet the time bound is small for $n \gg d$.

Ajtai and Megiddo [AM96] attempted to improve the processor utilization in parallelizing Megiddo's algorithm for general d . They gave an intricate algorithm based on using an expander graph to select more nondisjoint pairs so as to utilize all the processors and obtain more rapid elimination. The resulting algorithm for (45.1.1) runs in $O((\log \log n)^d)$ time, but in a nonuniform model of parallel computation based on Valiant's comparison model. The model, which is stronger than the CRCW PRAM, requires $O(\log \log n)$ time median selection from n numbers using n processors, and employs an $O(\log \log n)$ time scheme for compacting the data after deletions, again based on a nonuniform use of expander graphs. A lower bound of $\Omega(\log n/\log \log n)$ time for median-finding on the CRCW PRAM follows from results of Beame and Håstad. Thus Ajtai and Megiddo's algorithm could not be implemented directly on the CRCW PRAM. Within Ajtai and Megiddo's model there is a lower bound $\Omega(\log \log n)$ for the case $d = 1$ implied by results of Valiant. This extends to the CRCW PRAM, and is the only lower bound known for solving (45.1.1) in this model.

Dyer [Dye95] gave a different parallelization of Megiddo's algorithm, which avoids the use of expanders. The method is based on forming groups of size $r \geq 2$, rather than simple pairs. As constraints are eliminated, the size of the groups is gradually increased to utilize the extra processors. Using this, Dyer [Dye95] establishes an $O(\log n(\log \log n)^{d-1})$ bound in the EREW model. It is easy to show that there is an $\Omega(\log n)$ lower bound for solving (45.1.1) on the EREW PRAM, even with $d = 1$. (See [KR90].) Thus improvements on Dyer's bound for the EREW model can only be made in the $\log \log n$ term. However, there was still an open question in the CRCW model, since exact median-finding and data compaction cannot be performed in time polynomial in $\log \log n$.

Goodrich [Goo93] solved these problems for the CRCW model by giving fast implementations of derandomization techniques similar to those outlined in Section 45.5. However, the randomized algorithm that underlies the method is not a parallelization of Clarkson's algorithm, but is similar to a parallelized version of that of Dyer and Frieze [DF89]. He achieves a work-optimal (i.e., $O(n)$ work) algorithm running in $O(\log \log n)^d$ time on the CRCW PRAM. The methods also imply a work-optimal EREW algorithm, but only with the same time bound as Dyer's. Neither Dyer nor Goodrich is explicit about the dependence on d of the execution time of their algorithms.

Independently of Goodrich's work, Sen [Sen95] has shown how to directly modify Dyer's algorithm to give a work-optimal algorithm with $O((\log \log n)^{d+1})$ execution time in the CRCW model. The “constant” in the running time is shown to be $2^{O(d^2)}$. To achieve this, he uses approximate median-finding and approximate data compaction operations, both of which can be done in time polynomial in $\log \log n$.

on the common CRCW PRAM. These additional techniques are, in fact, both examples of derandomized methods and similar to those Goodrich uses for the same purpose. Note that this places linear programming in NC provided $d = O(\sqrt{\log n})$. This is the best result known, although Goodrich's algorithm may give a better behavior once the “constant” has been explicitly evaluated. We may also observe that the Goodrich/Sen algorithms improve on Deng's result in \mathbb{R}^2 .

There is still room for some improvements in this area, but there now seems to be a greater need for sharper lower bounds, particularly in the CRCW case.

45.8 RELATED ISSUES

GLOSSARY

Integer programming problem: A linear programming problem with the additional constraint that the solution must be integral.

k -violation linear programming: A problem as in 45.1.1, except that we want to maximize the linear objective function subject to all but at most k of the given linear constraints.

Average case analysis: Expected performance of an algorithm for random input (under certain distributions).

Smoothed analysis: Expected performance of an algorithm under small random perturbations of the input.

Linear programming is a problem of interest in its own right, but it is also representative of a class of geometric problems to which similar methods can be applied. Many of the references given below discuss closely related problems, and we have mentioned them in passing above.

An important related area is integer programming. Here the size of the numbers cannot be relegated to a secondary consideration. In general this problem is NP-hard, but in fixed dimension is polynomial-time solvable. See [Sch86] for further information. It may be noted that Clarkson's methods and the LP-type framework are applicable in this situation; some care with the interpretation of the primitive operations is in order, though.

We have considered only the solution of a single linear program. However, there are some situations where one might wish to solve a sequence of closely related linear programs. In this case, it may be worth the computational investment of building a data structure to facilitate fast solution of the linear programs. For results of this type see, for example, [Epp90, Mat93, Cha96, Cha98].

Finally there has been some work about optimization, where we are asked to satisfy all but at most k of the given constraints, see, e.g., [RW94, ESZ94, Mat95b, DLSS95, Cha99]. In particular, Matoušek [Mat95a] has investigated this question in the general setting of LP-type problems. Recently, Chan [Cha02] solved this problem in \mathbb{R}^2 with a randomized algorithm in expected time $O(n + k^2)$ (see this paper for the best bounds known for $d = 3, 4$).

A direction we did not touch upon here is *average analysis*, where we analyze a deterministic algorithm for random inputs [Bor87, Sma83]. Of course, the issue

here is to what extent the assumed input distribution is justified, even if the results relate to the measurements made in experiments. More recently, there has been an interesting new direction, where a simplex method is analyzed for small random perturbations of the input (smoothed analysis, [ST01]).

45.9 SOURCES AND RELATED MATERIAL

BOOKS AND SURVEYS

A good general introduction to linear programming may be found in Chvátal's book [Chv83]. A theoretical treatment is given in Schrijver's book [Sch86]. The latter is a very good source of additional references. Karp and Ramachandran [KR90] is a good source of information on models of parallel computation. See [Mat96] for a survey of derandomization techniques for computational geometry.

RELATED CHAPTERS

- [Chapter 16: Basic properties of convex polytopes](#)
 - [Chapter 20: Polytope skeletons and paths](#)
 - [Chapter 31: Computational convexity](#)
 - [Chapter 42: Parallel algorithms in geometry](#)
 - [Chapter 43: Parametric search](#)
 - [Chapter 46: Mathematical programming](#)
 - [Chapter 64: Software](#)
-

REFERENCES

- [AM96] M. Ajtai and N. Megiddo. A deterministic poly($\log \log n$)-time n -processor algorithm for linear programming in fixed dimension. *SIAM J. Comput.*, 25:1171–1195, 1996.
- [AM94] N. Alon and N. Megiddo. Parallel linear programming in fixed dimension almost surely in constant time. *J. Assoc. Comput. Mach.*, 41:422–434, 1994.
- [Ame94] N. Amenta. Helly-type theorems and generalized linear programming. *Discrete Comput. Geom.*, 12:241–261, 1994.
- [Ame96] N. Amenta. A new proof of an interesting Helly-type theorem. *Discrete Comput. Geom.*, 15:423–427, 1996.
- [AZ99] N. Amenta and G.M. Ziegler. Deformed products and maximal shadows. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemp. Math.*, pages 57–90. Amer. Math. Soc., Providence, 1999.
- [AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.
- [Bor87] K.H. Borgwardt. *The Simplex Method: A Probabilistic Analysis*. Volume 1 of *Algorithms Combin.*, Springer-Verlag, Berlin, 1987.

- [BDF⁺95] A. Broder, M.E. Dyer, A.M. Frieze, P. Raghavan, and E. Upfal. The worst case running time of the randomized simplex algorithm is exponential in the height. *Inform. Process. Lett.*, 56:79–82, 1995.
- [Cha96] T.M. Chan. Fixed-dimensional linear programming queries made easy. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 284–290, 1996.
- [Cha98] T.M. Chan. Deterministic algorithms for 2-d convex programming and 3-d online linear programming. *J. Algorithms*, 27:147–166, 1998.
- [Cha99] T.M. Chan. Geometric applications of a randomized optimization technique. *Discrete Comput. Geom.*, 22:547–567, 1999.
- [Cha02] T.M. Chan. Low-dimensional linear programming with violations. In *Proc. 43rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 570–579, 2002.
- [Chv83] V. Chvátal. *Linear Programming*. Freeman, New York, 1983.
- [Cla86a] K.L. Clarkson. Further applications of random sampling to computational geometry. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 414–423, 1986.
- [Cla86b] K.L. Clarkson. Linear programming in $O(n3^d)$ time. *Inform. Process. Lett.*, 22:21–24, 1986.
- [Cla88] K.L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 452–456, 1988.
- [Cla95] K.L. Clarkson. Las Vegas algorithms for linear and integer programming when the dimension is small. *J. Assoc. Comput. Mach.*, 42:488–499, 1995. (Improved version of [Cla88].)
- [CM96] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization in fixed dimension. *J. Algorithms*, 21:579–597, 1996.
- [DLSS95] A. Datta, H.-P. Lenhof, C. Schwarz, and M. Smid. Static and dynamic algorithms for k -point clustering problems. *J. Algorithms*, 19:474–503, 1995.
- [Den90] X. Deng. An optimal parallel algorithm for linear programming in the plane. *Inform. Process. Lett.*, 35:213–217, 1990.
- [DF89] M.E. Dyer and A.M. Frieze. A randomized algorithm for fixed-dimensional linear programming. *Math. Programming*, 44:203–212, 1989.
- [Dye83] M.E. Dyer. The complexity of vertex enumeration methods. *Math. Oper. Res.*, 8:381–402, 1983.
- [Dye84] M.E. Dyer. Linear time algorithms for two- and three-variable linear programs. *SIAM J. Comput.*, 13:31–45, 1984.
- [Dye86] M.E. Dyer. On a multidimensional search problem and its application to the Euclidean one-centre problem. *SIAM J. Comput.*, 15:725–738, 1986.
- [Dye92] M.E. Dyer. A class of convex programs with applications to computational geometry. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 9–15, 1992.
- [Dye95] M.E. Dyer. A parallel algorithm for linear programming in fixed dimension. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 345–349, 1995.
- [ESZ94] A. Efrat, M. Sharir, and A. Ziv. Computing the smallest k -enclosing circle and related problems. *Comput. Geom. Theory Appl.*, 4:119–136, 1994.
- [Epp90] D. Eppstein. Dynamic three-dimensional linear programming. *ORSA J. Comput.*, 4:360–368, 1990.

- [G  r95] B. G  rtner. A subexponential algorithm for abstract optimization problems. *SIAM J. Comput.*, 24:1018–1035, 1995.
- [G  r02] B. G  rtner. The random-facet simplex algorithm on combinatorial cubes. *Random Structures Algorithms*, 20:353–381, 2002.
- [Goo93] M.T. Goodrich. Geometric partitioning made easier, even in parallel. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 73–82, 1993.
- [GHZ98] B. G  rtner, M. Henk, and G.M. Ziegler. Randomized simplex algorithms on Klee-Minty cubes. *Combinatorica*, 18:349–371, 1998.
- [GST⁺01] B. G  rtner, J. Solymosi, F. Tschir schnitz, P. Valtr, and E. Welzl. One line and n points. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 306–315, 2001.
- [GW96] B. G  rtner and E. Welzl. Linear programming – randomization and abstract frameworks. In *Proc. 13th Annu. Sympos. Theoret. Aspects Comput. Sci.*, volume 1046 of *Lecture Notes in Comput. Sci.*, pages 669–687. Springer-Verlag, Berlin, 1996.
- [GW01] B. G  rtner and E. Welzl. A simple sampling lemma: Analysis and applications in geometric optimization. *Discrete Comput. Geom.*, 25:569–590, 2001.
- [Kal92] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 475–482, 1992.
- [Kal97] G. Kalai. Linear programming, the simplex algorithm and simple polytopes. *Math. Programming*, 79:217–233, 1997.
- [KR90] R. Karp and V. Ramachandran. Parallel algorithms for shared-memory machines. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity*, pages 869–941. Elsevier, Amsterdam, 1990.
- [KM72] V. Klee and G.J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, New York, 1972.
- [Mat93] J. Matou  ek. Linear optimization queries. *J. Algorithms*, 14:432–448, 1993.
- [Mat94] J. Matou  ek. Lower bounds for a subexponential optimization algorithm. *Random Structures Algorithms*, 5:591–607, 1994.
- [Mat95a] J. Matou  ek. On geometric optimization queries with few violated constraints. *Discrete Comput. Geom.*, 14:365–384, 1995.
- [Mat95b] J. Matou  ek. On enclosing k points by a circle. *Inform. Process. Lett.*, 53:217–221, 1995.
- [Mat96] J. Matou  ek. Derandomization in computational geometry. *J. Algorithms*, 20:545–580, 1996.
- [Meg83] N. Megiddo. Linear time algorithms for linear programming in \mathbb{R}^3 and related problems. *SIAM J. Comput.*, 12:759–776, 1983.
- [Meg84] N. Megiddo. Linear programming in linear time when dimension is fixed. *J. Assoc. Comput. Mach.*, 31:114–127, 1984.
- [Meg89] N. Megiddo. On the ball spanned by balls. *Discrete Comput. Geom.*, 4:605–610, 1989.
- [MP78] D.E. Muller and F.P. Preparata. Finding the intersection of two convex polyhedra. *Theoret. Comput. Sci.*, 7:217–236, 1978.
- [MSW96] J. Matou  ek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.
- [RW94] T. Roos and P. Widmayer. k -violation linear programming. *Inform. Process. Lett.*, 52:109–114, 1994.

- [Sch86] A. Schrijver. *Introduction to the Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.
- [Sei91] R. Seidel. Low dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.
- [Sen95] S. Sen. A deterministic poly($\log \log n$) time optimal CRCW PRAM algorithm for linear programming in fixed dimension. Technical Report 95-08, Dept. of Comput. Sci., Univ. of Newcastle, Australia, 1995.
- [Sha78] M.I. Shamos. *Computational Geometry*. Ph.D. thesis, Yale Univ., New Haven, 1978.
- [SW92] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Annu. Sympos. Theoret. Aspects Comput. Sci.*, volume 577 of *Lecture Notes in Comput. Sci.*, pages 569–579. Springer-Verlag, Berlin, 1992.
- [Sma83] S. Smale. On the average number of steps in the simplex method of linear programming. *Math. Programming*, 27:241–262, 1983.
- [ST01] D.A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the Simplex algorithm usually takes polynomial time. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 296–305, 2001.
- [Syl57] J.J. Sylvester. A question in the geometry of situation. *Quart. J. Math.*, 1:79, 1857
- [Tar86] É. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34:250–256, 1986.
- [Wel91] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Comput. Sci.*, pages 359–370. Springer-Verlag, Berlin, 1991.

46 MATHEMATICAL PROGRAMMING

Michael J. Todd

INTRODUCTION

Mathematical programming is concerned with minimizing a real-valued function of several variables, which may be either discrete or continuous, subject to equality and/or inequality constraints on other functions of the variables. Optimality conditions and computational schemes for such problems frequently rely on geometrical properties of the set of feasible solutions or subsidiary geometrical constructions. Here we consider these aspects of general nonlinear optimization problems (Section 46.1), general convex programming (Section 46.2, where we discuss the ellipsoid method and its relatives), linear programming (Section 46.3, where we consider the simplex algorithm and more recent interior-point methods), integer and combinatorial optimization (Section 46.4), and special convex programming problems (Section 46.5). The treatment here focuses mainly on methods involving geometric ideas, especially those for which global complexity estimates are known.

46.1 GENERAL NONLINEAR PROGRAMMING

Consider the problem of choosing x to

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && g_i(x) \leq 0, \quad i = 1, \dots, m, \\ & && h_j(x) = 0, \quad j = 1, \dots, p, \end{aligned} \tag{P}$$

where f and all g_i 's and h_j 's are smooth functions on real n -dimensional Euclidean space \mathbb{R}^n and x can vary over all \mathbb{R}^n . This is a *general nonlinear programming* problem.

Research on the general nonlinear programming problem seeks to characterize global or local minimizers by appropriate optimality conditions, and to compute or approximate a local minimizer or stationary point by some iterative method.

GLOSSARY

Objective function: The function f above.

Constraint functions: The functions g_i and h_j above.

Feasible point: Point in \mathbb{R}^n satisfying all constraints.

Active constraints: All equality constraints and those inequality constraints holding with equality at a given feasible point.

Global minimizer: Feasible point with objective function value at most that of any other feasible point.

Local minimizer: Feasible point with objective function value at most that of any other sufficiently close feasible point.

Optimality conditions: Conditions that are necessary or sufficient, perhaps under regularity restrictions, for a given point to be a local or global minimizer.

Stationary point: Point satisfying optimality conditions.

Lagrangian function: Function $L(x, u, v) := f(x) + u^T g(x) + v^T h(x)$.

46.1.1 OPTIMALITY CONDITIONS

These are based on Taylor approximations of the objective and constraint functions. Let \bar{x} be a feasible point. We seek conditions that are necessary or sufficient for \bar{x} to be a local minimizer. The first-order **Karush-Kuhn-Tucker conditions, necessary** under a regularity condition (such as that the gradients of all constraints active at \bar{x} are linearly independent), involve the Lagrangian function, but because of the presence of inequalities, are more general than the classical Lagrange conditions. They can be stated simply as follows: there exist multipliers \bar{u}, \bar{v} , such that

$$\begin{aligned}\nabla_x L(\bar{x}, \bar{u}, \bar{v}) &= 0, \\ \nabla_u L(\bar{x}, \bar{u}, \bar{v}) &\leq 0, \quad \bar{u} \geq 0, \quad \bar{u}^T \nabla_u L(\bar{x}, \bar{u}, \bar{v}) = 0, \\ \nabla_v L(\bar{x}, \bar{u}, \bar{v}) &= 0.\end{aligned}$$

THEOREM 46.1.1

Let \bar{x} be a feasible point, and assume the regularity condition that the gradients of all constraints active at \bar{x} are linearly independent. Then the Karush-Kuhn-Tucker conditions are necessary for \bar{x} to be a local minimizer for (P).

Second-order conditions, involving the Hessian (second derivative matrix) of the Lagrangian, are also important because of the role of curvature in nonlinear optimization. For example:

THEOREM 46.1.2

Suppose that the first-order conditions above hold, and in addition that, for all nonzero directions d with $\nabla h_j(\bar{x})^T d = 0$ for all j , $\nabla g_i(\bar{x})^T d = 0$ for all i with $\bar{u}_i > 0$, and $\nabla g_i(\bar{x})^T d \leq 0$ for all other constraints g_i active at \bar{x} , $d^T \nabla_{xx}^2 L(\bar{x}, \bar{u}, \bar{v})d > 0$. Then \bar{x} is a local minimizer. (Thus these are sufficient conditions.)

These results can be found, for example, in Chapter 12 of [NW99] or Chapter 9 of [Fle87]. Note the following special case: For unconstrained problems, $\nabla f(\bar{x}) = 0$ is necessary for \bar{x} to be a local minimizer, while this equality together with $\nabla^2 f(\bar{x})$ positive definite is sufficient.

46.1.2 ALGORITHMS

Methods to approximate stationary points or possibly local minimizers of general smooth nonlinear programming problems are often based on solving a sequence of simpler problems, using the final approximation or solution of the previous problem as a starting point for the new problem. Examples of simpler problems include un-

constrained minimization, using barrier, penalty, or (augmented) Lagrangian functions to incorporate the constraints, or a quadratic minimization subject to linear constraints, where the original nonlinear constraints are linearized and the Hessian of the quadratic objective approximates that of the Lagrangian of the original problem. (Such ***quadratic programming*** problems can be solved exactly when the objective function is convex, by extensions of methods for linear programming or other algorithms.) If the original problem is unconstrained, and we make a quadratic approximation to the function at each iteration, we recover Newton's method for optimization if the Hessian is exact, and various quasi-Newton methods if approximations are iterated.

Let us describe some typical examples of such algorithms. We state these in simplified form without worrying about important subjects like step size selection, termination criteria, or implementation details. We also omit *globalization* techniques, designed to force convergence to a stationary point or local minimizer from arbitrary starting points (not guaranteeing global minimizers, which is in general much harder!). The subscript k here refers to the iteration number, not a component.

NEWTON'S METHOD FOR UNCONSTRAINED MINIMIZATION

Given iterate x_k , calculate $\nabla f(x_k)$ and $H_k := \nabla^2 f(x_k)$. Stop if $\nabla f(x_k) = 0$ (success) or if H_k is not positive definite (failure). Otherwise, compute the direction d_k as the solution to $H_k d_k = -\nabla f(x_k)$: note that $x_k + d_k$ minimizes the Taylor approximation

$$f(x_k) + \nabla f(x_k)^T (x - x_k) + (1/2)(x - x_k)^T H_k (x - x_k).$$

Let $x_{k+1} := x_k + \alpha_k d_k$ for some step size α_k chosen so that $f(x_{k+1}) < f(x_k)$, and repeat.

BFGS METHOD FOR UNCONSTRAINED MINIMIZATION

This is a very popular quasi-Newton method. Instead of the Hessian matrix being calculated, a positive definite approximation to it is *updated* at each iteration using new information obtained about f . This method is named for Broyden, Fletcher, Goldfarb, and Shanno, who independently developed the update formula below. (More details on the BFGS and related methods can be found in [Chapter 9](#) of [DS96] or [Chapter 8](#) of [NW99].) Initially, choose H_0 , say, as some positive multiple of the identity matrix. At the k th iteration, proceed as above but with H_k the updated approximation. The step size α_k is chosen so that $f(x_{k+1}) < f(x_k)$ and so that, with $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$ and $s_k := x_{k+1} - x_k$, we have $y_k^T s_k > 0$. Then update H_k to

$$H_{k+1} := H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

(this formula maintains positive definiteness) and repeat. Note that we have $H_{k+1} s_k = y_k$, the so-called secant or quasi-Newton equation; clearly, if f is quadratic, its constant Hessian matrix satisfies this equation.

A SEQUENTIAL QUADRATIC PROGRAMMING METHOD FOR CONSTRAINED MINIMIZATION

Given the iterate x_k and estimates of the Lagrange multipliers u_k and v_k , evaluate the gradients of all functions and $\nabla_{xx}^2 L(x_k, u_k, v_k)$. Then solve the quadratic programming subproblem:

$$\begin{array}{lll} \min_d & \nabla f(x_k)^T d & + (1/2)d^T \nabla_{xx}^2 L(x_k, u_k, v_k) d \\ & g_i(x_k) & + \nabla g_i(x_k)^T d \\ & h_j(x_k) & + \nabla h_j(x_k)^T d \end{array} \begin{array}{ll} \leq 0, & \text{all } i \\ = 0, & \text{all } j \end{array}$$

to get d_k . Let $x_{k+1} := x_k + \alpha_k d_k$ for some step size α_k chosen, for example, so that the penalty function

$$f(x) + \mu \sum_i \max\{g_i(x), 0\} + \nu \sum_j |h_j(x)|$$

is reduced in moving from x_k to x_{k+1} , for suitable positive μ and ν . Replace u_k and v_k by the Lagrange multipliers for the constraints in the quadratic programming problem above, and repeat. There are also quasi-Newton versions of this method.

CONVERGENCE

Some global, local, or rate of convergence results can be established for such methods, for example:

THEOREM 46.1.3

If Newton's method is started sufficiently close to a point x_ satisfying the second-order sufficient conditions to be a local minimizer of f , then the iterates will converge to x_* and the convergence will be quadratic: $\|x_{k+1} - x_*\|/\|x_k - x_*\|^2$ remains bounded.*

(A similar result holds for the sequential quadratic programming method using an exact Hessian. See, e.g., [NW99]: Theorem 3.7 for the unconstrained and Theorem 18.4 for the constrained case.) For the quasi-Newton method, it is generally necessary to assume also that H_0 is sufficiently close to $\nabla^2 f(x_*)$, and the convergence is only superlinear: $\|x_{k+1} - x_*\|/\|x_k - x_*\|$ converges to zero. These are local convergence and rate of convergence results. For an example of a global convergence result, consider the unconstrained minimization problem. Then, assuming f is bounded below, if an algorithm of the form above has the angle between each search direction d_k and the direction of the negative gradient $-\nabla f(x_k)$ bounded away from 90° , and the step sizes are chosen appropriately, then $\nabla f(x_k)$ necessarily converges to zero, and so every limit point is a stationary point. However, no bounds on the total computation required for a prescribed precision are known in general (or to be expected lacking convexity). Vavasis [Vav91] describes what complexity results have been obtained for certain special nonconvex problems; for example, minimizing a general quadratic function subject to simple bound constraints is NP-hard, while minimizing such a function subject to lying in a ball is polynomially approximable.

46.2 CONVEX PROGRAMMING

Now we suppose that the functions f and all g_i 's are convex, and that all h_j 's are linear (affine). Then (P) is called a ***convex programming*** problem; it involves the minimization of a convex function over a convex set.

46.2.1 OPTIMALITY CONDITIONS

If all the functions involved are smooth, then the first-order conditions that are necessary (under a regularity condition) also turn out to be sufficient, not just for local but for global optimality. In other words, stationary points are global minimizers (see, e.g., Theorem 9.4.2 in [Fle87]):

THEOREM 46.2.1

Suppose \bar{x} is feasible for the convex programming problem (P), and that there exist multipliers \bar{u} and \bar{v} such that the Karush-Kuhn-Tucker conditions hold. Then \bar{x} is a global minimizer for (P).

There are also optimality conditions in the nonsmooth case, since convex functions admit subgradients (linear supports) even if they are not differentiable at a point. For a convex function k and a point \bar{x} ,

$$\partial k(\bar{x}) := \{z \mid k(x) \geq k(\bar{x}) + z^T(x - \bar{x}) \text{ for all } x\}$$

is called the ***subdifferential*** of k at \bar{x} , and it is a nonempty compact convex set whose members are called ***subgradients*** of k at \bar{x} .

THEOREM 46.2.2

Consider the (modified) Karush-Kuhn-Tucker conditions at a point \bar{x} , where the first equation is replaced by

$$0 \in \partial_x L(\bar{x}, \bar{u}, \bar{v}).$$

These conditions are sufficient for \bar{x} to be a global minimizer for (P). In the case that (P) satisfies the regularity condition that there is some feasible point \hat{x} satisfying all inequality constraints strictly, these conditions are also necessary for \bar{x} to be a local minimizer for (P).

See, e.g., Theorem 28.3 in [Roc70]. In addition, optimality conditions can be stated as saddle-point properties of the Lagrangian. Indeed, $(\bar{x}, \bar{u}, \bar{v})$ satisfies these conditions iff

$$L(x, \bar{u}, \bar{v}) \geq L(\bar{x}, \bar{u}, \bar{v}) \geq L(\bar{x}, u, v)$$

for any $x \in \mathbb{R}^n$, $u \in \mathbb{R}_+^m$, $v \in \mathbb{R}^p$. Whether the functions are smooth or not, local minimizers are always global minimizers. There is also a rich duality theory for convex programming, with many results mirroring those for linear programming. See [Roc70, RW98].

46.2.2 ALGORITHMS

As far as algorithms are concerned, in the smooth case one can again employ the general methods discussed above. Slightly stronger results are available about convergence; for example, for the unconstrained minimization of a convex function f , global convergence of the BFGS quasi-Newton method is assured for suitable step size rules, and it is no longer necessary to assume H_0 close to $\nabla^2 f(x_*)$ to obtain superlinear convergence. However, again no global estimates of the work required to attain a certain precision are known for such methods.

There are also methods designed for nonsmooth problems, such as subgradient and bundle methods. See, e.g., [Fle87, HL93a, HL93b].

LOCALIZER ALGORITHMS

On the other hand, a different class of methods for which such guarantees are available can be applied, even in the nonsmooth case. Various methods are appropriate in the case of smooth functions or the case of nonsmooth functions when the dimension n is high and the desired accuracy low. Here we will briefly describe methods for nonsmooth problems where n is small and high accuracy is required; these are based on very geometrical ideas involving *localizers*. While more general problems can be treated, suppose we wish to minimize a convex function f over the cube $G := \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$, where the variation of f over G is at most 1 (this is just a normalization condition).

GLOSSARY

ϵ -optimal point: $x \in G$ with $f(x) \leq \inf_G f + \epsilon$.

Localizer: Pair (H, z) , $H \subseteq \mathbb{R}^n$, $z \in G$, such that if $x \in G \setminus H$, $f(x) \geq f(z)$.

We mention three such methods here.

METHOD OF CENTRAL SECTIONS (MCS)

This algorithm, due to Levin [Lev65] and Newman [New65], generates a sequence $\{x_k\}$ of test points and a sequence $\{(Q_k, z_k)\}$ of localizers by the following rules. Choose $Q_0 := G$, $z_0 \in G$ arbitrary, and at the k th iteration, choose x_k as the center of gravity of Q_k and compute $f(x_k)$ and a subgradient g_k of f at x_k . If $g_k = 0$, x_k minimizes f ; in this case, stop. Otherwise, $Q_k^+ := \{x \in Q_k \mid g_k^T x \leq g_k^T x_k\}$ contains all minimizers of f over G . Set $Q_{k+1} := Q_k^+$ and let z_{k+1} be whichever of z_k and x_k has the lower function value. It is easy to see by induction that all (Q_k, z_k) 's are localizers. (For $n = 1$, this amounts to just the well-known bisection method.)

The key fact is that a substantial reduction in volume is obtained in successive localizers: $\text{vol}(Q_{k+1}) \leq (1 - 1/e)\text{vol}(Q_k)$. (Here e denotes the base of the natural logarithm.) From this it is not hard to see that an ϵ -optimal point will be found within $O(n \ln \frac{1}{\epsilon})$ iterations. This is optimal from a worst-case viewpoint—no algorithm can ask fewer questions of f (up to a constant factor) and guarantee ϵ -optimality. Unfortunately, it is not easy to find or approximate centers of gravity for general n , although Bertsimas and Vempala [BV01] provide a polynomial randomized algorithm.

ELLIPSOID METHOD (EM)

The (circumscribing) ellipsoid method due to Yudin-Nemirovskii [YN76] and Shor [Sho77] is similar, with the following variations: Q_0 is taken to be the minimum-volume ellipsoid containing G , and Q_{k+1} the minimum-volume ellipsoid containing the semiellipsoid Q_k^+ . The formulas for updating x_k and Q_k are then trivially implemented, thus removing the drawback of the MCS. However, the volume reduction is much less: $\text{vol}(Q_{k+1}) \approx (1 - [2(n+1)]^{-1})\text{vol}(Q_k)$, and this leads to a complexity bound of $O(n^2[\ln n + \ln \frac{1}{\epsilon}])$ iterations to get an ϵ -optimal point. An iteration of the ellipsoid method is shown in Figure 46.2.1.

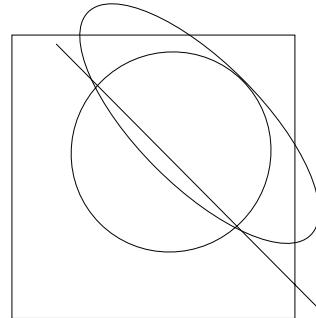


FIGURE 46.2.1
An iteration of the ellipsoid method.

METHOD OF INSCRIBED ELLIPSOIDS (MIE)

This final method, due to Tarasov, Khachiyan, and Erlikh [TKE88], chooses the localizers Q_k as in the MCS, but at each iteration takes x_k as the center of (an approximation to) the maximum-volume ellipsoid E_k contained in Q_k . It can be shown that $\text{vol}(E_{k+1}) \leq (8/9)\text{vol}(E_k)$, which leads to an $O(n \ln \frac{1}{\epsilon})$ -iteration bound. After every $O(n \ln n)$ iterations, the polytope Q_k can be enlarged slightly to one with $O(n)$ facets, so that all Q_l 's can be restricted to only $O(n \ln n)$ facets, without changing the complexity. For these polytopes, x_k can be well approximated in $O(n^{3.5+\delta})$ arithmetic operations (see [KT93]), where δ is an arbitrarily small positive number whose presence compensates for various logarithmic factors.

One last remark: In the EM (because possibly $x_k \notin G$) or when (convex) constraints are present, x_k may not be feasible. In this case, g_k is chosen so that all feasible solutions satisfy $g_k^T x \leq g_k^T x_k$, and $z_{k+1} := z_k$.

Table 46.2.1 summarizes the complexities of all three methods (see, e.g., [TKE88, KT93]).

TABLE 46.2.1 Localizer algorithms for convex programming.

ALGORITHM	COMPLEXITY
MCS	$O(n \ln[1/\epsilon])$
EM	$O(n^2[\ln n + \ln(1/\epsilon)])$
MIE	$O(n \ln[1/\epsilon])$

These methods are not practical for large n , and may not be as efficient for small dimensions as other smooth methods, since they are based on a worst-case perspective. For more efficient methods that have global complexity estimates for certain classes of convex programming problems, see Section 46.5.

46.3 LINEAR PROGRAMMING

Now we discuss the case where all functions defining (P) are linear (affine). By performing simple manipulations, we can express any such problem in standard form, where the constraints take the form of m equations in n nonnegative variables:

$$\begin{array}{ll} \min & c^T x \\ Ax & = b, \\ x & \geq 0. \end{array} \quad (\text{LP})$$

Here A is $m \times n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and the variable $x \in \mathbb{R}^n$.

Such problems are of great interest in a wide variety of areas, and their solution comprises a not insignificant fraction of all scientific computation. Problems with m of the order of 10^3 and n of the order of 10^4 are solved routinely, and larger instances can also be solved without too much difficulty in most cases. (Note the contrast with [Chapter 45](#), where it is typically assumed that n is small.) In large-scale settings, the sparsity of the matrix A (typically it has at most 5–10 nonzero entries per column) is very important, and numerical methods must exploit sparse matrix technology.

46.3.1 DUALITY

Conditions for a feasible solution to be optimal in (LP) are best stated in terms of another linear programming problem, now involving n inequalities in m variables unrestricted in sign, constructed from the same data. This is the *dual* problem (we call (LP) the *primal*), and is

$$\begin{array}{ll} \max & b^T y \\ A^T y & \leq c. \end{array} \quad (\text{LD})$$

It is easy to see that $c^T x \geq b^T y$ for any feasible solutions x for (LP) and y for (LD) (this is called *weak duality*), so that if equality holds for \bar{x} and \bar{y} they must both be optimal. Indeed, the converse holds: a feasible point \bar{x} is optimal in (LP) if and only if there is a \bar{y} feasible in (LD) with $c^T \bar{x} = b^T \bar{y}$. This is *strong duality*. There is also a nice geometric way of looking at the optimal dual solution. Let us write \bar{s} for the vector $c - A^T \bar{y}$ (we shall see more of this dual *slack vector* later). Then $c^T \bar{x} = b^T \bar{y}$ implies that $\bar{x}^T \bar{s} = 0$, and feasibility in the dual implies that $\bar{s} \geq 0$. Then the dual constraints $A^T \bar{y} + \bar{s} = c$ can be viewed as expressing the gradient of the objective function of the primal problem as a linear combination of the gradients of the constraints that are active at \bar{x} (since \bar{s}_j is nonzero only if \bar{x}_j is zero), where we can take arbitrary multiples of the equality constraint gradients but only nonnegative multiples of the inequality constraint gradients.

Besides its use in proving optimality, the dual problem has important economic interpretations in many instances, and its solution provides crucial sensitivity infor-

mation about the effects of changes in the data on the solution and optimal value of the primal problem. In addition, it is much exploited in solution techniques for linear programming.

46.3.2 ALGORITHMS

Once again, algorithms for more general problems, i.e., from convex programming, can be applied to these more special problems. Indeed, the first polynomial-time algorithm for linear programming (for the case where all the data are integer-valued) was constructed by Khachiyan [Kha79] based on the ellipsoid method. This result has great theoretical significance (even more for combinatorial optimization, see below), but little practical importance, since the method is very inefficient even for small problems.

Efficient methods for solving (LP) rely on two geometrical ways of looking at its feasible region. This is a polyhedron, typically of dimension $d := n - m$, with at most n facets. The simplex method of Dantzig [Dan63] relies on the combinatorial properties of this polyhedron, in particular on its 1-skeleton, while interior-point methods originating from Karmarkar's projective algorithm [Kar84] (or the earlier method of Dikin [Dik67]) use a more analytic view relying on a Riemannian metric at points of the interior of the feasible region that reflects its local geometry. (There are also methods based on completely different geometric ideas, for example those of Welzl and coauthors (see, e.g., [GW96] and the references therein), and special methods appropriate for low-dimensional problems: see [Chapter 45](#).)

The differences among the various classes of methods are summarized in Table 46.3.1.

TABLE 46.3.1 Classes of linear programming methods.

METHOD	GEOMETRY	ITERATES	TERMINATION
Ellipsoid	convex	arbitrary points	in the limit
Simplex	combinatorial	vertices	finite
Interior-point	analytic	interior points	in the limit

The “efficiencies” of the methods are shown in Table 46.3.2 (incidentally, this shows the difference between practical and theoretical efficiency).

TABLE 46.3.2 Complexities of linear programming methods.

METHOD	WORST-CASE COMPLEXITY	EXPECTED COMPLEXITY	PRACTICAL
Ellipsoid	polynomial	polynomial	no
Simplex	exponential/super-polynomial?	polynomial	yes
Interior-point	polynomial	polynomial	yes

Articles on the practical efficiency of algorithms for large-scale linear programming problems may be found in [ORJ94].

46.3.3 SIMPLEX METHODS

Given a vertex of the feasible region, a variant of the simplex method proceeds from vertex to vertex along edges while improving the objective function, stopping either at an indication that the objective function is unbounded below or at an optimal solution. (In order to find an initial vertex, the same process is applied to an artificial problem.) The choice of a particular edge to follow is called a pivot rule, and leads to a particular variant. For example, we can choose the edge yielding the maximal decrease in the objective function per unit length (in the Euclidean norm) moved; this is the steepest-edge rule. Various rules can be shown to terminate finitely, even for degenerate problems when the “edge” followed has zero length and merely leads to a different representation of the same vertex (this may happen when the feasible region is not a simple polyhedron). However, no rule is currently known for which the number of steps taken is always bounded by a polynomial in the dimensions m and n or the input size (total number of bits to represent the data, assumed integer-valued). Indeed, for many rules, examples have shown the worst-case complexity to be exponential (see Amenta and Ziegler [AZ98]), although Kalai [Kal92] has described a randomized rule whose expected number of steps in the worst case is subexponential, though superpolynomial; see [Section 45.2](#). Nevertheless, the simplex method using appropriate pivot rules works very well in practice, requiring only a small number of iterations (possibly $O(m \ln n)$) for typical problems. To explain this gap, some studies have shown that the expected number of steps for certain (deterministic) simplex variants, when applied to a random problem generated by a suitable probabilistic distribution, is polynomial; see Borgwardt [Bor86] and the references therein.

The number of steps taken by a simplex variant starting at an arbitrary vertex is clearly related to the *diameter* of the associated polyhedron of feasible solutions; this is the largest, over all pairs of vertices, of the smallest number of edges required to go from one to the other. The famous *Hirsch conjecture* states that this is at most $n - d$ for polytopes (bounded polyhedra) of dimension d with at most n facets. It is known to be true for $d \leq 3$ and $n - d \leq 5$, and for certain classes of polytopes arising in special applications, but the general case remains open; see [KK87]. A nice result [Nad89] shows that the conjecture is true for a polytope that is the convex hull of a set of $(0,1)$ -vectors, as arises in combinatorial optimization. Note that knowing that the diameter of a polytope is small does not immediately lead to an efficient simplex method for optimizing a linear function over that polytope.

46.3.4 INTERIOR-POINT METHODS

Algorithms for linear programming of this type generate a sequence of iterates for (LP) that satisfy all the equations and satisfy all the inequalities strictly (we call such points *strictly feasible*). At such an iterate, one can move in a direction of steepest descent restricted to the affine space $\{x \mid Ax = b\}$, but if the strictly feasible iterate is very close to the boundary of the feasible region, it is possible that only a very short step can be taken.

The first method of this kind, due to Dikin [Dik67], applied an affine transformation (or *scaling*) to move the current point to the vector e of ones. A steepest descent step was taken in the scaled space, and the resulting point was transformed

back to yield the next iterate. This very simple algorithm performs surprisingly well, but polynomial convergence has not been established. Dikin's paper was not well-known, but was rediscovered soon after Karmarkar's independent work [Kar84], which used a projective transformation and achieved a polynomial time bound (given in Table 46.3.3). The proof used an ingenious potential function, of the form

$$\phi(x, \zeta) := n \ln(c^T x - \zeta) - \sum_j \ln x_j,$$

with ζ a lower bound on the optimal value of (P), that is closely related to classical barrier functions used in nonlinear programming since the '50s (see, e.g., [Fle87]).

Instead of performing the transformations, we can view the search directions in the original space as given by steepest descent directions for a certain function with respect to a Riemannian metric. At the strictly feasible point x , the length of a displacement d_x is defined as $\|d_x\|_x := (d_x^T X^{-2} d_x)^{1/2}$, where X is the diagonal matrix containing the components of x . Thus the metric is defined by the matrix X^{-2} , which is the Hessian of the *barrier function*

$$F(x) := - \sum_j \ln x_j.$$

The Dikin, or affine-scaling, direction is the steepest direction for the objective function $c^T x$ in the null space of A with respect to this norm, whereas the Karmarkar, or projective-scaling, direction is a similar steepest descent direction for a certain linear combination of $c^T x$ and the barrier $F(x)$. This metric, and in the second case the presence of the barrier, steers the direction away from approaching the boundary of the feasible region too closely prematurely. We will not describe Karmarkar's algorithm in detail, since it is quite complicated and the method has been superseded by those in the next subsection.

46.3.5 PRIMAL-DUAL METHODS

Recent attention has focused on primal-dual methods, which iterate both primal and dual strictly feasible points. It is helpful to write the dual with explicit slack variables as

$$\begin{array}{lll} \max & b^T y \\ & A^T y + s \geq 0. \end{array} \quad (\text{LD})$$

GLOSSARY

Barrier function: Convex function defined on the relative interior of the feasible region, tending to infinity as the boundary is approached.

Interior point: Point satisfying all inequality constraints strictly.

Strictly feasible point: Interior point satisfying all constraints.

e : Vector of ones in \mathbb{R}^n .

X (resp. S): Diagonal matrix containing the components of x (resp. s).

Central path: Set of pairs of strictly feasible points x and (y, s) with $XSe = \mu e$ for some $\mu > 0$.

Neighborhoods of central path: Sets of strictly feasible pairs with $XSe - \mu e$ suitably bounded.

Primal-dual potential function: $\rho \ln x^T s + F(x) + F(s)$ for $\rho \geq n + \sqrt{n}$, defined for strictly feasible pairs.

Noting that for feasible solutions we have

$$c^T x - b^T y = (A^T y + s)^T x - (Ax)^T y = x^T s \geq 0$$

(a short proof of weak duality), we see that optimality conditions for (LP) and (LD) can be written as

$$\begin{aligned} A^T y + s &= c & (s \geq 0), \\ Ax &= b & (x \geq 0), \\ XSe &= 0. \end{aligned} \quad (\text{OC})$$

Indeed, the simplex method can be viewed as seeking to satisfy (OC) by maintaining all conditions except the nonnegativity of s at each iteration.

On the other hand, (OC) can be viewed as $m + 2n$ mildly nonlinear equations in $m + 2n$ variables, with nonnegativity restrictions as side constraints, and then Newton's method for nonlinear equations seems appropriate. If we start with strictly feasible solutions \hat{x} and (\hat{y}, \hat{s}) ($\hat{x} > 0$ and $\hat{s} > 0$) and compute the Newton step for (OC), we can take a partial step to maintain strict feasibility for the next iterates (*damped* Newton step). Indeed, we can take a damped Newton step for a *perturbed* system with the zero right-hand side replaced by $\gamma \hat{\mu} e$, where $\hat{\mu} := \hat{x}^T \hat{s} / n$ is the current duality gap divided by n and $0 \leq \gamma \leq 1$, to encourage the iterates to remain positive while taking a large step.

This primal-dual framework appears to be rather far from the steepest descent view of primal-only interior-point methods. However, the direction for x can be seen to be a steepest descent direction for $c^T x + \gamma \hat{\mu} F(x)$ with respect to the norm $(d_x^T \hat{X}^{-1} \hat{S} d_x)^{1/2}$, rather than the expected $(d_x^T \hat{X}^{-2} d_x)^{1/2}$. Similarly, the directions for y and s are the steepest descent directions for $-b^T y + \gamma \hat{\mu} F(s)$ with respect to the norm (on just d_s) $(d_s^T \hat{S}^{-1} \hat{X} d_s)^{1/2}$, rather than the expected $(d_s^T \hat{S}^{-2} d_s)^{1/2}$. Note that these two norms are dual, as is appropriate since x and s lie in dual spaces (the duality gap is $x^T s = \langle s, x \rangle$), and that they are scalar multiples of the norms given by the Hessians of the barrier functions if $\hat{X} \hat{S} e = \hat{\mu} e$. (They can be viewed as the closest dual norms to the latter.)

Several algorithms are based on taking such damped perturbed Newton steps. We next describe a generic algorithm of this type.

A GENERIC PRIMAL-DUAL INTERIOR-POINT METHOD

Suppose we are given the strictly feasible points x_k and (y_k, s_k) at the k th iteration. Let $\mu_k := x_k^T s_k / n$ and choose $\gamma \in [0, 1]$. Solve for the directions d_x , d_y , and d_s from

$$\begin{aligned} A^T d_y + d_s &= 0, \\ Ad_x &= 0, \\ S_k d_x + X_k d_s &= \gamma \mu_k e - X_k S_k e, \end{aligned}$$

where X_k and S_k denote the diagonal matrices corresponding to x_k and s_k . Then set $x_{k+1} := x_k + \alpha_P d_x$ and $(y_{k+1}, s_{k+1}) := (y_k, s_k) + \alpha_D (d_y, d_s)$, where the positive step sizes α_P and α_D are chosen so that the new iterates are strictly feasible.

A practical version of this method (which takes long steps to try to converge fast, but which lacks a polynomial bound) might choose $\gamma = 1/n$ and α_P and α_D as .999 of the maximum values that would maintain primal and dual feasibility, respectively.

Most theoretically attractive primal-dual methods try to stay close to the *central path*. For each $\mu > 0$, there is a unique pair x and (y, s) of strictly feasible solutions with $XSe = \mu e$, and the set of all these forms the central path, which leads (as $\mu \rightarrow 0$) to the set of optimal solutions. Some methods maintain $\|XSe/\mu - e\| \leq \beta$ with either the ℓ_2 - or ℓ_∞ -norm, and others only require $XSe \geq (1-\beta)(x^T s/n)e$, for some $0 < \beta < 1$, for all iterates. When the ℓ_2 -neighborhood is used, we get a close path-following method, whereas the other restrictions yield loose path-following methods. For example, one can choose $\alpha_P = \alpha_D = 1$ and $\gamma \in [0, 1]$ as small as possible to maintain $\|XSe/\mu - e\|_2 \leq 1/4$ in the generic algorithm above. (One can show that $\gamma \leq 1 - 1/(4\sqrt{n})$ with this choice.) The first close path-following method was due to Renegar [Ren88] and operated in the dual space alone; this was also the first method with the improved complexity of $O(\sqrt{n} \ln \frac{1}{\epsilon})$ steps to attain ϵ -optimality. Also, primal-dual methods can be based on a primal-dual potential function and require no neighborhood. All the methods described in this subsection, with the exception of the affine-scaling method and the practical primal-dual algorithm outlined above, are polynomial. However, the methods with the better complexity bounds (see Table 46.3.3) seem not to be as useful practically, as they tend to force short steps.

INFEASIBLE-INTERIOR-POINT METHODS

Finally, as with the simplex method, initial feasible (here, strictly feasible) solutions are rarely known. However, a damped perturbed Newton step can also be taken from iterates \hat{x} and (\hat{y}, \hat{s}) with $x > 0$ and $\hat{s} > 0$ (*infeasible interior points*) even if the equations in (LP) and (LD) are violated. These infeasible-interior-point methods strive for feasibility and optimality at the same time, and are the basis for most interior-point codes. Polynomial bounds have recently been obtained (see below). Another approach [YTM94] applies a “feasible” method to an artificial homogeneous self-dual problem and either generates optimal solutions or proves primal or dual infeasibility in the limit.

COMPLEXITY OF INTERIOR-POINT METHODS

The types of algorithms and their iteration complexities to obtain ϵ -optimality given suitable starting points are given in Table 46.3.3: see, e.g., [Wri96, Ye97]. Note that all except the last two assume that a feasible solution is at hand.

All these methods require $O(n^3)$ arithmetical operations at each iteration (although an acceleration trick can reduce this to $O(n^{2.5})$ operations on average for some methods), assuming dense linear algebra is used. To get the complexity of solving exactly a linear programming problem with integer data of length L , replace $(1/\epsilon)$ in Table 46.3.3 by L ; we then get polynomial complexity. Roughly, from an ϵ -optimal solution with $\epsilon = 2^{-O(L)}$ we can obtain an exact solution.

 TABLE 46.3.3 Complexities of interior-point methods.

ALGORITHM		COMPLEXITY
Primal	affine-scaling projective-scaling	— $O(n \ln(1/\epsilon))$
Primal-dual	close path-following	$O(\sqrt{n} \ln(1/\epsilon))$
	loose path-following	$O(n \ln(1/\epsilon))$
	potential-reduction	$O(\sqrt{n} \ln(1/\epsilon))$
	infeasible-interior-point	$O(n \ln(1/\epsilon))$
	homogeneous self-dual	$O(\sqrt{n} \ln(1/\epsilon))$

46.4 INTEGER AND COMBINATORIAL OPTIMIZATION

Now suppose we wish to optimize a linear function subject to linear constraints and the requirement that the variables be integer. Such a problem is called an integer (linear) programming problem. For simplicity, we assume that all variables must be 0 or 1. Then the problem can be formulated as:

$$\begin{array}{lll} \max & c^T x \\ & Ax \leq b, \\ 0 \leq x \leq e, & & \\ & x \text{ integer.} & \end{array} \quad (\text{IP})$$

(Recall that e denotes the vector of ones.) Clearly, such problems (sometimes with only some of the variables required to be integer) arise from applications like those leading to linear programming instances—for example, production of certain items (e.g., aircraft carriers) is essentially discrete. But it is important to realize the modeling possibilities of (0,1) variables: they can be used to represent either-or situations, such as whether to build a new factory, invest in a new product, etc., or to model such nonconvexities as setup costs and minimum batch sizes.

There are also inherently combinatorial problems that can be represented in the form (IP). Consider for example the notorious *traveling salesman problem* [LLRS85], which arises in routing and sequencing applications. Here it is desired to visit each of n cities exactly once, starting and finishing at the same city, at minimum cost. By introducing variables x_{ij} , $1 \leq i < j \leq n$, equal to 1 if the salesman goes directly from city i to city j or vice versa, we can model the problem as minimizing a linear function over a finite (but large!) set of (0,1)-vectors of length $n(n - 1)/2$. It is not hard to see that this can be written in the form (IP) by introducing appropriate constraints. Indeed, this can be done in several ways.

46.4.1 OPTIMALITY CONDITIONS AND DUALITY

Solving integer programming problems is NP-hard in general, although it is polynomial when the dimension is fixed and for certain special problems (see [Len83, Sch86, GLS88]). One reason for the difficulty is that it is far from trivial to check whether a given feasible solution is optimal. Very often, a heuristic method or

routine within an algorithm will produce a very good or optimal solution quickly, but proving that it is (near-)optimal is very time-consuming.

The main tool for establishing the quality of a feasible solution for (IP) is linear programming. Note that the optimal value of (IP) is bounded above by that of its *linear programming relaxation*

$$\begin{aligned} \max \quad & c^T x \\ & Ax \leq b, \\ & 0 \leq x \leq e. \end{aligned} \tag{LR}$$

There are some problems (mostly network-flow-related) for which the linear programming relaxation has only integer vertices. Thus solving (LR) by, say, the simplex method will solve (IP). If this is not the case, then the optimal value of (LR) (or of its linear programming dual) provides information about the quality of a given feasible solution of (IP), and its optimal solution may help to locate a good integer solution. In any case, it is clear that a “tight” formulation (so that (LR) is “closer” to (IP), and the *integrality gap*—the gap between their optimal values—is smaller) will be very helpful in (approximately) solving (IP).

There are also specialized duals for integer programming involving subadditive functions, but they do not seem to be as useful computationally.

46.4.2 ALGORITHMS

Certain integer or combinatorial optimization problems (for example, network flow problems, see [CCPS98]) can be solved very efficiently. This can be done either using the simplex method (as indicated above, for some such problems it suffices to solve the linear programming relaxation, and in some cases a polynomial bound on the number of iterations has been proved) or specialized combinatorial methods (e.g., for certain graph, network, and matroid problems), but these usually have little geometric content. For harder problems two general approaches are possible.

BRANCH-AND-BOUND

The first is an implicit enumeration scheme. Suppose we solve the linear programming relaxation (LR). If the solution is integer, we are done; otherwise, we obtain a bound on the optimal value, and by choosing a variable whose optimal value is fractional, we construct two other problems, in one of which the variable is restricted to be 0 and in the other, 1. If we continued branching in this way, we would eventually reach a tree with 2^n leaves, with each corresponding to a particular (0,1) assignment for all the variables. Two things may allow us to construct only a very much smaller enumeration tree. First, if the optimal solution for the relaxation at some node (corresponding to a partial assignment of the variables) is integer, we need not perform any more branching from this node. If this solution is the best seen so far, we record it. Second, if the linear programming relaxation at a node is either infeasible or has optimal value below that of some known integer solution (obtained by a heuristic or from examination of other parts of the tree), the node need not be considered further. If neither of these cases holds, we choose a fractional variable and branch as above, thus creating two new nodes. Since we keep track of the best integer solution found, when all nodes have been considered, the current best solution solves (IP), or, if there is none, (IP) is infeasible. Note that

the efficiency of this technique depends on the tightness of the formulation, since this helps both to generate integer solutions to linear programming relaxations and to give good bounds allowing nodes to be rejected as above.

CUTTING-PLANE METHODS

The other approach tries to generate ever tighter linear programming relaxations. Note that if we could optimize the objective function over the convex hull C of the finitely many feasible solutions of (IP), we would obtain the optimal solution directly. Since this convex hull is a polytope (such a polytope is called a **(0,1)-polytope**), it can be expressed as the solution set to a set of linear inequalities, so that there is *some* linear programming relaxation that allows the integer programming problem to be solved using linear programming methods. The problem is that we do not know all of these inequalities, and even if we could describe them completely (as we can, say, for the matching problem in a nonbipartite graph) there might be exponentially many. Thus we would like to generate them “on the fly.”

One algorithm that does not require all the inequalities to be available explicitly is the ellipsoid method. Indeed, there is a precise sense in which, if we can determine in polynomial time whether a given point is within the convex hull C and, if not, produce a separating hyperplane, then we can optimize over C in polynomial time using the ellipsoid method, and vice versa; see [GLS88]. Of course, this is not a practical method, but it has significant theoretical consequences in combinatorial optimization (showing that some problems are polynomially solvable and, conversely, that others are NP-hard), and gives a strong indication that similar constraint-generation methods using efficient linear programming methods can be practically useful for many problems. We need to be able to reoptimize a linear programming problem easily after slight modifications (addition of constraints, fixing of variables), and for this simplex methods seem preferable at the present time to interior-point algorithms. We also need a way to generate a “good” set of constraints to add when we discover that our current linear programming relaxation is not tight enough. These constraints should be valid for all feasible solutions to (IP), but violated by the optimal solution of the current relaxation, so they are called *cutting planes*.

This approach to the study of combinatorial problems is called:

POLYHEDRAL COMBINATORICS

Given a collection of subsets of some finite ground set (e.g., each subset could be the set of edges of a Hamiltonian circuit of a given graph), one can consider the corresponding collection of their (0,1) incidence vectors, and the convex hull C of these vectors. This is a (0,1)-polytope. Questions about the combinatorial system can then be reduced to questions about the resulting polytope; in particular, optimizing over the set of subsets often becomes a linear programming problem over C . It is therefore of interest to obtain complete or partial descriptions of the linear inequalities defining C . These can then be used in developing special algorithms or within the context of the cutting-plane methods above. Thus much recent research has been devoted to finding deep valid inequalities or, if possible, facets of C for practically interesting problems, and also to developing separation techniques to identify a member of a class of facets that is violated by a given point. A very

interesting recent result testifies to the complexity of the convex hull of feasible solutions for at least one class of hard problems: Billera and Sarangarajan [BS96] show that *any* (0,1)-polytope is a traveling salesman polytope. Another indication of the complexity of such facetial descriptions is the following. For the case of 8 cities, the traveling salesman polytope has been completely described [CJR91]—it has 194,187 facets! No such description is known for the case of 9 cities. Recall that (0,1)-polytopes have small diameter—they satisfy the Hirsch conjecture. Unfortunately this is of little comfort when we cannot give a complete facetial description of them; moreover, they are often very degenerate, so the simplex method might take many pivot steps going nowhere. Further discussion and references on polyhedral combinatorics can be found in [Chapter 7](#) of this Handbook.

REMARKS

The two techniques above can be combined, so that cutting planes are added as long as one can be found, and then enumeration resorted to if necessary. If possible, the inequalities found at one node of the tree should also be valid for other nodes in the tree, so we always have tight relaxations. Using these ideas, very large combinatorial and integer optimization problems have been solved. We note that recently a traveling salesman problem with 15,112 cities was solved to optimality by Applegate, Bixby, Chvátal, and Cook: see <http://www.math.princeton.edu/tsp/d15sol/index.html>.

46.5 SPECIAL CONVEX PROGRAMMING PROBLEMS

In Section 46.2, we described the ellipsoid method and its relatives, but noted that they are not practical for large (or even medium-sized) convex programming problems. On the other hand, Section 46.3 gave methods that are efficient even for very large linear programming problems. Here we address the possibility of solving efficiently large convex optimization problems falling into nice classes, using methods with global complexity estimates.

The first such class is that of (convex) quadratic programming problems. As we noted in Section 46.1, these arise as subproblems in methods for general smooth nonlinear programming; they are also important in their own right, with applications in portfolio analysis and constrained data-fitting. The optimality conditions for such problems are very similar to those for linear programming: another linear term is added to one of the sets of linear equations in (OC). Thus it is not too surprising that extensions of linear programming methods, of both the simplex and interior-point persuasions, have been devised for quadratic programming. The complexity of the latter kinds of algorithm is the same as for linear programming. There are also direct methods for quadratic programming, described in [Fle87, GMW81, NW99].

46.5.1 INTERIOR-POINT METHODS FOR NONLINEAR PROGRAMMING

There has recently been great interest in extending interior-point methods to certain classes of convex programming problems, maintaining a polynomial complexity

bound. The most extensive work has been done by Nesterov and Nemirovskii, and appears in their book [NN94]; see also [BTN01, Ren01].

Any convex programming problem can be rewritten, by adding one or two variables, in the form of either

$$\min\{c^T x \mid x \in G\}$$

or

$$\min\{c^T x \mid Ax = b, x \in K\},$$

where G and K are closed convex sets in \mathbb{R}^n with nonempty interiors, and K is a cone. The second formulation, said to be in conical form, is clearly closely related to the standard form linear programming problem (LP), and allows a nice statement of the dual as

$$\max\{b^T y \mid A^T y + s = c, s \in K^*\},$$

where K^* is the dual cone $\{s \in \mathbb{R}^n \mid x^T s \geq 0 \text{ for all } x \in K\}$. Weak duality is immediate, while strong duality holds if, for example, there are feasible solutions to both problems with x and s in the interiors of their respective cones.

GLOSSARY

Self-concordant barrier: Barrier function satisfying certain smoothness conditions allowing efficient interior-point methods.

Self-scaled barrier: Self-concordant barrier satisfying further conditions allowing greater freedom in efficient methods.

Semidefinite programming: Convex optimization with constraints that certain symmetric matrices be positive semidefinite.

Symmetric cones: Self-dual homogeneous cones.

The view of interior-point methods given in Section 46.3 leads us to consider steepest descent steps for a strictly feasible point (in $\text{int } G$ or $\{x \in \text{int } K \mid Ax = b\}$) with respect to the norm defined by the Hessian of a barrier function for the set G or K . Nesterov and Nemirovskii devise path-following and potential-reduction methods with polynomial complexity for the problems above as long as a barrier function F for G or K can be found satisfying certain key properties. These properties, roughly convexity and Lipschitz continuity of F and its second derivative with respect to the norm defined by the second derivative itself, define the set of *self-concordant barriers*. An attractive feature of these functions is that Newton's method performs well (in a precise sense) when applied to their minimization, not just locally but also globally. One of the Lipschitz constants, the *parameter* ν of the barrier, takes the place of the number of linear inequalities n in linear programming in complexity bounds. Thus the number of iterations of their algorithms to attain ϵ -optimality is $O(\nu \ln \frac{1}{\epsilon})$ or $O(\sqrt{\nu} \ln \frac{1}{\epsilon})$.

To develop symmetric primal-dual methods like those used in interior-point codes for LP, we need to consider the problem in conical form and require a further condition on the barrier: it should be *self-scaled* [NT97]. We will not give the precise definition here, but we note that one of its consequences is that, for every $x \in \text{int } K$ and $s \in \text{int } K^*$, there is a unique $w \in \text{int } K$ with $F''(w)x = s$. Then the

generic primal-dual interior-point method is exactly as in the linear programming case, except that the last equation defining the directions becomes

$$F''(w_k)d_x + d_s = -s_k - \gamma\mu_k F'(x_k),$$

where $F''(w_k)x_k = s_k$.

It turns out that a cone admits a self-scaled barrier exactly when it is symmetric, i.e., self-dual (isomorphic to its dual) and homogeneous (there is an automorphism of the cone taking any point of its interior into any other). Such cones have been studied in depth since the 1960s. This connection was made by Güler [Gü96].

Barriers of these types have strong geometric consequences. For example, the ball of radius 1 centered at a strictly feasible point and defined by the norm based on the Hessian of a self-concordant barrier at that point lies completely within G or K . Moreover, if such a barrier for G has a minimizer x^* (the *analytic center* of G), then G not only contains the ball of radius 1 centered at x^* with respect to this norm, but is also contained in the ball of radius $1 + 3\nu$, where ν is the parameter of the barrier. For self-scaled barriers, we can find even larger inscribed sets, corresponding to ℓ_∞ -balls. The fact that G or K can be thus “well-fitted” by simple convex sets gives an indication of the reason that efficient algorithms can be found to optimize over them.

Because of the general theory it is obviously desirable to construct self-concordant or self-scaled barriers for convex sets and cones that can be used to model important optimization problems. Here is a list of some special cases:

1. Quadratic constraints: Let $g_i, i = 1, \dots, m$, be convex quadratic functions (note that these include linear functions). Then

$$F(x) := - \sum_{i=1}^m \ln[-g_i(x)]$$

is a self-concordant barrier with parameter m for

$$G := \{x \mid g_i(x) \leq 0, i = 1, \dots, m\}.$$

2. Second-order, or Lorentz, or “ice-cream” cone: The function

$$F(x) := -\ln(x_0^2 - \sum_{j=1}^n x_j^2)$$

is a self-concordant (and self-scaled) barrier with parameter 2 for the cone

$$K := \{x \in \mathbb{R}^{n+1} \mid x_0 \geq (\sum_{j=1}^n x_j^2)^{1/2}\}.$$

3. Symmetric positive semidefinite matrices: The function

$$F(X) := -\ln \det X$$

is a self-concordant (and self-scaled) barrier with parameter n for the cone K of symmetric positive semidefinite matrices of order n .

This last example is particularly interesting, because several important applications (including obtaining bounds in hard combinatorial optimization or control theory problems, and eigenvalue optimization) can be modeled as optimizing a linear function over the cone of symmetric positive semidefinite matrices, subject to linear constraints; such problems are the subject of semidefinite programming.

46.6 SOURCES AND RELATED MATERIAL

BOOKS AND SURVEYS

Besides the references cited above, the reader can consult the following sources for more background and further citations of the literature. In general, the chapters in [NRT89] provide state-of-the-art surveys of various fields of mathematical programming as of 1989. For nonlinear programming, see also [Fle87, GMW81, HL93a, HL93b, NW99, Vav91]. The book [NY83] contains almost all you want to know about localizer algorithms and the (informational) complexity of convex programming, although it predates the MIE (discussed in [NN94]); see also [BGT81]. For linear programming, the classic texts are [Dan63, Chv83, Sch86]; more recent surveys on path-following methods and potential-reduction algorithms are [Gon92, Tod96], and the books [BTN01, Ren01, Van96, Wri96, Ye97] are recommended. For combinatorial optimization, see [CCPS98] and consult the chapters on optimization, convex polytopes, and polyhedral combinatorics in [GLL95]. The application of interior-point methods to convex programming is discussed in [BTN01, dH93, NN94, Ren01]. Semidefinite programming and more general conic programming problems are covered in [WSV00].

RELATED CHAPTERS

- [Chapter 7: Lattice points and lattice polytopes](#)
 - [Chapter 16: Basic properties of convex polytopes](#)
 - [Chapter 20: Polytope skeletons and paths](#)
 - [Chapter 31: Computational convexity](#)
 - [Chapter 45: Linear programming](#)
-

REFERENCES

- [AZ98] N. Amenta and G.M. Ziegler. Deformed products and maximal shadows of polytopes. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, pages 57–90. Amer. Math. Soc., Providence, 1998.
- [BGT81] R.G. Bland, D. Goldfarb, and M.J. Todd. The ellipsoid method: a survey. *Oper. Res.*, 29:1039–1091, 1981.
- [Bor86] K.H. Borgwardt. *The Simplex Method: A Probabilistic Analysis*. Springer-Verlag, Berlin, 1986.
- [BS96] L.J. Billera and A. Sarangarajan. All 0-1 polytopes are traveling salesman polytopes. *Combinatorica*, 16:175–188, 1996.
- [BTN01] A. Ben-Tal and A.S. Nemirovski. *Lectures on Modern Convex Optimization*. SIAM, Philadelphia, 2001.
- [BV01] D. Bertsimas and S. Vempala. Solving convex programs by random walks. Manuscript, Laboratory of Computer Science, MIT, 2001.

- [CCPS98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, New York, 1998.
- [Chv83] V. Chvátal. *Linear Programming*. Freeman, San Francisco, 1983.
- [CJR91] T. Christof, M. Junger, and G. Reinelt. A complete description of the traveling salesman polytope on 8 nodes. *Oper. Res. Lett.*, 10:497–500, 1991.
- [Dan63] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.
- [Dik67] I.I. Dikin. Iterative solution of problems of linear and quadratic programming. *Soviet Math. Dokl.*, 8:674–675, 1967.
- [DS96] J.E. Dennis, Jr. and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, volume 16 of *Classics Appl. Math.* SIAM, Philadelphia, 1996. Corrected reprint of the 1983 original.
- [Fle87] R. Fletcher. *Practical Methods of Optimization*. Wiley, New York, 1987.
- [GGL95] R.L. Graham, M. Grötschel, and L. Lovász. *Handbook of Combinatorics*. North-Holland, Amsterdam, 1995.
- [GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, New York, 1988.
- [GMW81] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, New York, 1981.
- [Gon92] C.C. Gonzaga. Path following methods for linear programming. *SIAM Rev.*, 34:167–227, 1992.
- [Gül96] O. Güler. Barrier functions in interior-point methods. *Math. Oper. Res.*, 21:860–885, 1996.
- [GW96] B. Gärtner and E. Welzl. Linear programming — randomization and abstract frameworks. In *Proc. 13th Annu. Sympos. Theoret. Aspects Comput. Sci.*, volume 1046 of *Lecture Notes in Comput. Sci.*, pages 669–687. Springer-Verlag, Berlin, 1996.
- [dH93] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming*. Kluwer, Dordrecht, 1993.
- [HL93a] J.B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms I. Fundamentals*. Springer-Verlag, Berlin, 1993.
- [HL93b] J.B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms II. Advanced Theory and Bundle Methods*. Springer-Verlag, Berlin, 1993.
- [Kal92] G. Kalai. A subexponential randomized simplex algorithm. In *Proc. 24th Annu. ACM Symp. Theory Comput.*, pages 475–482, 1992.
- [Kar84] N.K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [Kha79] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Dokl.*, 20:191–194, 1979.
- [KK87] V. Klee and P. Kleinschmidt. The d -step conjecture and its relatives. *Math. Oper. Res.*, 12:718–755, 1987.
- [KT93] L.G. Khachiyan and M.J. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Math. Programming*, 61:137–159, 1993.
- [Len83] H.W. Lenstra, Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8:538–548, 1983.
- [Lev65] A.Yu. Levin. On an algorithm for the minimization of convex functions. *Soviet Math. Dokl.*, 6:286–290, 1965.
- [LLRS85] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The Traveling Salesman Problem*. Wiley, New York, 1985.
- [Nad89] D. Naddef. The Hirsch conjecture is true for $(0,1)$ -polytopes. *Math. Programming*, 45:109–110, 1989.

- [New65] D.J. Newman. Location of the maximum on unimodal surfaces. *J. Assoc. Comput. Mach.*, 12:395–398, 1965.
- [NN94] Yu.E. Nesterov and A.S. Nemirovski. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM, Philadelphia, 1994.
- [NRT89] G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd. *Optimization*, Volume 1 of *Handbooks Oper. Res. Management Sci.*, North-Holland, New York, 1989.
- [NT97] Yu.E. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*, 22:1–42, 1997.
- [NW99] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, New York, 1999.
- [NY83] A.S. Nemirovski and D.B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, 1983.
- [ORJ94] ORSA *J. Comput.*, 6:1–34, 1994.
- [Ren88] J. Renegar. A polynomial-time algorithm based on Newton’s method for linear programming. *Math. Programming*, 40:59–93, 1988.
- [Ren01] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. SIAM, Philadelphia, 2001.
- [Roc70] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
- [RW98] R.T. Rockafellar and R.J.-B. Wets. *Variational Analysis*. Springer-Verlag, Berlin, 1998.
- [Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.
- [Sho77] N.Z. Shor. Cut-off method with space extension in convex programming problems. *Cybernetics*, 13:94–96, 1977.
- [TKE88] S.P. Tarasov, L.G. Khachiyan, and I.I. Erlikh. The method of inscribed ellipsoids. *Soviet Math. Dokl.*, 37:226–230, 1988.
- [Tod96] M.J. Todd. Potential-reduction methods in mathematical programming. *Math. Programming*, 76:3–45, 1996.
- [Van96] R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer, Boston, 1996.
- [Vav91] S.A. Vavasis. *Nonlinear Optimization: Complexity Issues*. Oxford University Press, New York, 1991.
- [Wri96] S. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1996.
- [WSV00] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook on Semidefinite Programming*. Kluwer, Boston, 2000.
- [Ye97] Y. Ye. *Interior Point Algorithms: Theory and Analysis*. Wiley, New York, 1997.
- [YN76] D.B. Yudin and A.S. Nemirovski. Informational complexity and efficient methods for the solution of convex extremal problems. *Matekon*, 13:3–25, 1976.
- [YTM94] Y. Ye, M.J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$ -iteration homogeneous and self-dual linear programming algorithm. *Math. Oper. Res.*, 19:53–67, 1994.

47 ALGORITHMIC MOTION PLANNING

Micha Sharir

INTRODUCTION

Motion planning is a fundamental problem in robotics. It comes in a variety of forms, but the simplest version is as follows. We are given a robot system B , which may consist of several rigid objects attached to each other through various joints, hinges, and links, or moving independently, and a 2D or 3D environment V cluttered with obstacles. We assume that the shape and location of the obstacles and the shape of B are known to the planning system. Given an initial placement Z_1 and a final placement Z_2 of B , we wish to determine whether there exists a collision-avoiding motion of B from Z_1 to Z_2 , and, if so, to plan such a motion. In this simplified and purely geometric setup, we ignore issues such as incomplete information, nonholonomic constraints, control issues related to inaccuracies in sensing and motion, nonstationary obstacles, optimality of the planned motion, and so on.

Since the early 1980s, motion planning has been an intensive area of study in robotics and computational geometry. In this chapter we will focus on *algorithmic motion planning*, emphasizing theoretical algorithmic analysis of the problem and seeking worst-case asymptotic bounds, and only mention briefly practical heuristic approaches to the problem. The majority of this chapter is devoted to the simplified version of motion planning, as stated above. Section 47.1 presents general techniques and lower bounds. Section 47.2 considers efficient solutions to a variety of specific moving systems with a small number of degrees of freedom. These efficient solutions exploit various sophisticated methods in computational and combinatorial geometry related to arrangements of curves and surfaces ([Chapter 24](#)). Section 47.3 then briefly discusses various extensions of the motion planning problem, incorporating uncertainty, moving obstacles, etc. We conclude in Section 47.4 with a brief review of Davenport-Schinzel sequences, a combinatorial structure that plays an important role in many motion planning and other geometric algorithms.

47.1 GENERAL TECHNIQUES AND LOWER BOUNDS

GLOSSARY

Some of the terms defined here are also defined in [Chapter 48](#).

Robot B : A mechanical system consisting of one or more rigid bodies, possibly connected by various joints and hinges.

Physical space (workspace): The 2D or 3D environment in which the robot

moves.

Placement: The portion of physical space occupied by the robot at some instant.

Degrees of freedom k : The number of real parameters that determine the robot B 's placements. Each placement can be represented as a point in \mathbb{R}^k .

Free placement: A placement at which the robot is disjoint from the obstacles.

Semifree placement: A placement at which the robot does not meet the interior of any obstacle (but may be in contact with some obstacles).

Configuration space \mathcal{C} : A portion of k -space (where k is the number of degrees of freedom of B) that represents all possible robot placements; the coordinates of any point in this space specify the corresponding placement.

Expanded obstacle / C-obstacle / forbidden region: For an obstacle O , this is the portion O^* of configuration space consisting of placements at which the robot intersects (collides with) O .

Free configuration space \mathcal{F} : The subset of configuration space consisting of free placements of the robot: $\mathcal{F} = \mathcal{C} \setminus \bigcup_O O^*$. (In the literature, this usually also includes semifree placements. In that case, \mathcal{F} is the complement of the union of the *interiors* of the expanded obstacles.)

Contact surface: For an obstacle feature a (corner, edge, face, etc.) and for a feature b of the robot, this is the locus in \mathcal{C} of placements at which a and b are in contact with each other. In most applications, these surfaces are semialgebraic sets of constant description complexity (see definitions below).

Collision-free motion of B : A path contained in \mathcal{F} . Any two placements of B that can be reached from each other via a collision-free path must lie in the same (arcwise-)connected component of \mathcal{F} .

Arrangement $\mathcal{A}(\Sigma)$: The decomposition of k -space into cells of various dimensions, induced by a collection Σ of surfaces in \mathbb{R}^k . Each cell is a maximal connected portion of the intersection of some fixed subcollection of surfaces that does not meet any other surface. See [Chapter 24](#). Since a collision-free motion should not cross any contact surface, \mathcal{F} is the union of some of the cells of $\mathcal{A}(\Sigma)$, where Σ is the collection of contact surfaces.

Semialgebraic set: A subset of \mathbb{R}^k defined by a Boolean combination of polynomial equalities and inequalities in the k coordinates. See [Section 33.2](#).

Constant description complexity: Said of a semialgebraic set if it is defined by a constant number of polynomial equalities and inequalities of constant maximum degree (where the number of variables is also assumed to be constant).

Example. Let B be a rigid polygon with k edges, moving in a planar polygonal environment V with n edges. The system has three degrees of freedom, (x, y, θ) , where (x, y) are the coordinates of some reference point on B , and θ is the orientation of B . Each contact surface is the locus of placements where some vertex of B touches some edge of V , or some edge of B touches some vertex of V . There are $2kn$ contact surfaces, and if we replace θ by $\tan \frac{\theta}{2}$, then each contact surface becomes a portion of some algebraic surface of degree at most 4, bounded by a constant number of algebraic arcs, each of degree at most 2.

47.1.1 GENERAL SOLUTIONS

GLOSSARY

Cylindrical algebraic decomposition of \mathcal{F} : A recursive decomposition of \mathcal{C} into cylindrical-like cells originally proposed by Collins [Col75]. Over each cell of the decomposition, each of the polynomials involved in the definition of \mathcal{F} has a fixed sign (positive, negative, or zero), implying that \mathcal{F} is the union of some of the cells of this decomposition. See [Section 33.5](#) for further details.

Connectivity graph: A graph whose nodes are the (free) cells of a decomposition of \mathcal{F} and whose arcs connect pairs of adjacent cells.

Roadmap \mathcal{R} : A network of one-dimensional curves within \mathcal{F} , having the properties that (i) it *preserves the connectivity* of \mathcal{F} , in the sense that the portion of \mathcal{R} within each connected component of \mathcal{F} is (nonempty and) connected; and (ii) it is **reachable**, in the sense that there is a simple procedure to move from any free placement of the robot to a placement on \mathcal{R} ; we denote the mapping resulting from this procedure by $\phi_{\mathcal{R}}$.

Retraction of \mathcal{F} onto \mathcal{R} : A continuous mapping of \mathcal{F} onto \mathcal{R} that is the identity on \mathcal{R} . The roadmap mapping $\phi_{\mathcal{R}}$ is usually a retraction. When this is the case, we note that for any path ψ within \mathcal{F} , represented as a continuous mapping $\psi : [0, 1] \rightarrow \mathcal{F}$, $\phi_{\mathcal{R}} \circ \psi$ is a path within \mathcal{R} , and, concatenating to it the motions from $\psi(0)$ and $\psi(1)$ to \mathcal{R} , we see that there is a collision-free motion of B between two placements Z_1, Z_2 iff there is a path within \mathcal{R} between $\phi_{\mathcal{R}}(Z_1)$ and $\phi_{\mathcal{R}}(Z_2)$.

Silhouette: The set of critical points of a mapping; see [Section 33.6](#).

CELL DECOMPOSITION

\mathcal{F} is a semialgebraic set in \mathbb{R}^k . Applying Collins's cylindrical algebraic decomposition results in a collection of cells whose total complexity is $O((nd)^{3^k})$, where d is the maximum algebraic degree of the polynomials defining the contact surfaces; the decomposition can be constructed within a similar time bound. If the coordinate axes are generic, then we can also compute all pairs of cells of \mathcal{F} that are **adjacent** to each other (i.e., cells whose closures (within \mathcal{F}) overlap), and store this information in the form of a connectivity graph. It is then easy to search for a collision-free path through this graph, if one exists, between the (cell containing the) initial robot placement and the (cell containing the) final placement. This leads to a doubly-exponential general solution for the motion planning problem:

THEOREM 47.1.1 Cylindrical Cell Decomposition [SS83]

Any motion planning problem, with k degrees of freedom, for which the contact surfaces are defined by a total of n polynomials of maximum degree d , can be solved by Collins's cylindrical algebraic decomposition, in randomized expected time $O((nd)^{3^k})$.

(The randomization is needed only to choose a generic direction for the coordinate axes.)

ROADMAPS

A more recent and improved solution is given in [Can87, BPR00] based on the notion of a *roadmap* \mathcal{R} , a network of one-dimensional curves within (the closure of) \mathcal{F} , having properties defined in the glossary above. Once such a roadmap \mathcal{R} has been constructed, any motion planning instance reduces to path searching within \mathcal{R} , which is easy to do. \mathcal{R} is constructed recursively, as follows. One projects \mathcal{F} onto some generic 2-plane, and computes the silhouette of \mathcal{F} under this projection. Next, the critical values of the projection of the silhouette on some line are found, and a roadmap is constructed recursively within each slice of \mathcal{F} at each of these critical values. The resulting “sub-roadmaps” are then merged with the silhouette, to obtain the desired \mathcal{R} .

The original algorithm of Canny relies heavily on the polynomials defining \mathcal{F} being in general position, and on the availability of a generic plane of projection. This algorithm runs in $n^k(\log n)d^{O(k^4)}$ deterministic time, and in $n^k(\log n)d^{O(k^2)}$ expected randomized time. Recent work [BPR00] addresses and overcomes the general position issue, and produces a roadmap for any semialgebraic set; the running time of this solution is $n^{k+1}d^{O(k^2)}$.

If we ignore the dependence on the degree d , the algorithm of Canny is close to optimal in the worst case, assuming that some representation of the entire \mathcal{F} has to be output, since there are easy examples where the free configuration space consists of $\Omega(n^k)$ connected components.

THEOREM 47.1.2 *Roadmap Algorithm* [Can87]

Any motion planning problem, as in the preceding theorem, in general position can be solved by the roadmap technique in $n^k(\log n)d^{O(k^4)}$ deterministic time, and in $n^k(\log n)d^{O(k^2)}$ expected randomized time.

47.1.2 LOWER BOUNDS

The upper bounds for both general solutions are (at least) exponential in k (but are polynomial in the other parameters when k is fixed). This raises the issue of calibrating the complexity of the problem when k can be arbitrarily large.

THEOREM 47.1.3 *Lower Bounds*

The motion planning problem, with arbitrarily many degrees of freedom, is PSPACE-hard for the instances of: (a) coordinated motion of many rectangular boxes along a rectangular floor [HSS84]; (b) motion planning of a planar mechanical linkage with many links [HJW84]; and (c) motion planning for a multi-arm robot in a 3-dimensional polyhedral environment [Rei87].

All these results can also be found in the collection [HSS87]. There are also many NP-hardness results for other systems; see, e.g., [HJW85].

Facing these findings, we can either approach the general problem with heuristic and approximate schemes, or attack specific problems with small values of k , with the goal of obtaining solutions better than those yielded by the general techniques. We will mostly survey here the latter approach, and mention toward the end what has been achieved by the first approach.

47.2 MOTION PLANNING WITH A SMALL NUMBER OF DEGREES OF FREEDOM

In this main section of the chapter, we review solutions to a variety of specific motion planning problems, most of which have 2 or 3 degrees of freedom. Exploiting the special structure of these problems leads to solutions that are more efficient than the general methods described above.

GLOSSARY

Jordan arc/curve: The image of the closed unit interval under a continuous bijective mapping into the plane. A closed Jordan curve is the image of the unit circle under a similar mapping, and an unbounded Jordan curve is an image of the open unit interval (or of the entire real line) that separates the plane.

Randomized algorithm: An algorithm that applies internal randomization (“coin-flips”). We consider here algorithms that always terminate, and produce the correct output, but whose running time is a random variable that depends on the internal coin-flips. We will state upper bounds on the expectation of the running time (the **randomized expected time**) of such an algorithm, which hold for any input. See [Chapter 40](#).

Minkowski sum: For two planar (or spatial) sets A and B , their Minkowski sum, or pointwise vector addition, is the set $A \oplus B = \{x + y \mid x \in A, y \in B\}$.

General position: The input to a geometric problem is said to be in general position if no nontrivial algebraic identity with integer coefficients holds among the parameters that specify the input (assuming the input is not overspecified). For example: no three input points should be collinear, no four points cocircular, no three lines concurrent, etc.

Convex distance function: A convex region B that contains the origin in its interior induces a convex distance function d_B defined by

$$d_B(p, q) = \min \{\lambda \mid q \in p \oplus \lambda B\}.$$

If B is centrally symmetric with respect to the origin then d_B is a metric whose unit ball is B .

B -Voronoi diagram: For a set S of sites, and a convex region B as above, the B -Voronoi diagram $\text{Vor}_B(S)$ of S is a decomposition of space into Voronoi cells $V(s)$, for $s \in S$, such that

$$V(s) = \{p \mid d_B(p, s) \leq d_B(p, s') \text{ for all } s' \in S\}.$$

Here $d_B(p, s) = \min_{q \in s} d_B(p, q)$.

$\alpha(n)$: The extremely slowly-growing inverse Ackermann function; see Section 47.4.

Contact segment: The locus of (not necessarily free) placements of a polygon B translating in a planar polygonal workspace, at each of which either some specific vertex of B touches some specific obstacle edge, or vice-versa.

Contact curve: A generalization of “contact segment” to the locus of (not necessarily free) placements of a more general robot system B , assuming that B has only two degrees of freedom, where some specific feature of B makes contact with some specific obstacle feature.

47.2.1 TWO DEGREES OF FREEDOM

A TRANSLATING POLYGON IN 2D

This is a system with two degrees of freedom (translations in the x and y directions).

A CONVEX POLYGON

Suppose first the translating polygon B is a *convex* k -gon, and there are m convex polygonal obstacles, A_1, \dots, A_m , with pairwise disjoint interiors, having a total of n edges. The region of configuration space where B collides with A_i is the *Minkowski sum*

$$K_i = A_i \oplus (-B) = \{x - y \mid x \in A_i, y \in B\}.$$

The free configuration space is the complement of $\bigcup_{i=1}^m K_i$. Assuming general position, one can show:

THEOREM 47.2.1 [KLPS86]

- (a) *Each K_i is a convex polygon, with $n_i + k$ edges, where n_i is the number of edges of A_i .*
- (b) *For each $i \neq j$, the boundaries of K_i and K_j intersect in at most two points. (This also holds when the A_i 's and B are not polygons.)*
- (c) *Given a collection of planar regions K_1, \dots, K_m , each enclosed by a closed Jordan curve, such that any pair of the bounding curves intersects at most twice, then the boundary of the union $\bigcup_{i=1}^m K_i$ consists of at most $6m - 12$ maximal connected portions of the boundaries of the K_i 's, provided $m \geq 3$, and this bound is tight in the worst case.*

These properties, combined with several algorithmic techniques [KLPS86, MMP⁺91, dBDS95], imply:

THEOREM 47.2.2

- (a) *The free configuration space for a translating convex polygon, as above, is a polygonal region with at most $6m - 12$ convex vertices and $N = \sum_{i=1}^m (n_i + k) = n + km$ nonconvex vertices.*
- (b) *\mathcal{F} can be computed in deterministic time $O(N \log^2 n)$, or in randomized expected time $O(N \log n)$.*

If the robot is translating in a convex room with n walls, then the complexity of the free space is $O(n)$ and it can be computed in $O(n + k)$ time.

AN ARBITRARY POLYGON

Suppose next that B is an arbitrary polygonal region with k edges. Let A be the union of all obstacles, which is another polygonal region with n edges. As above, the free configuration space is the complement of the Minkowski sum

$$K = A \oplus (-B) = \{x - y \mid x \in A, y \in B\}.$$

K is again a polygonal region, but, in this case, its maximum possible complexity is $\Theta(k^2n^2)$ (see, e.g., [AFH02]), so computing it might be considerably more expensive than in the convex case. Efficient practical algorithms for the exact computation of Minkowski sums in this case (together with their implementation) are described in [AFH02].

A single face suffices. If the initial placement Z of B is given, then we do not have to compute the entire (complement of) K ; it suffices to compute the connected component f of the complement of K that contains Z , because no other placement is reachable from Z via a collision-free motion.

Let Σ be the collection of all contact segments; there are $2kn$ such segments. The desired component f is the face of $\mathcal{A}(\Sigma)$ that contains Z . Using the theory of *Davenport-Schinzel sequences* (Section 47.4), one can show that the maximum possible combinatorial complexity of a single face in a two-dimensional arrangement of N segments is $\Theta(N\alpha(N))$. A more careful analysis [HCA⁺95], combined with the algorithmic techniques of [CEG⁺93, GSS89], shows:

THEOREM 47.2.3

- (a) *The maximum combinatorial complexity of a single face in the arrangement of contact segments for the case of an arbitrary translating polygon is $\Theta(kn\alpha(k))$ (this improvement is significant only when $k \ll n$).*
- (b) *Such a face can be computed in deterministic time $O(kn\alpha(k)\log^2 n)$ [GSS89], or in randomized expected time $O(kn\alpha(k)\log n)$ [CEG⁺93].*

VORONOI DIAGRAMS

Another approach to motion planning for a translating *convex* object B , is via generalized *Voronoi diagrams* (see [Chapter 23](#)), based on the convex distance function $d_B(p, q)$. This function effectively places B centered at p and expands it until it hits q . The scaling factor at this moment is the d_B -distance from p to q (if B is a unit disk, d_B is the Euclidean distance). d_B satisfies the triangle inequality, and is thus “almost” a metric, except that it is not symmetric in general; it is symmetric iff B is centrally symmetric with respect to the point of reference.

Using this distance function d_B , a B -Voronoi diagram $\text{Vor}_B(\mathcal{S})$ of \mathcal{S} may be defined for a set \mathcal{S} of m pairwise disjoint obstacles. See [LS87a, Yap87a].

THEOREM 47.2.4

Assuming that each of B and the obstacles in \mathcal{S} has constant description complexity, and that they are in general position, the B -Voronoi diagram has $O(m)$ complexity, and can be computed in $O(m \log m)$ time (in an appropriate model of computation). If B and the obstacles are convex polygons, as above, then the complexity of $\text{Vor}_B(\mathcal{S})$ is $O(N)$ and it can be computed in time $O(N \log m)$.

One can show that if Z_1 and Z_2 are two free placements of B , then there exists a collision-free motion from Z_1 to Z_2 if and only if there exists a collision-free motion of B where its center moves only along the edges of $\text{Vor}_B(\mathcal{S})$, between two corresponding placements W_1, W_2 , where W_i , for $i = 1, 2$, is the placement obtained by pushing B from the placement Z_i away from its d_B -nearest obstacle, until it becomes equally nearest to two or more obstacles (so that its center lies on an edge of $\text{Vor}_B(\mathcal{S})$).

Thus motion planning of B reduces to a path-searching in the one-dimensional network of edges of $\text{Vor}_B(\mathcal{S})$. This technique is called the *retraction technique*, and can be regarded as a special case of the general roadmap algorithm. The resulting motions have “high clearance,” and so are safer than arbitrary motions, because they stay equally nearest to at least two obstacles.

THEOREM 47.2.5

The motion planning problem for a convex object B translating amidst m convex and pairwise disjoint obstacles can be solved in $O(m \log m)$ time, by constructing and searching in the B -Voronoi diagram of the obstacles, assuming that B and the obstacles have constant description complexity each. If B and the obstacles are convex polygons, then the same technique yields an $O(N \log m)$ solution, where $N = n + km$ is as above.

THE GENERAL MOTION PLANNING PROBLEM WITH TWO DEGREES OF FREEDOM

If B is any system with two degrees of freedom, its configuration space is 2D, and, for simplicity, let us think of it as the plane (spaces that are topologically more complex can be decomposed into a constant number of “planar” patches). We construct a collection Σ of contact curves, which, under reasonable assumptions concerning B and the obstacles, are each an algebraic Jordan arc or curve of some fixed maximum degree b . In particular, each pair of contact curves will intersect in at most some constant number, $s \leq b^2$, of points.

As above, it suffices to compute the single face of $\mathcal{A}(\Sigma)$ that contains the initial placement of B . The theory of Davenport-Schinzel sequences implies that the complexity of such a face is $O(\lambda_{s+2}(n))$, where $\lambda_{s+2}(n)$ is the maximum length of an $(n, s+2)$ -Davenport-Schinzel sequence (Section 47.4), which is slightly super-linear in n when s is fixed.

The face in question can be computed in deterministic time $O(\lambda_{s+2}(n) \log^2 n)$, using a fairly involved divide-and-conquer technique based on line-sweeping; see [GSS89] and Section 24.5. (Some slight improvements in the running time have been subsequently obtained.) Using randomized incremental (or divide-and-conquer) techniques, the face can be computed in randomized expected $O(\lambda_{s+2}(n) \log n)$ time [CEG⁺93, SA95].

THEOREM 47.2.6 *see* [GSS89, CEG⁺93, dBDS95]

Under the above assumptions, the general motion planning problem for systems with two degrees of freedom can be solved in deterministic time $O(\lambda_{s+2}(n) \log^2 n)$, or in $O(\lambda_{s+2}(n) \log n)$ randomized expected time.

47.2.2 THREE DEGREES OF FREEDOM

A ROD IN A PLANAR POLYGONAL ENVIRONMENT

We next pass to systems with three degrees of freedom. Perhaps the simplest instance of such a system is the case of a line segment B (“rod,” “ladder,” “pipe”) moving (translating and rotating) in a planar polygonal environment with n edges. The maximum combinatorial complexity of the free configuration space \mathcal{F} of B is $\Theta(n^2)$ (recall that the naive bound for systems with three degrees of freedom is $O(n^3)$). A cell-decomposition representation of \mathcal{F} can be constructed in (deterministic) $O(n^2 \log n)$ time [LS87b]. Several alternative near-quadratic algorithms have also been developed, including one based on constructing a Voronoi diagram in \mathcal{F} [OSY87]. A worst-case optimal algorithm, with running time $O(n^2)$, has been given in [Veg90].

An $\Omega(n^2)$ lower bound for this problem has been established in [KO88]. It exhibits a polygonal environment with n edges and two free placements of B that are reachable from each other. However, any free motion between them requires $\Omega(n^2)$ “elementary moves,” that is, the specification of any such motion requires $\Omega(n^2)$ complexity. This is a fairly strong lower bound, since it does not rely on lower bounding the complexity of the free configuration space (or of a single connected component thereof); after all, it is not clear why a motion planning algorithm should have to produce a full description of the whole free space (or of a single component).

THEOREM 47.2.7

Motion planning for a rod moving in a polygonal environment bounded by n edges can be performed in $O(n^2)$ time. There are instances where any collision-free motion of the rod between two specified placements requires $\Omega(n^2)$ “elementary moves.”

A CONVEX POLYGON IN A PLANAR POLYGONAL ENVIRONMENT

Here B is a convex k -gon, free to move (translate and rotate) in an arbitrary polygonal environment bounded by n edges. The free configuration space is 3D, and there are at most $2kn$ contact surfaces, of maximum degree 4. The naive bound on the complexity of \mathcal{F} is $O((kn)^3)$ (attained if B is nonconvex), but, using Davenport-Schinzel sequences, one can show that the complexity of \mathcal{F} is only $O(kn\lambda_6(kn))$. Geometrically, a vertex of \mathcal{F} is a semifree placement of B at which it makes simultaneously three obstacle contacts. The above bound implies that the number of such *critical placements* is only slightly super-quadratic (and not cubic) in kn .

Computing \mathcal{F} in time close to this bound has proven more difficult, and only recently has a complete solution, running in $O(kn\lambda_6(kn) \log kn)$ time and constructing the entire \mathcal{F} , been attained [AAS99]. Previous solutions that were either incomplete with the same time bound or complete and somewhat more expensive are given in [KS90, HS96, KST97].

Another approach was given in [CK93]. It computes the Delaunay triangulation of the obstacles under the distance function d_B , when the orientation of B is fixed, and then traces the discrete combinatorial changes in the diagram as the orientation

varies. The number of changes was shown to be $O(k^4 n \lambda_3(n))$. Using this structure, the algorithm of [CK93] produces a high-clearance motion of B between any two specified placements, in time $O(k^4 n \lambda_3(n) \log n)$.

Since all these algorithms are fairly complicated, one might consider in practice an alternative approximate scheme, proposed in [AFK⁺92]. This scheme, originally formulated for a rectangle, discretizes the orientation of B , solves the translational motion planning for B at each of the discrete orientations, and finds those placements of B at which it can rotate (without translating) between two successive orientations. This scheme works very well in practice.

THEOREM 47.2.8

Motion planning for a k -sided convex polygon, translating and rotating in a planar polygonal environment bounded by n edges, can be performed in $O(kn\lambda_6(kn) \log kn)$ or $O(k^4 n \lambda_3(n) \log n)$ time.

EXTREMAL PLACEMENTS

A related problem is to find the largest free placement of B in the given polygonal environment. This has applications in manufacturing, where one wants to cut out copies of B that are as large as possible from a sheet of some material.

If only translations are allowed, the B -Voronoi diagram can be used to find the largest free homothetic copy of B . If general rigid motions are allowed, the technique of [CK93] computes the largest free similar copy of B in time $O(k^4 n \lambda_3(n) \log n)$. An alternative technique is given in [AAS98], with randomized expected running time $O(kn\lambda_6(kn) \log^4 kn)$. Both bounds are nearly quadratic in n . See also earlier work on this problem in [ST94].

Finally, we mention the special case where the polygonal environment is the interior of a convex n -gon. This is simpler to analyze. The number of free critical placements of (similar copies of) B , at which B makes simultaneously four obstacle contacts, is $O(kn^2)$ [AAS98], and they can all be computed in $O(kn^2 \log n)$ time. If only translations are allowed, this problem can easily be expressed as a linear program, and can be solved in $O(n + k)$ time [ST94].

THEOREM 47.2.9

The largest similar placement of a k -sided convex polygon in a planar polygonal environment bounded by n edges can be computed in randomized expected time $O(kn\lambda_6(kn) \log^4 kn)$ or in deterministic time $O(k^4 n \lambda_3(n) \log n)$. When the environment is the interior of an n -sided convex polygon, the running time improves to $O(kn^2 \log n)$, and to $O(n + k)$ if only translations are allowed.

A NONCONVEX POLYGON

Next we consider the case where B is an arbitrary polygonal region (not necessarily connected), translating and rotating in a polygonal environment bounded by n edges, as above. Here one can show that the maximum complexity of \mathcal{F} is $\Theta((kn)^3)$. Using standard techniques, \mathcal{F} can be constructed in $\Theta((kn)^3 \log kn)$ time, and algorithms with this running time bound have been implemented; see, e.g., [ABF89]. However, as in the purely translational case, it suffices to construct the connected component of \mathcal{F} containing the initial placement of B . The general result, stated

below, for systems with three degrees of freedom, implies that the complexity of such a component is only near-quadratic in kn . A special-purpose algorithm that computes the component in time $O((kn)^{2+\epsilon})$ is given in [HS96]. A more general algorithm with a similar running time bound is reported below. An earlier work considered the case where B is an L-shaped object moving amid n point obstacles [HOS92]. Motion planning can be performed in this case in time $O(n^2 \log^2 n)$.

THEOREM 47.2.10

Motion planning for an arbitrary k -sided polygon, translating and rotating in a planar polygonal environment bounded by n edges, can be performed in time $O((kn)^{2+\epsilon})$, for any $\epsilon > 0$. If the polygon is L-shaped and the obstacles are points, the running time improves to $O(n^2 \log^2 n)$.

A TRANSLATING POLYTOPE IN A 3D POLYHEDRAL ENVIRONMENT

Another interesting motion planning problem with three degrees of freedom involves a polytope B , with a total of k vertices, edges, and facets, translating amidst polyhedral obstacles in \mathbb{R}^3 , with a total of n vertices, edges, and faces. The contact surfaces in this case are planar polygons, composed of a total of $O(kn)$ triangles in 3-space.

Without additional assumptions, the complexity of \mathcal{F} can be $\Theta((kn)^3)$ in the worst case. However, the complexity of a single component is only $O((kn)^2 \log kn)$. Such a component can be constructed in $O((kn)^{2+\epsilon})$ time, for any $\epsilon > 0$ [AS94].

If B is a convex polytope, and the obstacles consist of m convex polyhedra, with pairwise disjoint interiors and with a total of n faces, the complexity of the entire \mathcal{F} is $O(kmn \log m)$ and it can be constructed in $O(kmn \log^2 m)$ time [AS97]. (Note that, in analogy with the two-dimensional case, \mathcal{F} is the complement of the union of the Minkowski sums $A_i \oplus (-B)$, where A_i are the given obstacles. The above-cited bound is about the complexity and construction of such a union.) An earlier study [HY98] considered the case where B is a box, and obtained an $O(n^2 \alpha(n))$ bound for the complexity of \mathcal{F} .

THEOREM 47.2.11

Translational motion planning for an arbitrary polytope with k facets, in an arbitrary 3D polyhedral environment bounded by n facets, can be performed in time $O((kn)^{2+\epsilon})$, for any $\epsilon > 0$. If B is a convex polytope, and there are m convex pairwise disjoint obstacles with a total of n facets, then the motion planning can be performed in $O(kmn \log^2 m)$ time.

A BALL IN A 3D POLYHEDRAL ENVIRONMENT

Let B be a ball moving in 3D amidst polyhedral obstacles with a total of n vertices, edges, and faces. The complexity of the entire \mathcal{F} is $O(n^{2+\epsilon})$, for any $\epsilon > 0$ [AS00a]. Note that, for the special case of line obstacles, the expanded obstacles are congruent (infinite) cylinders, and \mathcal{F} is the complement of their union.

THEOREM 47.2.12

Motion planning for a ball in an arbitrary 3D polyhedral environment bounded by n facets can be performed in time $O(n^{2+\epsilon})$, for any $\epsilon > 0$.

3D B -VORONOI DIAGRAMS

A more powerful approach to translational motion planning in three dimensions is via B -Voronoi diagrams, defined in three dimensions in full analogy to the two-dimensional case mentioned above. The goal is to establish a near-quadratic bound for the complexity of such a diagram. This would yield near-quadratic algorithms for planning the motion of the moving body B , for planning a high-clearance motion, and for finding largest homothetic free placements of B . The analysis of B -Voronoi diagrams is considerably more difficult in 3-space, and there are only a few instances where a near-quadratic complexity bound is known. One instance is for the case where B is a translating convex polytope with $O(1)$ facets in a 3D polyhedral environment [KS02b]; the complexity of the diagram in this case is $O(n^{2+\epsilon})$. If the obstacles are lines or line segments, the complexity is $O(n^2\alpha(n)\log n)$ [CKS⁺98, KS02b].

The case where B is a ball appears to be more challenging. Even for the special case where the obstacles are lines, no near-quadratic bounds are known. However, if the obstacles are n lines with a constant number of orientations, the B -diagram has complexity $O(n^{2+\epsilon})$ [KS02a].

THE GENERAL MOTION PLANNING PROBLEM WITH THREE DEGREES OF FREEDOM

The last several instances were special cases of the general motion planning problem with three degrees of freedom. In abstract terms, we have a collection Σ of N contact surfaces in \mathbb{R}^3 , where these surfaces are assumed to be (patches of) algebraic surfaces of constant maximum degree. The free configuration space consists of some cells of the arrangement $\mathcal{A}(\Sigma)$, and a single connected component of \mathcal{F} is just a single cell in that arrangement.

Inspecting the preceding cases, a unifying observation is that while the maximum complexity of the entire \mathcal{F} can be $\Theta(N^3)$, the complexity of a single component is invariably only near-quadratic in N . This was recently shown in [HS95a] to hold in general: the combinatorial complexity of a single cell of $\mathcal{A}(\Sigma)$ is $O(N^{2+\epsilon})$, for any $\epsilon > 0$, where the constant of proportionality depends on ϵ and on the maximum degree of the surfaces; cf. Section 24.5.

A general-purpose algorithm for computing a single cell in such an arrangement was recently given in [SS97]. It runs in randomized expected time $O(N^{2+\epsilon})$, for any $\epsilon > 0$, and is based on *vertical decompositions* in such arrangements (see Section 24.3.2).

THEOREM 47.2.13

An arbitrary motion planning problem with three degrees of freedom, involving N contact surface patches, each of constant description complexity, can be solved in time $O(N^{2+\epsilon})$, for any $\epsilon > 0$.

47.2.3 OTHER PROBLEMS WITH FEW DEGREES OF FREEDOM

MORE DEGREES OF FREEDOM

The general motion planning problem for systems with d degrees of freedom, for $d \geq 4$, calls for estimating the complexity of a single cell in the d -dimensional arrangement of the appropriate contact surfaces, and for efficient algorithms for constructing such a cell. A recent result [Bas03] shows that the complexity of such a cell in a d -dimensional arrangement of n surfaces of constant description complexity is $O(n^{d-1+\epsilon})$, for any $\epsilon > 0$, where the constant of proportionality depends on d , ϵ , and the maximum degree of the polynomials defining the surfaces.

In contrast, computing such a cell within a comparable time bound remains an open problem.

COORDINATED MOTION PLANNING

Another class of motion planning problems involves coordinated motion planning of several independently moving systems. Conceptually, this situation can be handled as just another special case of the general problem: Consider all the moving objects as a single system, with $k = \sum_{i=1}^t k_i$ degrees of freedom, where t is the number of moving objects, and k_i is the number of degrees of freedom of the i th object. However, k will generally be too large, and the problem then will be more difficult to tackle.

A better approach is as follows [SS91]. Let B_1, \dots, B_t be the given independent objects. For each $i = 1, \dots, t$, construct the free configuration space $\mathcal{F}^{(i)}$ for B_i alone (ignoring the presence of all other moving objects). The actual free configuration space \mathcal{F} is a subset of $\prod_{i=1}^t \mathcal{F}^{(i)}$. Suppose we have managed to decompose each $\mathcal{F}^{(i)}$ into subcells of constant description complexity. Then \mathcal{F} is a subset of the union of Cartesian products of the form $c_1 \times c_2 \times \dots \times c_t$, where c_i is a subcell of $\mathcal{F}^{(i)}$.

We next compute the portion of \mathcal{F} within each such product. Each such subproblem can be intuitively interpreted as the coordinated motion planning of our objects, where each moves within a small portion of space, amidst only a constant number of nearby obstacles; so these subproblems are much easier to solve. Moreover, in typical cases, for most products $P = c_1 \times c_2 \times \dots \times c_t$ the problem is trivial, because P represents situations where the moving objects are far from one another, and so cannot interact at all, meaning that $\mathcal{F} \cap P = P$. The number of subproblems that really need to be solved will be relatively small.

The connectivity graph that represents \mathcal{F} is also relatively easy to construct. Its nodes are the connected components of the intersections of \mathcal{F} with each of the above cell products P , and two nodes are connected to each other if they are adjacent in the overall \mathcal{F} . In many typical cases, determining this adjacency is easy.

As an example, one can apply this technique to the coordinated motion planning of k disks moving in a planar polygonal environment bounded by n edges, to get a solution with $O(n^k)$ running time [SS91]. Since this problem has $2k$ degrees of freedom, this is a significant improvement over the bound $O(n^{2k} \log n)$ yielded by Canny's general algorithm.

See [ABS⁺99] for another treatment of coordinated motion planning, for two or three general independently moving robots, where algorithms that are also faster than Canny's general technique are developed.

A ROD IN A 3D POLYHEDRAL ENVIRONMENT

This problem has five degrees of freedom (three of translation and two of rotation). Recent work shows that the complexity of \mathcal{F} is only $O(n^3\lambda_4(n))$ [Kol].

TABLE 47.2.1 Summary of motion planning algorithms.

SYSTEM	MOTION	ENVIRONMENT	df	RUNNING TIME
Convex k -gon	translation	planar polygonal	2	$O(N \log m)$
Arbitrary k -gon	translation	planar polygonal	2	$O(kn \log^2 n)$
General			2	$O(\lambda_{s+2}(n) \log^2 n)$
Line segment	trans & rot	planar polygonal	3	$O(n^2 \log n)$
Convex k -gon	trans & rot	planar polygonal	3	$O(k^4 n \lambda_3(n) \log n)$ $O(kn \lambda_6(kn) \log n)$
Arbitrary k -gon	trans & rot	planar polygonal	3	$O((kn)^{2+\epsilon})$
Convex polytope	translation	3D polyhedral	3	$O(kmn \log^2 m)$
Arbitrary polytope	translation	3D polyhedral	3	$O((kn)^{2+\epsilon})$
Ball		3D polyhedral	3	$O(n^{2+\epsilon})$
General			3	$O(N^{2+\epsilon})$

MOTION PLANNING AND ARRANGEMENTS

As can be seen from the preceding subsections, motion planning is closely related to the study of arrangements of surfaces in higher dimensions. Motion planning has motivated many problems in arrangements, such as the problem of bounding the complexity of, and designing efficient algorithms for, computing a single cell in an arrangement of n low-degree algebraic surface patches in d dimensions, the problem of computing the union of geometric objects (the expanded obstacles), and the problem of decomposing higher-dimensional arrangements into subcells of constant description complexity. These problems are only partially solved and present major challenges in the study of arrangements. See [Chapter 24](#) and [SA95] for further details.

SUMMARY

Some of the above results are summarized in Table 47.2.1. For each specific system, only one or two algorithms are listed.

47.3 VARIANTS OF THE MOTION PLANNING PROBLEM

We now briefly review several variants of the basic motion planning problem, in which additional constraints are imposed on the problem. Further material on many of these problems can be found in [Chapter 48](#).

OPTIMAL MOTION PLANNING

The preceding section described techniques for determining the existence of a collision-free motion between two given placements of some moving system. It paid no attention to the optimality of the motion, which is an important consideration in practice. There are several problems involved in optimal motion planning. First, optimality is a notion that can be defined in many ways, each of which leads to different algorithmic considerations. Second, optimal motion planning is usually much harder than motion planning per se.

SHORTEST PATHS

The simplest case is when the moving system B is a single point. In this case the cost of the motion is simply the length of the path traversed by the point (normally, we use the Euclidean distance, but other metrics have been considered as well). We thus face the problem of computing **shortest paths** amidst obstacles in a 2D or 3D environment.

The planar case. Let V be a closed planar polygonal environment bounded by n edges, and let s (the “source”) be a point in V . For any other point $t \in V$, let $\pi(s, t)$ denote the (Euclidean) shortest path from s to t within V . Finding $\pi(s, t)$ for any t is facilitated by construction of the *shortest path map* $SPM(s, V)$ from s in V , a decomposition of V into regions detailed in [Chapter 27](#). A recent result [HS99] computes $SPM(s, V)$ in optimal $O(n \log n)$ time.

The same problem may be considered in other metrics. For example, it is easier to give an $O(n \log n)$ algorithm for the shortest path problem under the L_1 or L_∞ metric. See [Section 27.3](#).

The three-dimensional case. Let V be a closed polyhedral environment bounded by a total of n faces, edges, and vertices. Again, given two points $s, t \in V$, we wish to compute the shortest path $\pi(s, t)$ within V from s to t . Here $\pi(s, t)$ is a polygonal path, bending at *edges* (sometimes also at vertices) of V . To compute $\pi(s, t)$, we need to solve two subproblems: to find the sequence of edges (and vertices) of V visited by $\pi(s, t)$ (the *shortest-path sequence* from s to t), and to compute the actual points of contact of $\pi(s, t)$ with these edges. These points obey the rule that the incoming angle of $\pi(s, t)$ with an edge is equal to the outgoing angle. Hence, given the shortest-path sequence of length m , we need to solve a system of m quartic equations in m variables in order to find the contact points. This can be solved either approximately, using an iterative scheme, or exactly, using techniques of computational real algebraic geometry; the latter method requires exponential time. Even the first, more “combinatorial,” problem of computing the shortest-path sequence is NP-hard [CR87], so the general shortest-path problem is certainly much harder in three dimensions.

Many special cases of this problem, with more efficient solutions, have been studied, of which we mention the problem of computing shortest paths on a convex polytope (see [MMP87] for an exact $O(n^2 \log n)$ algorithm, which has been subsequently improved to $O(n^2)$ [CH96], and [AHSV97] for an approximate linear-time solution), and on a polyhedral terrain [MMP87, VA01, LMS97]. See also [Section 27.5](#).

VARIOUS OPTIMAL MOTION PLANNING PROBLEMS

Suppose next that the moving system B is a rigid body free only to translate in two or three dimensions. Then the notion of optimality is still well defined—it is the total distance traversed by (any reference point attached to) B . One can then apply the same techniques as above, after replacing the obstacles by their expanded versions. For example, if B is a convex polygon in the plane, and the obstacles are m pairwise openly-disjoint convex polygons A_1, \dots, A_m , then we form the Minkowski sums $K_i = A_i \oplus (-B)$, for $i = 1, \dots, m$, and compute a shortest path in the complement of their union. Since the K_i 's may overlap, we first need to compute their union, as above. A similar approach can be used in planning shortest motion of a polyhedron translating amidst polyhedra in 3-space, etc.

If B admits more complex motions, then the notion of optimality begins to be fuzzy. For example, consider the case of a line segment (“rod”) translating and rotating in a planar polygonal environment. One could measure the cost of a motion by the total distance traveled by a designated endpoint (or the centerpoint) of B , or by a weighted average between such a distance and the total turning angle of B , etc. A version of this problem has recently been shown to be NP-hard [AKY96]. See [Section 27.3](#).

The notion of optimality gets even more complicated when one introduces kinematic constraints on the motion of B . It is then often challenging even without obstacles; see [Section 48.5.4](#).

PRACTICAL APPROACHES TO MOTION PLANNING

When the number of degrees of freedom is even moderately large, exact and complete solutions of the motion planning problem are very inefficient in practice, so one seeks heuristic or other incomplete but practical solutions. Several such techniques have been developed.

Potential field. The first heuristic regards the robot as moving in a potential field induced by the obstacles and by the target placement, where the obstacles act as repulsive barriers, and the target as a strongly attracting source. By letting the robot follow the gradient of such a potential field, we obtain a motion that avoids the obstacles and that can be expected to reach the goal. An attractive feature of this technique is that planning and executing the desired motion are done in a single stage. Another important feature is the generality of the approach; it can easily be applied to systems with many degrees of freedom.

This technique, however, may lead to a motion where the robot gets stuck at a local minimum of the potential field, leaving no guarantee that the goal will be reached. To overcome this problem, several solutions have been proposed. One is to try to escape from such a “potential well” by making a few small random moves, in the hope that one of them will put the robot in a position from which the field leads it away from this well. Another approach is to use the potential field only for subproblems where the initial and final placements are close to each other, so the chance to get stuck at a local minimum is small.

Probabilistic roadmaps. In the past decade, this method has picked up momentum, and has become the method of choice in many practical motion planning systems [BKL⁺97, KSLO96]. The general approach is to generate many random

free placements throughout the workspace, and to apply any “local” simple-minded planner to plan a motion between pairs of these placements; one may use for this purpose the potential field approach, or simply attempt to connect the two placements by a straight segment in configuration space. If the configuration space is sufficiently densely sampled, enough local free paths will be generated, and they will form a roadmap, in the sense of Section 47.1.1, which can then be used to perform motion planning between any pair of input placements. This technique has been applied to the difficult problem of protein folding with some success [SA01].

A significant problem that arises is how to sample well the free configuration space; informally, the goal is to detect all “tight” passages within \mathcal{F} , which will be missed unless some placements are generated near them. See [ABD⁺98, BKL⁺97, HLM99, KSLO96, KL01] and Section 48.4 for more details concerning this technique, its extensions and variants.

Fat obstacles. Another technique exploits the fact that, in typical layouts, the obstacles can be expected to be “fat” (this has several definitions; intuitively, they do not have long and skinny parts). Also, the obstacles tend not to be too clustered, in the sense that each placement of the robot can interact with only a constant number of obstacles. These facts tend to make the problem easier to solve in such so-called *realistic* input scenes. See [vdS⁺93] for the case of fat obstacles, [vdS⁺98] for the case of environments with low obstacle density, and [BKO⁺02] for two other models of realistic input scenes.

EXPLORATORY MOTION PLANNING

If the environment in which the robot moves is not known to the system a priori, but the system is equipped with sensory devices, motion planning assumes a more “exploratory” character. If only tactile (or proximity) sensing is available, then a plausible strategy might be to move along a straight line (in physical or configuration space) directly to the target position, and when an obstacle is reached, to follow its boundary until the original straight line of motion is reached again. This technique has been developed and refined for arbitrary systems with two degrees of freedom (see, e.g., [LS87]). It can be shown that this strategy provably reaches the goal, if at all possible, with a reasonable bound on the length of the motion. This technique has been implemented on several real and simulated systems, and has applications to maze-searching problems.

One attempt to extend this technique to a system with three degrees of freedom is given in [CY91]. This technique computes within \mathcal{F} a certain one-dimensional skeleton (roadmap) \mathcal{R} which captures the connectivity of \mathcal{F} . The twist here is that \mathcal{F} is not known in advance, so the construction of \mathcal{R} has to be done in an incremental, exploratory manner. This exploration can be implemented in a controlled manner that does not require too many “probing” steps, and which enables the system to recognize when the construction of \mathcal{R} has been completed (if the goal has not been reached beforehand).

If vision is also available, then other possibilities need to be considered, e.g., the system can obtain partial information about its environment by viewing it from the present placement, and then “explore” it to gain progressively more information until the desired motion can be fully planned. Results that involve such *model-building* tasks can be found in [GMR97, ZF96] and Section 48.7.

TIME-VARYING ENVIRONMENTS

Interesting generalizations of the motion planning problem arise when some of the obstacles in the robot's environment are assumed to be moving along known trajectories. In this case the robot's goal will be to "dodge" the moving obstacles while moving to its target placement. In this "dynamic" motion planning problem, it is reasonable to assume some limit on the robot's velocity and/or acceleration. Two studies of this problem are [SM88, RS94]. They show that the problem of avoiding moving obstacles is substantially harder than the corresponding static problem. By using time-related configuration changes to encode Turing machine states, they show that the problem is PSPACE-hard even for systems with a small and fixed number of degrees of freedom. However, polynomial-time algorithms are available in a few particularly simple special cases. Another variant of this problem involves movable obstacles, which the robot B can, say, push aside to clear its passage. Again, it can be shown that the general problem of this kind is PSPACE-hard, some special instances are NP-hard, and polynomial-time algorithms are available in certain other special cases [Wil91, DZ99].

COMPLIANT MOTION PLANNING

In realistic situations, the moving system has only approximate knowledge of the geometry of the obstacles and/or of its current position and velocity, and it has an inherent amount of error in controlling its motion. The objective is to devise a strategy that will guarantee that the system reaches its goal, where such a strategy usually proceeds through a sequence of free motions (until an obstacle is hit) intermixed with *compliant motions* (sliding along surfaces of contacted obstacles) until it can be ascertained that the goal has been reached.

A standard approach to this problem is through the construction of pre-images (or back projections) [LPMT84]. Specific algorithms that solve various special cases of the problem can be found in [Bri89, Don90, FHS96]. See [Section 48.5.3](#).

NONHOLONOMIC MOTION PLANNING

Another realistic constraint on the possible motions of a given system is kinematic (or *kinodynamic*). For example, the moving object B might be constrained not to exceed certain velocity or acceleration thresholds, or has only limited steering capability. Even without any obstacles, such problems are usually quite hard, and the presence of (stationary or moving) obstacles makes them extremely complicated to solve. These so-called *nonholonomic motion planning* problems are usually handled using tools from control theory. A relatively simple special case is that of a car-like robot in a planar workspace, with a bound on the radius of curvature of its motion. Issues like reachability between two given placements (even in the absence of obstacles) raise interesting geometric considerations, where one of the goals is to identify canonical motions that always suffice to get to any reachable placement. See [Lat91, LC92, Lau98] for several books that cover this topic, and [Section 48.5.2](#). Kinodynamic motion planning is treated in [CDRX88, CRR91], and bounded-curvature motion planning is treated in [AW01, RW98].

GENERAL TASK AND ASSEMBLY PLANNING

In task planning problems, the system is given a complex task to perform, such as assembling a part from several components or restructuring its workcell into a new layout, but the precise sequence of substeps needed to attain the final goal is not specified and must be inferred by the system.

Suppose we want to manufacture a product consisting of several parts. Let S be the set of parts in their final assembled form. The first question is whether the product can be disassembled by translating in some fixed direction one part after the other, so that no collision occurs. An order of the parts that satisfies this property is called a *depth order*. It need not always exist, but when it does, the product can be assembled by translating the constituent parts one after another, in the reverse of the depth order, to their target positions. Products that can be assembled in this manner are called *stack products* [WL94]. The simplicity of the assembly process makes stack products attractive to manufacture. Computing a depth order in a given direction (or deciding that no such order exists) can be done in $O(m^{4/3+\epsilon})$ time, for any $\epsilon > 0$, for a set of polygons in 3-space with m vertices in total [dBOS94]. Faster algorithms are known for the special cases of axis-parallel polygons, c -oriented polygons, and “fat” objects.

Many products, however, are not stack products, that is, a single direction in which the parts must be moved is not sufficient to assemble the product. One solution is to search for an assembly sequence that allows a subcollection of parts to be moved as a rigid body in *some* direction. This can be accomplished in polynomial time, though the running time is rather high in the worst case: it may require $\Omega(m^4)$ time for a collection of m tetrahedra in 3-space [WL94]. A more modest, but considerably more efficient, solution allows each disassembly step to proceed in one of a few given directions [ABHS96]. It has running time $O(m^{4/3+\epsilon})$, for any $\epsilon > 0$.

A general approach to assembly planning, based on the concept of a *nondirectional blocking graph* [WL94], is proposed in [HLW00]. It is called the *motion space approach*, where the motion space plays a role parallel to configuration space in motion planning. Every point in the motion space represents a possible (dis)assembly sequence motion, all having the same number of degrees of freedom. The motion space is decomposed into an arrangement of cells where in each cell the blocking relations among the parts are invariant, namely, for a every pair of parts P, Q , P will either hit Q for all the possible motions of a cell, or avoid it. It thus suffices to check one specific motion sequence from each cell, leading to a finite complete solution. specific motion

See [Section 48.3](#) and [dML91] for further details on assembly sequencing, and [Chapter 55](#) for related problems.

ON-LINE MOTION PLANNING

Consider the problem of a point robot moving through a planar environment filled with polygonal obstacles, where the robot has no a priori information about the obstacles that lie ahead. One models this situation by assuming that the robot knows the location of the target position and of its own absolute position, but that it only acquires knowledge about the obstacles as it contacts them. The goal is to

minimize the distance that the robot travels. See also the discussion on exploratory motion planning above.

Because the robot must make decisions without knowing what lies ahead, it is natural to use the *competitive ratio* to evaluate the performance of a strategy. In particular, one would like to minimize the ratio between the distance traveled by the robot and the length of the shortest start-to-target path in that scene. The competitive ratio is the worst-case ratio achieved over all scenes having a given source-target distance. A special case of interest is when all obstacles are axis-parallel rectangles of width at least 1 located in the infinite Euclidean plane. Natural greedy strategies yield a competitive ratio of $\Theta(n)$, where n is the Euclidean source-target distance. More sophisticated algorithms obtain competitive ratios of $\Theta(\sqrt{n})$ [BRS97]. Randomized algorithms can do much better [BBF⁺96]. Through the use of randomization, one can translate the case of arbitrary convex obstacles [BRS97] to rectilinearly-aligned rectangles, at the cost of some increase in the competitive ratio. If the scene is not on an infinite plane but rather within some finite rectangular “warehouse,” and the start location is one of the warehouse corners, then the competitive ratio drops to $\log n$ [BBFY94].

COLLISION DETECTION

Although not a motion planning problem per se, collision detection is a closely related problem in robotics [LG98]. It arises, for example, when one tries to use some heuristic approach to motion planning, where the planned path is not guaranteed apriori to be collision-free. In such cases, one wishes to test whether collisions occur during the proposed motion. Several methods have been developed, including: (a) Keeping track of the closest pair of features between two objects, at least one of which is moving, and updating the closest pair, either at discrete time steps, or using *kinetic data structures* ([Chapter 50](#)). (b) Using a hierarchical representation of more complex moving systems, by means of bounding boxes or spheres, and testing for collision recursively through the hierarchical representation (see, e.g., [LGLM00] and references therein). See [Chapter 35](#) for more details.

IMPLEMENTATION OF COMPLETE SOLUTIONS

Previously, complete solutions have barely been implemented, mainly due to lack of the nontrivial infrastructure that is needed for such tasks. With the recent advancement in the laying out of such infrastructure, and in particular with tools now available in the software libraries LEDA [MN99] and CGAL [CGAL] (cf. Chapter 65), implementing complete solutions to motion planning has become feasible. A summary of progress and prospects in this domain can be found in [Hal02].

47.4 DAVENPORT-SCHINZEL SEQUENCES

Davenport-Schinzel sequences are interesting and powerful combinatorial structures that arise in the analysis and calculation of the lower or upper envelope of collections of functions, and therefore have applications in many geometric problems, including numerous motion planning problems, which can be reduced to the calculation of such an envelope. A recent comprehensive survey of Davenport-Schinzel sequences and their geometric applications can be found in [SA95].

An (n, s) **Davenport-Schinzel sequence**, where n and s are positive integers, is a sequence $U = (u_1, \dots, u_m)$ composed of n symbols with the properties:

- (i) No two adjacent elements of U are equal: $u_i \neq u_{i+1}$ for $i = 1, \dots, m - 1$.
- (ii) U does not contain as a subsequence any alternation of length $s + 2$ between two distinct symbols: there do not exist $s + 2$ indices $i_1 < i_2 < \dots < i_{s+2}$ so that $u_{i_1} = u_{i_3} = u_{i_5} = \dots = a$ and $u_{i_2} = u_{i_4} = u_{i_6} = \dots = b$, for two distinct symbols a and b .

Thus, for example, an $(n, 3)$ sequence is not allowed to contain any subsequence of the form $(a \cdots b \cdots a \cdots b \cdots a)$. Let $\lambda_s(n)$ denote the maximum possible length of an (n, s) Davenport-Schinzel sequence.

The importance of Davenport-Schinzel sequences lies in their relationship to the combinatorial structure of the lower (or upper) envelope of a collection of functions (Section 24.2). Specifically, for any collection of n real-valued continuous functions f_1, \dots, f_n defined on the real line, having the property that each pair of them intersect in at most s points, one can show that the sequence of function indices i in the order in which these functions attain their lower envelope (i.e., their pointwise minimum $f = \min_i f_i$) from left to right is an (n, s) Davenport-Schinzel sequence. Conversely, any (n, s) Davenport-Schinzel sequence can be realized in this way for an appropriate collection of n continuous univariate functions, each pair of which intersect in at most s points.

The crucial and surprising property of Davenport-Schinzel sequences is that, for a fixed s , the maximal length $\lambda_s(n)$ is nearly linear in n , although for $s \geq 3$ it is slightly super-linear. Specifically, one has

$$\begin{aligned}\lambda_1(n) &= n \\ \lambda_2(n) &= 2n - 1 \\ \lambda_3(n) &= \Theta(n\alpha(n)) \\ \lambda_4(n) &= \Theta(n \cdot 2^{\alpha(n)}) \\ \lambda_{2s}(n) &\leq n \cdot 2^{\alpha(n)^{s-1} + C_{2s}(n)} \\ \lambda_{2s+1}(n) &\leq n \cdot 2^{\alpha(n)^{s-1} \log \alpha(n) + C_{2s+1}(n)} \\ \lambda_{2s}(n) &= \Omega(n \cdot 2^{\frac{1}{(s-1)!} \alpha(n)^{s-1} + C'_{2s}(n)}),\end{aligned}$$

where $\alpha(n)$ is the inverse of Ackermann's function, and where $C_r(n)$, $C'_r(n)$ are asymptotically smaller than the leading terms in the respective exponents. Ackermann's function $A(n)$ grows extremely quickly, with $A(4)$ an exponential “tower” of 65636 2's. Thus $\alpha(n) \leq 4$ for all practical values of n . See [SA95].

If one considers the lower envelope of n continuous, but only partially defined, functions, then the complexity of the envelope is at most $\lambda_{s+2}(n)$, where s is the maximum number of intersections between any pair of functions [SA95]. Thus for a collection of n line segments (for which $s = 1$), the lower envelope consists of at most $O(n\alpha(n))$ subsegments. A surprising result is that this bound is tight in the worst case: there are collections of n segments, for arbitrarily large n , whose lower envelope does consist of $\Omega(n\alpha(n))$ subsegments. This is perhaps the most natural example of a combinatorial structure defined in terms of n simple objects, whose complexity involves the inverse Ackermann's function; see [SA95, WS88].

Algorithms. The lower envelope of n given total or partial continuous functions, each pair of which intersect in at most s points, can be computed by a simple divide-and-conquer technique that runs (in an appropriate model of computation) in time $O(\lambda_s(n) \log n)$ or $O(\lambda_{s+2}(n) \log n)$ (depending on whether the functions are totally or partially defined). A refined technique (see [Her89]) reduces the time for partially-defined functions to $O(\lambda_{s+1}(n) \log n)$. Thus, in the case of segments, the algorithm computes their lower envelope in optimal $O(n \log n)$ time. More complex combinatorial and algorithmic applications of Davenport-Schinzel sequences (such as the complexity and construction of a single face in a planar arrangement) are mentioned throughout this chapter. Extensions to higher-dimensional instances, which arise naturally in many motion planning problems, are described in the book [SA95] and in the survey articles [AS00b, AS00c].

47.5 SOURCES AND RELATED MATERIAL

SURVEYS

The results not given an explicit reference above, and additional material on motion planning and related problems may be traced in these surveys:

[Lat91]: A book devoted to robot motion planning.

[HSS87]: A collection of early papers on motion planning.

[SA95]: A book on Davenport-Schinzel sequences and their geometric applications; contains a section on motion planning.

[HS95b]: A recent review on arrangements and their applications to motion planning.

[SS88, SS90, Sha89, Sha95, AY90]: Several survey papers on algorithmic motion planning.

[AS00b, AS00c]: Recent surveys on Davenport-Schinzel sequences and on higher-dimensional arrangements.

RELATED CHAPTERS

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

- Chapter 24: Arrangements
 - Chapter 27: Shortest paths and networks
 - Chapter 33: Computational real algebraic geometry
 - Chapter 35: Collision detection
 - Chapter 48: Robotics
 - Chapter 50: Algorithms for tracking moving objects
-

REFERENCES

- [AAS98] P.K. Agarwal, N. Amenta, and M. Sharir. Largest placement of one convex polygon inside another. *Discrete Comput. Geom.*, 19:95–104, 1998.
- [AAS99] P.K. Agarwal, B. Aronov, and M. Sharir. Motion planning for a convex polygon in a polygonal environment. *Discrete Comput. Geom.*, 22:201–221, 1999,
- [ABHS96] P.K. Agarwal, M. de Berg, D. Halperin, and M. Sharir. Efficient generation of k -directional assembly sequences. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 122–131, 1996.
- [AFH02] P.K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of Minkowski sums. *Comput. Geom. Theory Appl.*, 21:39–61, 2002.
- [AHSV97] P.K. Agarwal, S. Har-Peled, M. Sharir, and K.R. Varadarajan. Approximate shortest paths on a convex polytope in three dimensions. *J. Assoc. Comput. Mach.*, 44:567–584, 1997.
- [AS00a] P.K. Agarwal and M. Sharir. Pipes, cigars, and kreplach: The union of Minkowski sums in three dimensions. *Discrete Comput. Geom.*, 24:645–685, 2000.
- [AS00b] P.K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In *Handbook of Computational Geometry*, J.R. Sack and J. Urrutia, editors, pages 1–47, Elsevier North-Holland, Amsterdam, 2000.
- [AS00c] P.K. Agarwal and M. Sharir. Arrangements of surfaces in higher dimensions. in *Handbook of Computational Geometry*, J.R. Sack and J. Urrutia, editors, pages 49–119, Elsevier North-Holland, Amsterdam, 2000.
- [AW01] P.K. Agarwal and H. Wang. Approximation algorithms for shortest paths with bounded curvature. *SIAM J. Comput.*, 30:1739–1772, 2001.
- [AFK⁺92] H. Alt, R. Fleischer, M. Kaufmann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Approximate motion planning and the complexity of the boundary of the union of simple geometric figures. *Algorithmica*, 8:391–406, 1992.
- [AY90] H. Alt and C.K. Yap. Algorithmic aspects of motion planning: A tutorial, Parts 1 and 2. *Algorithms Rev.*, 1:43–60, 61–77, 1990.
- [ABD⁺98] N.M. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective (WAFR '98)*, P.K. Agarwal, L.E. Kavraki, and M. Mason, editors, pages 155–168, A.K. Peters, Wellesley, 1998.
- [ABS⁺99] B. Aronov, M. de Berg, A.F. van der Stappen, P. Švestka, and J. Vleugels. Motion planning for multiple robots. *Discrete Comput. Geom.*, 22:505–525, 1999.
- [AS94] B. Aronov and M. Sharir. Castles in the air revisited. *Discrete Comput. Geom.*, 12:119–150, 1994.
- [AS97] B. Aronov and M. Sharir. On translational motion planning of a convex polyhedron in 3-space. *SIAM J. Comput.*, 26:1785–1803, 1997.

- [AST97] B. Aronov, M. Sharir, and B. Tagansky. The union of convex polyhedra in three dimensions. *SIAM J. Comput.*, 26:1670–1688, 1997.
- [AKY96] Te. Asano, D.G. Kirkpatrick, and C.K. Yap. d_1 -optimal motion for a rod. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 252–263, 1996.
- [ABF89] F. Avnaim, J.-D. Boissonnat, and B. Faverjon. A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. *Lecture Notes Comput. Sci.*, volume 391:67–86, Springer-Verlag, Berlin, 1989.
- [BBFY94] E. Bar-Eli, P. Berman, A. Fiat, and P. Yan. On-line navigation in a room. *J. Algorithms*, 17:319–341, 1994.
- [BKL⁺97] J. Barraquand, L.E. Kavraki, J.-C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling framework for path planning in large-dimensional configuration spaces. *Internat. J. Robot. Res.*, 16:759–774, 1997.
- [Bas03] S. Basu. On the combinatorial and topological complexity of a single cell. *Discrete Comput. Geom.*, 29:41–59, 2003.
- [BPR00] S. Basu, R. Pollack, and M.-F. Roy. Computing roadmaps of semi-algebraic sets on a variety. *J. Amer. Math. Soc.*, 13:55–82, 2000.
- [BBF⁺96] P. Berman, A. Blum, A. Fiat, H. Karloff, A. Rosen, and M. Saks. Randomized robot navigation algorithms. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 75–84, 1996.
- [Ber00] R.-P. Berretty. *Geometric Design of Part Feeders*. Ph.D. thesis, Utrecht Univ., Utrecht, The Netherlands, 2000.
- [BRS97] A. Blum, P. Raghavan, and B. Schieber. Navigating in unfamiliar geometric terrain. *SIAM J. Comput.*, 26:110–137, 1997.
- [Bri89] A.J. Briggs. An efficient algorithm for one-step planar compliant motion planning with uncertainty. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 187–196, 1989.
- [CGAL] CGAL, The Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [Can87] J.F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, 1987. See also: Computing roadmaps in general semi-algebraic sets. *Comput. J.*, 36:504–514, 1993.
- [CDRX88] J.F. Canny, B.R. Donald, J.H. Reif, and P. Xavier. On the complexity of kinodynamic planning. In *Proc. 29th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 306–316, 1988.
- [CRR91] J.F. Canny, A. Rege, and J.H. Reif. An exact algorithm for kinodynamic planning in the plane. *Discrete Comput. Geom.*, 6:461–484, 1991.
- [CR87] J.F. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.
- [CEG⁺93] B. Chazelle, H. Edelsbrunner, L.J. Guibas, M. Sharir, and J. Snoeyink. Computing a face in an arrangement of line segments and related problems. *SIAM J. Comput.*, 22:1286–1302, 1993.
- [CH96] J. Chen and Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.
- [CK93] L.P. Chew and K. Kedem. A convex polygon among polygonal obstacles: placement and high-clearance motion. *Comput. Geom. Theory Appl.*, 3:59–89, 1993.
- [CKS⁺98] L.P. Chew, K. Kedem, M. Sharir, B. Tagansky, and E. Welzl. Voronoi diagrams of lines in three dimensions under polyhedral convex distance functions. *J. Algorithms*, 29:238–255, 1998.

- [Col75] G.E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conf. Automata Theory Formal Languages, Lecture Notes Comput. Sci.*, volume 33, pages 134–183, Springer-Verlag, Berlin, 1975.
- [CY91] J. Cox and C.K. Yap. On-line motion planning: Case of a planar rod. *Ann. Math. Artif. Intell.*, 3:1–20, 1991.
- [dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.
- [BKO⁺02] M. de Berg, M.J. Katz, M.H. Overmars, A.F. van der Stappen, and J. Vleugels. Models and motion planning. *Comput. Geom. Theory Appl.*, 23:53–68, 2002.
- [dBOS94] M. de Berg, M.H. Overmars, and O. Schwarzkopf. Computing and verifying depth orders. *SIAM J. Comput.*, 23:437–446, 1994.
- [dML91] L.S. Homem de Mello and S. Lee, editors. *Computer-Aided Mechanical Assembly Planning*. Kluwer Academic Publishers, Boston, 1991.
- [Don90] B.R. Donald. The complexity of planar compliant motion planning under uncertainty. *Algorithmica*, 5:353–382, 1990.
- [DZ99] D. Dor and U. Zwick. SOKOBAN and other motion planning problems. *Comput. Geom. Theory Appl.*, 13:215–228, 1999.
- [FHS96] J. Friedman, J. Hershberger, and J. Snoeyink. Efficiently planning compliant motion in the plane. *SIAM J. Comput.*, 25:562–599, 1996.
- [GMR97] L.J. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. *SIAM J. Comput.*, 26:1120–1138, 1997.
- [GSS89] L.J. Guibas, M. Sharir, and S. Sifrony. On the general motion planning problem with two degrees of freedom. *Discrete Comput. Geom.*, 4:491–521, 1989.
- [Hal02] D. Halperin. Robust geometric computing in motion. *Internat. J. Robot. Res.*, 21:219–232, 2002.
- [HLW00] D. Halperin, J.-C. Latombe, and R.H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26:577–601, 2000.
- [HOS92] D. Halperin, M.H. Overmars, and M. Sharir. Efficient motion planning for an L-shaped object in the plane. *SIAM J. Comput.*, 21:1–23, 1992.
- [HS95a] D. Halperin and M. Sharir. Almost tight upper bounds for the single cell and zone problems in three dimensions. *Discrete Comput. Geom.*, 14:385–410, 1995.
- [HS95b] D. Halperin and M. Sharir. Arrangements and their applications in robotics: Recent developments. In *The Algorithmic Foundations of Robotics*, K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, pages 495–511. A.K. Peters, Boston, 1995.
- [HS96] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.
- [HY98] D. Halperin and C.K. Yap. Combinatorial complexity of translating a box in polyhedral 3-space. *Comput. Geom. Theory Appl.*, 9:181–196, 1998.
- [HCA⁺95] S. Har-Peled, T.M. Chan, B. Aronov, D. Halperin, and J. Snoeyink. The complexity of a single face of a Minkowski sum. In *Proc. 7th Canad. Conf. Comput. Geom.*, Québec City, pages 91–96, 1995.
- [Her89] J. Hershberger. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inform. Process. Lett.*, 33:169–174, 1989.
- [HS99] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28:2215–2256, 1999.

- [HJW84] J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides. Movement problems for 2-dimensional linkages. *SIAM J. Comput.*, 13:610–629, 1984.
- [HJW85] J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM J. Comput.* 14:315–333, 1985.
- [HSS84] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects: P-space hardness of the “Warehouseman’s Problem.” *Internat. J. Robot. Res.*, 3:76–88, 1984.
- [HSS87] J.E. Hopcroft, J.T. Schwartz, and M. Sharir, editors. *Planning, Geometry, and Complexity of Robot Motion*. Ablex, Norwood, 1987.
- [HLM99] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Internat. J. Comput. Geom. Appl.*, 9:495–512, 1999.
- [KSLO96] L.E. Kavraki, P. Švestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580, 1996.
- [KO88] Y. Ke and J. O’Rourke. Lower bounds on moving a ladder in two and three dimensions. *Discrete Comput. Geom.*, 3:197–217, 1988.
- [KLPS86] K. Kedem, R. Livne, J. Pach, and M. Sharir. On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. *Discrete Comput. Geom.*, 1:59–71, 1986.
- [KS90] K. Kedem and M. Sharir. An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space. *Discrete Comput. Geom.*, 5:43–75, 1990.
- [KST97] K. Kedem, M. Sharir, and S. Toledo. On critical orientations in the Kedem-Sharir motion planning algorithm. *Discrete Comput. Geom.*, 17:227–239, 1997.
- [Kol] V. Koltun. Personal communication.
- [KS02a] V. Koltun and M. Sharir. Three-dimensional Euclidean Voronoi diagrams of lines with a fixed number of orientations. In *Proc. 18th ACM Annu. Sympos. Comput. Geom.*, pages 217–226, 2002.
- [KS02b] V. Koltun and M. Sharir. Polyhedral Voronoi diagrams of polyhedral sites in three dimensions. In *Proc. 18th ACM Annu. Sympos. Comput. Geom.*, pages 227–236, 2002.
- [KL01] J.J. Kuffner and S.M. LaValle. Rapidly exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Dimensions (WAFR ’00)*, B.R. Donald, K.M. Lynch, and D. Rus, editors, pages 293–308, A.K. Peters, Wellesley, 2001.
- [LMS97] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 274–283, 1997.
- [LGGM00] E. Larsen, S. Gottschalk, M.C. Lin, and D. Manocha. Fast distance queries using rectangular swept sphere volume. In *Proc. IEEE Internat. Conf. Robotics Autom.*, 2000.
- [Lat91] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic, Boston, 1991.
- [Lau98] J.-P. Laumond, editor. *Robot Motion Planning and Control. Lectures Notes Control Inform. Sci.*, volume 229, Springer-Verlag, Berlin, 1998.
- [LS87a] D. Leven and M. Sharir. Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. *Discrete Comput. Geom.*, 2:9–31, 1987.

- [LS87b] D. Leven and M. Sharir. An efficient and simple motion planning algorithm for a ladder moving in 2-dimensional space amidst polygonal barriers. *J. Algorithms*, 8:192–215, 1987.
- [LC92] Z. Li and J.F. Canny, editors. *Nonholonomic Motion Planning*. Kluwer Academic, Norwell, 1992.
- [LG98] M.C. Lin and S. Gottschalk. Collision detection between geometric models: A survey. In *Proc. IMA Conf. Math. Surfaces*, 1998.
- [LPMT84] T. Lozano-Pérez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *Internat. J. Robot. Res.*, 3:3–24, 1984.
- [LS87] V.J. Lumelsky and A.A. Stepanov. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.
- [MMP⁺91] J. Matoušek, N. Miller, J. Pach, M. Sharir, S. Sifrony, and E. Welzl. Fat triangles determine linearly many holes. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–58, 1991.
- [MN99] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*, Cambridge University Press, 1999.
- [MMP87] J.S.B. Mitchell, D.M. Mount, and C.H. Padadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.
- [OSY87] C. Ó'Dúnlaing, M. Sharir, and C.K. Yap. Generalized Voronoi diagrams for a ladder: II. Efficient construction of the diagram. *Algorithmica*, 2:27–59, 1987.
- [Rei87] J.H. Reif. Complexity of the generalized mover's problem. In *Planning, Geometry, and Complexity of Robot Motion*, J.E. Hopcroft, J.T. Schwartz, and M. Sharir, editors, pages 267–281, Ablex, Norwood, 1987.
- [RS94] J.H. Reif and M. Sharir. Motion planning in the presence of moving obstacles. *J. Assoc. Comput. Mach.*, 41:764–790, 1994.
- [RW98] J.H. Reif and H. Wang. The complexity of the two-dimensional curvature-constrained shortest-path problem. In *Proc. 3rd Workshop the Algo. Found. Robotics*, P.K. Agarwal, L.E. Kavraki, and M. Mason, editors, pages 49–58, A.K. Peters, Natick, 1998.
- [SS83] J.T. Schwartz and M. Sharir. On the piano movers' problem: II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, 4:298–351, 1983.
- [SS88] J.T. Schwartz and M. Sharir. A survey of motion planning and related geometric algorithms. *Artif. Intell.*, 37:157–169, 1988. Also in D. Kapur and J. Mundy, editors, *Geometric Reasoning*, pages 157–169. MIT Press, Cambridge, 1989. And in S.S. Iyengar and A. Elfes, editors, *Autonomous Mobile Robots*, volume I, pages 365–374. IEEE Computer Society Press, Los Alamitos, 1991.
- [SS90] J.T. Schwartz and M. Sharir. Algorithmic motion planning in robotics. In J. van Leeuwen, editor, *Handbook of Theoret. Comput. Sci., Volume A: Algorithms and Complexity*, pages 391–430. Elsevier, Amsterdam, 1990.
- [SS97] O. Schwarzkopf and M. Sharir. Vertical decomposition of a single cell in a 3-dimensional arrangement of surfaces. *Discrete Comput. Geom.*, 18:269–288, 1997.
- [Sha89] M. Sharir. Algorithmic motion planning in robotics. *Computer*, 22:9–20, 1989.
- [Sha95] M. Sharir. Robot motion planning. *Comm. Pure Appl. Math.*, 48:1173–1186, 1995. Also in E. Schonberg, editor, *The Houses That Jack Built*. Courant Institute, New York, 1995, 287–300.

- [SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [SS91] M. Sharir and S. Sifrony. Coordinated motion planning for two independent robots. *Ann. Math. Artif. Intell.*, 3:107–130, 1991.
- [ST94] M. Sharir and S. Toledo. Extremal polygon containment problems. *Comput. Geom. Theory Appl.*, 4:99–118, 1994.
- [SS87] S. Sifrony and M. Sharir. A new efficient motion planning algorithm for a rod in two-dimensional polygonal space. *Algorithmica*, 2:367–402, 1987.
- [SA01] G. Song and N.M. Amato. Using motion planning to study protein folding pathways. In *Internat. Conf. Research Comput. Molecular Biology*, pages 287–296, 2001.
- [SM88] K. Sutner and W. Maass. Motion planning among time-dependent obstacles. *Acta Inform.*, 26:93–122, 1988.
- [vdS⁺93] A.F. van der Stappen, D. Halperin, and M.H. Overmars. The complexity of the free space for a robot moving amidst fat obstacles. *Comput. Geom. Theory Appl.*, 3:353–373, 1993.
- [vdS⁺98] A.F. van der Stappen, M.H. Overmars, M. de Berg, and J. Vleugels. Motion planning in environments with low obstacle density. *Discrete Comput. Geom.*, 20:561–587, 1998.
- [VA01] K.R. Varadarajan and P.K. Agarwal. Approximate shortest paths on a nonconvex polyhedron. *SIAM J. Comput.*, 30:1321–1340, 2001.
- [Veg90] G. Vegter. The visibility diagram: A data structure for visibility problems and motion planning. *Proc. 2nd Scand. Workshop Algorithm Theory, Lecture Notes Comput. Sci.*, volume 447, pages 97–110, Springer-Verlag, Berlin, 1990.
- [WS88] A. Wiernik and M. Sharir. Planar realization of nonlinear Davenport–Schinzel sequences by segments. *Discrete Comput. Geom.*, 3:15–47, 1988.
- [Wil91] G. Wilfong. Motion planning in the presence of movable obstacles. *Ann. Math. Artif. Intell.*, 3:131–150, 1991.
- [WL94] R.H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artif. Intell.*, 71:371–396, 1994.
- [Yap87a] C.K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete Comput. Geom.*, 2:365–393, 1987.
- [Yap87b] C.K. Yap. Algorithmic motion planning. in *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics* (J.T. Schwartz and C.K. Yap, editors), Lawrence Erlbaum Associates, Hillsdale, 1987, 95–143.
- [ZF96] Z. Zhang and O. Faugeras. A 3D world model builder with a mobile robot. *Internat. J. Robot. Res.*, 11:269–285, 1996.

48 ROBOTICS

Dan Halperin, Lydia E. Kavraki, and Jean-Claude Latombe

INTRODUCTION

Robotics is concerned with the generation of computer-controlled motions of physical objects in a wide variety of settings. Because physical objects define spatial distributions in 3-space, geometric representations and computations play an important role in robotics. As a result the field is a significant source of practical problems for computational geometry. There are substantial differences, however, in the ways researchers in robotics and in computational geometry address related problems. Robotics researchers are primarily interested in developing methods that work well in practice and can be combined into integrated systems. They often pay less attention than researchers in computational geometry to the underlying combinatorial and complexity issues (the focus of [Chapter 47](#)). This difference in approach will become clear in the present chapter.

In Section 48.1 we survey basic definitions and problems in robot kinematics. Part manipulation is discussed in Section 48.2 with emphasis on part grasping, fixturing, and feeding. In Section 48.3 we present algorithms for assembly sequencing. The basic path planning problem is the topic of Section 48.4. Extensions of this problem, in particular nonholonomic motion planning, are discussed in Section 48.5. We briefly survey additional topics in two sections that follow: data structures for representing moving objects in Section 48.6, and sensing and localization in Section 48.7.

GLOSSARY

Workspace W : A subset of 2D or 3D physical space: $W \subset \mathbb{R}^k$ ($k = 2$ or 3).

Body: Rigid physical object modeled as a compact manifold with boundary $B \subset \mathbb{R}^k$ ($k = 2$ or 3). B 's boundary is assumed piecewise-smooth. We will use the terms “body,” “physical object,” and “part” interchangeably.

Robot: A collection of bodies capable of generating their own motions.

Configuration: Any mathematical specification of the position and orientation of every body composing a robot, relative to a fixed coordinate system. The configuration of a single body is also called a **placement** or a **pose**.

Configuration space \mathcal{C} : Set of all configurations of a robot. For almost any robot, this set is a smooth manifold. We will always denote the configuration space of a robot by \mathcal{C} and its dimension by m . Given a robot A , we will let $A(\mathbf{q})$ denote the subset of the workspace occupied by A at configuration \mathbf{q} .

Number of degrees of freedom: The dimension m of \mathcal{C} . In the following we will abbreviate “degree of freedom” by **dof**.

48.1 KINEMATICS

Many robots consist of multiple bodies connected by joints, which may be either actuated or passive. The spatial relations among these bodies and the space of their feasible motions is an important area of study in robotics; cf. Section 59.4.1.

GLOSSARY

Linkage: A collection of bodies, called *links*, in which some pairs of links are connected by *joints*. The graph whose nodes (resp. edges) represent links (resp. joints) is connected.

Prismatic joint: A joint between two links that allows one link to translate along a line attached to the other.

Revolute joint: A joint between two links that allows one link to rotate about a line attached to the other.

Joint parameter: A real parameter associated with a prismatic or revolute joint whose value uniquely determines the relative position or orientation of the two links connected by that joint.

Robot arm: Serial linkage such that the first link, called the *base*, is fixed in space. The last link is called the *end-effector*.

There are other types of joints besides the prismatic and revolute joints considered in this chapter. Most of them can be reduced to independent prismatic and/or revolute joints. For example, a *telescopic joint* is equivalent to collinear prismatic joints connecting links that penetrate one another. We also note that some industrial robot arms contain closed mechanical loops. For many computational purposes, however, they can be considered as serial linkages, as we assume here.

NUMBER OF DEGREES OF FREEDOM OF A LINKAGE

Let L be an arbitrary linkage with n_{link} links and n_{joint} joints, with each joint either prismatic or revolute. The number of dofs of L , denoted by n_{dof} , is the number of joints in L that can move independently with the others complying, and is given by the Grübeler formula [Rot94]:

$$n_{dof} \geq n_0(n_{link} - 1) - (n_0 - 1)n_{joint},$$

where $n_0 = 3$ if the linkage is planar, and $n_0 = 6$ if the linkage is in 3-space. In general, this formula holds with equality. The strict “greater-than” is needed only for mechanisms with special proportions or alignments.

If L is a serial linkage, we have $n_{link} = n_{joint} + 1$. So $n_{dof} = n_{joint}$. If L consists of a single closed loop, we have $n_{link} = n_{joint}$. So $n_{dof} = n_{joint} - n_0$; thus, one degree of freedom requires 4 joints in 2-space and 7 joints in 3-space. If L consists of multiple loops, the Grübeler formula yields $n_{dof} = n_{joint} - n_0\ell$, where ℓ is the number of independent loops.

FORWARD AND INVERSE KINEMATICS

The number of dofs of a robot arm is equal to its number of joints. The determination of the placement of the end-effector from the joint parameters is called the *direct kinematics problem*. In order for the last link's placement to span a 6-space, the arm must have at least 6 joints. (See [Figure 59.4.1](#).)

The determination of the values of the arm's joint parameters from the last link's placement is called the *inverse kinematics problem*. For a 6-joint arm this problem has at most 16 distinct solutions (except for some singularities). In other words, at most 16 distinct legal placements of the arm's links achieve the same specified placement of the end-effector. If the arm has two prismatic joints, then the maximum drops to 8. If it has three prismatic joints, it drops to 2. Any time three consecutive revolute joints have intersecting or parallel axes, the number is at most 8 (see [Rot94]).

OPEN PROBLEM

Given a workspace W , find the optimal design of a robot arm that can reach everywhere in W without collision. Several variants of this problem are solved in [Kol95]. However the 3D case is largely open. An extension of this problem also asks for a design of the layout of the workspace so that a certain task can be completed efficiently. (Additional reachability problems for planar robot arms and their solutions are presented in [O'R98, Section 8.6].)

48.2 PART MANIPULATION

Part manipulation is one of the most frequently performed operations in industrial robotics: parts are grasped from conveyor belts, they are oriented prior to feeding assembly workcells, and they are immobilized for machining operations.

GLOSSARY

Wrench: A pair $[\mathbf{f}, \mathbf{p} \times \mathbf{f}]$, where \mathbf{p} denotes a point in the boundary of a body B , represented by its coordinate vector in a frame attached to B , \mathbf{f} designates a force applied to B at \mathbf{p} , and \times is the vector cross-product. If \mathbf{f} is a unit vector, the wrench is said to be a *unit* wrench.

Finger: A tool that can apply a wrench.

Grasp: A set of unit wrenches $\mathbf{w}_i = [\mathbf{f}_i, \mathbf{p}_i \times \mathbf{f}_i]$, $i = 1, \dots, p$, defined on a body B , each created by a finger in contact with the boundary, ∂B , of B . For each \mathbf{w}_i , if the contact is frictionless, \mathbf{f}_i is normal to ∂B at \mathbf{p}_i ; otherwise, it can span the friction cone defined by the Coulomb law.

Force-closure grasp: A grasp $\{\mathbf{w}_i\}_{i=1,\dots,p}$ such that, for any arbitrary wrench \mathbf{w} , there exists a set of real values $\{f_1, \dots, f_p\}$ achieving $\sum_{i=1}^p f_i \mathbf{w}_i = -\mathbf{w}$. In other words, a force-closure grasp can resist any external wrenches applied to B . If contacts are nonsticky, we require that $f_i \geq 0$ for all $i = 1, \dots, p$, and the

grasp is called *positive*. In this section we only consider positive grasps.

Form-closure grasp: A positive force-closure grasp in which all finger-body contacts are frictionless.

48.2.1 GRASPING

Grasp analysis and synthesis has been an active research area over the last decade and has contributed to the development of robotic hands and grasping mechanisms.

SIZE OF A FORM/FORCE CLOSURE GRASP

The following results are shown in [MNP90, MSS87]:

- Bodies with rotational symmetry (e.g., disks in 2-space, spheres and cylinders in 3-space) admit no form-closure grasps.
- All other bodies admit a form-closure grasp with at most four fingers in 2-space and twelve fingers in 3-space.
- All polyhedral bodies have a form-closure grasp with seven fingers.
- With frictional finger-body contacts, all bodies admit a force-closure grasp that consists of three fingers in 2-space and four fingers in 3-space.

TESTING FOR FORM/FORCE CLOSURE

A necessary and sufficient condition for force closure in 2-space (resp. 3-space) is that the finger wrenches span three (resp. six) dimensions and that a strictly positive linear combination of them be zero. Said otherwise, the null wrench (the origin) should lie in the interior of the convex hull H of the finger wrenches [MSS87]. This condition provides an effective test for deciding in constant time whether a given grasp achieves force closure. A related quantitative measure of the quality of a grasp (one among several metrics proposed) is the radius of the maximum ball centered at the origin and contained in the convex hull H [KMY92].

SYNTHESIZING FORM/FORCE CLOSURE GRASPS

Most research has concentrated on computing grasps with two to four nonsticky fingers. Optimization techniques and elementary Euclidean geometry are used in [MNP90] to derive an algorithm computing a single force-closure grasp of a polygonal or polyhedral part. This algorithm is linear in the part complexity. Other linear-time techniques using results from combinatorial geometry (Steinitz's theorem) are presented in [MSS87, Mis95]. Optimal force-closure grasps are synthesized in [FC92] by maximizing the set of external wrenches that can be balanced by the contact wrenches.

Finding the maximal regions on a body where fingers can be positioned independently while achieving force closure makes it possible to accommodate errors in finger placement. Geometric algorithms for constructing such regions are proposed in [Ngu88] for grasping polygons with two fingers (with friction) and four fingers (without friction), and for grasping polyhedra with three fingers (with frictional contact capable of generating torques) and seven fingers (without friction).

Curved obstacles have also been studied [PSS⁺97]. The latter paper contains a good overview of work on the effect of curvature at contact points on grasp planning.

DEXTROUS GRASPING

Reorienting a part by moving fingers on the part’s surface is often considered to lie in the broader realm of grasping. Finger gait algorithms and nonholonomic rolling contacts (Section 48.5.2) for fingertips have been explored.

48.2.2 FIXTURING

Most manufacturing operations require fixtures to hold parts. To avoid the custom design of fixtures for each part, modular reconfigurable fixtures are often used. A typical modular fixture consists of a workholding surface, usually a plane, that has a lattice of holes where locators, clamps, and edge fixtures can be placed. Locators are simple round pins, while clamps apply pressure on the part.

Contacts between fixture elements and parts are generally assumed to be frictionless. In modular fixturing, contact locations are restricted by the lattice of holes, and form closure cannot always be achieved. In particular, when three locators and one clamp are used on a workholding plane, there exist polygons of arbitrary size for which no fixture design can be achieved [ZG95]. But if parts are restricted to be rectilinear, a fixture can always be found as long as all edges have length at least four lattice units [Mis91]. Algorithms for computing all placements of (frictionless) point fingers that put a polygonal part in form closure and all placements of point fingers that achieve “2nd-order immobility” [RB98] of a polygonal part are presented in [vSWO00].

When the fixturing kit consists of a latticed workholding plane, three locators, and one clamp, the algorithm in [BG96] finds all possible placements of a given part on the workholding surface where form closure can be achieved, along with the corresponding positions of the locators and the clamp. The algorithm in [ORSW95] computes the form-closure fixtures of input polygonal parts using a kit containing one edge fixture, one locator, and one clamp.

An algorithm for fixturing an assembly of parts that are not rigidly fastened together is proposed in [Mat95]. A large number of fixturing contacts are first scattered at random on the external boundary of the assembly. Redundant contacts are then pruned until the stability of the assembly is no longer guaranteed.

48.2.3 PART FEEDING

Part feeders account for a large fraction of the cost of a robotic assembly workcell. A typical feeder must bring parts at subsecond rates with high reliability. The problem of part-feeder design is formalized in [Nat89] in terms of a set of functions—called *transfer functions*—which map configurations to configurations. The goal is then to find a composition of these functions that maps each configuration to a unique final configuration (or a small set of final configurations). Given k transfer functions and n possible configurations, the shortest composition that will result in the smallest number of final configurations can be found in $O(kn^4)$ [Nat89]. If the transfer functions are all monotone, the complexity is reduced to $O(kn^2)$ [Epp90].

Part feeding often relies on *nonprehensile manipulation*. Nonprehensile

manipulation exploits task mechanics to achieve a goal state without grasping and frequently allows accomplishing complex feeding tasks with few dofs. It may also enable a robot to move parts that are too large or heavy to be grasped and lifted.

Pushing is one form of nonprehensile manipulation. Work on pushing originated in [Mas82] where a simple rule is established to qualitatively determine the motion of a pushed object. This rule makes use of the position of the center of friction of the object on the supporting surface. Given a part we can compute its *push* transfer function. The push function, $p_\alpha : S_1 \rightarrow S_1$, when given an orientation θ returns the orientation of the part $p_\alpha(\theta)$ after it has been pushed from direction α by a fence orthogonal to the push direction. With a sequence of different push operations it is possible to uniquely orient a part. The push function has been used in several nonprehensile manipulation algorithms:

- A planning algorithm for a robot that tilts a tray containing a planar part of known shape to orient it to a desired orientation [EM88]. This algorithm was extended to the polyhedral case in [EMV93].
- An algorithm to compute the design of a sequence of curved fences along a conveyor belt to reorient a given polygonal part [WGPB96]. See also [BGO⁺98].
- An algorithm that computes a sequence of motions of a single articulated fence on a conveyor belt that achieves a goal orientation of an object [AHLM00].

A frictionless parallel-jaw gripper was used in [Gol93] to orient polygonal parts. For any part P having an n -sided convex hull, there exists a sequence of $2n - 1$ squeezes achieving a single orientation of P (up to symmetries of the convex hull). This sequence is computed in $O(n^2)$ time [CI95]. The result has been generalized to planar parts having a piecewise algebraic convex hull [RG95]. It was shown [vSGO00] that one could design plans whose length depends on a parameter that describes the part's shape (called *geometric eccentricity* in [vSGO00]) rather than on the description of the combinatorial complexity of the part. For the parallel-jaw gripper we can define the *squeeze* transfer function. In [MGEF02] another transfer function is defined: the *roll* function. With this function a part is rolled between the jaws by making one jaw slide in the tangential direction. Using a combination of squeeze and roll primitives a polygonal part can be oriented without changing the orientation of the gripper.

Distributed manipulation systems provide another form of nonprehensile manipulation. These systems induce motions on objects through the application of many external forces. The part-orienting algorithm for the parallel-jaw gripper has been adapted for arrays of microelectromechanical actuators which—due to their tiny size—can generate almost continuous fields [BDM99]. Algorithms that position and orient parts based on identifying a finite number (depending on the number of vertices of the part) of distinct equilibrium configurations were also given in [BDM99]. Subsequent work showed that using a carefully selected actuators field, it is possible to position and orient parts in two stable equilibrium configurations [Kav97]. Finally, a long standing conjecture was proved, namely that there exists a field that can uniquely position and orient parts in a single step [BDKL00]. In fact, two different such fields were fully analyzed in [LK01b, SK01]. On the macroscopic scale it was shown that in-plane vibration can be used for closed-loop manipulation of objects using vision systems for feedback [RMC00], that arrays of controllable airjets can manipulate paper [YB00], and that foot-sized discrete actuator arrays can handle heavier objects under various manipulation strategies [LMC01].

OPEN PROBLEMS

A major open practical problem is to predict feeder throughputs to evaluate alternative feeder designs, given the geometry of the parts to be manipulated. In relation to this problem, simulation algorithms have been proposed recently to predict the pose of a part dropped on a horizontal surface [MZG⁺96], and on arbitrary surfaces [ME02b]. In distributed manipulation, an open problem is to analyze the effect of discrete arrays of actuators on the positioning and orientation of parts [LMC01, LK01b].

48.3 ASSEMBLY SEQUENCING

Most mechanical products consist of multiple parts. The goal of assembly sequencing is to compute both an order in which parts can be assembled and the corresponding required movements of the parts.

GLOSSARY

Assembly: Collection of bodies in some given relative placements.

Subassembly: Subset of the bodies composing an assembly A in their relative positions and orientations in A .

Separated subassemblies: Subassemblies that are arbitrarily far apart from one another.

Hand: A tool that can hold an arbitrary number of bodies in fixed relative placements.

Assembly operation: A motion that merges s pairwise-separated subassemblies ($s \geq 2$) into a new subassembly; each subassembly moves as a single body. No overlapping between bodies is allowed during the operation. The parameter s is called the **number of hands of the operation**. We call the reverse of an assembly operation **assembly partitioning**.

Assembly sequence: A total ordering on assembly operations that merge the separated parts composing an assembly into this assembly. The maximum, over all the operations in the sequence, of the number of hands required by an operation is called the **number of hands of the sequence**.

Monotone assembly sequence: A sequence in which no operation brings a body to an intermediate placement (relative to other bodies), before another operation transfers it to its final placement. See [Figure 48.3.1](#).

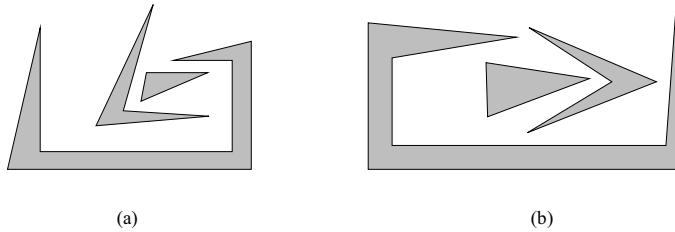
NUMBER OF HANDS IN ASSEMBLY

Every assembly of convex polygons in the plane has a two-handed assembly sequence of translations. In the worst case, s hands are necessary and sufficient for assemblies of s star-shaped polygons/polyhedra [Nat88].

There exists an assembly of six tetrahedra without a two-handed assembly sequence of translations, but with a three-handed sequence of translations. Every

FIGURE 48.3.1

Both assemblies below admit two-handed sequences with translational motions only. While (a) accepts such a monotone sequence, (b) does not. To disassemble (b) the triangle must be translated to an intermediate position [HW95]. If general motions are accepted, there exists a monotone two-handed sequence for (b). A monotone three-handed sequence with translations only is also possible.



assembly of five or fewer convex polyhedra admits a two-handed assembly sequence of translations. There exists an assembly of thirty convex polyhedra that cannot be assembled with two hands [SS94].

COMPLEXITY OF ASSEMBLY SEQUENCING

When arbitrary sequences are allowed, assembly sequencing is PSPACE-hard. The problem remains PSPACE-hard even when the bodies are polygons, each with a constant number of vertices [Nat88].

When only two-handed monotone sequences are permitted, deciding if an assembly A can be partitioned into two subassemblies S and $A \setminus S$ such that they can be separated by an arbitrary motion is NP-complete [WKL⁺95]. The problem remains NP-complete when both S and $A \setminus S$ are required to be connected and motions are restricted to translations [KK95].

MONOTONE TWO-HANDED ASSEMBLY SEQUENCING

A popular approach to assembly sequencing is disassembly sequencing [HS91]. A sequence that separates an assembly into its individual components is first generated and next reversed. Most existing assembly sequencers can only generate two-handed monotone sequences. Such a sequence is computed by partitioning the assembly and, recursively, the resulting subassemblies into two separated subassemblies.

The **nondirectional blocking graph** (NDBG) is proposed in [WL95] to represent all the blocking relations in an assembly. It is a subdivision of the space of all allowable motions of separation into a finite number of cells such that within each cell the set of blocking relations between all pairs of parts remains fixed. Within each cell this set is represented in the form of a directed graph, called the directional blocking graph (DBG). The NDBG is the collection of the DBGs over all the cells in the subdivision.

We illustrate this approach for polyhedral assemblies when the allowable motions are infinite translations. The partitioning of an assembly consisting of polyhedral parts into two subassemblies is performed as follows. For an ordered pair of parts P_i, P_j , the 3-vector \vec{d} is a **blocking direction** if translating P_i to infinity in

direction \vec{d} will cause P_i to collide with P_j . For each ordered pair of parts, the set of blocking directions is constructed on the unit sphere S^2 by drawing the boundary arcs of the union of the blocking directions (each arc is a portion of a great circle). The resulting collection of arcs partitions S^2 into maximal regions such that the blocking relation among the parts is the same for any direction inside such a region.

Next, the blocking graph is computed for one such maximal region. The algorithm then moves to an adjacent region and updates the DBG by the blocking relations that change at the boundary between the regions, and so on. After each time the construction of a DBG is completed, this graph is checked for strong connectivity in time linear in its number of edges. The algorithm stops the first time it encounters a DBG that is not strongly connected and it outputs the two subassemblies of the partitioning. The overall sequencing algorithm continues recursively with the resulting subassemblies. If all the DBG's that are produced during a partitioning step are strongly connected, the algorithm reports that the assembly does not admit a two-handed monotone assembly sequence with infinite translations.

Polynomial-time algorithms are proposed in [WL95] to compute and exploit NDBG's for restricted families of motions. In particular, the case of partitioning a polyhedral assembly by a single translation to infinity is analyzed in detail, and it is shown that partitioning an assembly of m polyhedra with a total of v vertices takes $O(m^2v^4)$ time. Another case studied in [WL95] is where the separating motions are infinitesimal rigid motions. Then partitioning the polyhedral assembly takes $O(mc^5)$ time, where m is the number of pairs of parts in contact and c is the number of independent point-plane contact constraints. (This result is improved in [GHH⁺98] by using the concept of maximally covered cells; see [Section 24.6](#).) Using these algorithms, every feasible disassembly sequence can be generated in polynomial time.

In [WL95], NDBG's are defined only for simple families of separating motions (infinitesimal rigid motions and infinite translations). An extension, called the *interference diagram*, is proposed in [WKL⁺95] for more complex motions. In the worst case, however, this diagram yields a partitioning algorithm that is exponential in the number of surfaces describing the assembly. When each separating motion is restricted to be a short sequence of concatenated translations (for example, a finite translation followed by an infinite translation), rather efficient partitioning algorithms are available [HW95]. A unified and general framework for assembly planning, based on the NDBG, called the *motion space approach* is presented in [HLW00].

OPEN PROBLEM

The complexity of the NDBG grows exponentially with the number of parameters that control the allowable motions, making this approach highly time consuming for assembly sequencing with compound motions. For the case of infinitesimal rigid motion it has been observed that only a (relatively small) subset of the NDBG needs to be constructed [GHH⁺98]. Are there additional types of motion where similar gains can be made? Are there situations where the full NDBG (or a structure of comparable size) must be constructed?

48.4 PATH PLANNING

Motion planning is aimed at providing robots with the capability of deciding automatically which motions to execute in order to achieve goals specified by spatial arrangements of physical objects. It arises in a variety of forms. The simplest form—the *basic path planning problem*—requires finding a geometric collision-free path for a single robot in a known static workspace. The path is represented by an arc connecting two points in the robot’s configuration space [LP83]. This arc must not intersect a forbidden region, the *C-obstacle region*, which is the image of the workspace obstacles. Other motion planning problems require dealing with moving obstacles, multiple robots, movable objects, uncertainty, etc.

In this section we consider basic path planning. In the next one we review other motion planning problems. Most of our presentation focuses on practical methods. See [Chapter 47](#) for a more extensive review of theoretical motion planning.

GLOSSARY

Path: A continuous map $\tau : [0, 1] \rightarrow \mathcal{C}$.

Obstacle: A workspace $W \subset \mathbb{R}^k$ is often defined by a set of obstacles B_i , $i = 1, \dots, q$, such that $W = \mathbb{R}^k \setminus \bigcup_1^q B_i$.

C-obstacle: Given an obstacle B_i , the subset $CB_i \subseteq \mathcal{C}$ such that, for any $\mathbf{q} \in CB_i$, $A(\mathbf{q})$ intersects B_i .

C-obstacle region: The union $CB = \cup_i CB_i$ plus the configurations that violate the mechanical limits of the robot’s joints.

Free space: The complement of the C-obstacle region in \mathcal{C} , $\mathcal{F} = \mathcal{C} \setminus CB$.

Free path: A path in free space.

Semifree path: A path in the complement of the union of the interior of *C*-obstacles.

Basic path planning problem: Compute a free or semifree path between two input configurations.

Path planning query: Given two points in configuration space find a (semi)free path between them. The term is often used in connection with algorithms that preprocess the configuration space in preparation for many queries.

Complete algorithm: A motion planning algorithm is complete if it is guaranteed to find a (semi)free path between two given configurations whenever such a path exists, and report that there is no (semi)free path otherwise. Complete algorithms are sometimes referred to as *exact* algorithms. There are weaker variants of completeness, for example, *probabilistic completeness*.

COMPLETE ALGORITHMS

Basic path planning for a 3D linkage made of polyhedral links is PSPACE-hard (Theorem 47.1.3c). The proof provides strong evidence that any complete algorithm will require exponential time in the number of dofs. This result remains true in more

specific cases, e.g., when the robot is a planar arm in which all joints are revolute (Theorem 47.1.3b). However, it no longer holds in some very simple settings; for instance, planning the path of a planar arm within an empty circle is in P [HJW85].

Two kinds of complete planners have been proposed: general ones, which apply to virtually any robot with an arbitrary number of dofs, and specific ones, which apply to a restricted family of robots usually having a fixed small number of dofs. The general “roadmap” algorithm in [Can88] is singly-exponential in the dimension of \mathcal{C} and polynomial in both the number of polynomial constraints defining the free space and their maximal degree (Theorem 47.1.2). Specific algorithms have been developed mainly for robots with 2 or 3 dofs. For a k -sided polygonal robot moving freely in a polygonal workspace, the algorithm in [HS96] takes $O((kn)^{2+\epsilon})$ time, where n is the total number of edges of the workspace (Theorem 47.2.10).

PROBABILISTIC ALGORITHMS

The complexity of path planning for robots with many dofs (more than 4 or 5) has led to the development of computational schemes that attempt to trade off completeness against time. One such scheme, *probabilistic planning* [BKL⁺97], avoids computing an explicit geometric representation of the free space. Instead, it uses an efficient procedure to compute distances between bodies in the workspace. It samples the configuration space by selecting a large number of configurations at random and retaining only the free configurations as *milestones*. It then checks if each pair of milestones can be connected by a collision-free straight path in configuration space. This computation yields the graph (V, E) , called a *probabilistic roadmap*, where V is the set of milestones and E is the set of pairs of milestones that have been connected.

Various strategies can be applied to sample the configuration space. The strategy in [KŠLO96] proceeds as sketched above. Once a roadmap has been precomputed, it is used to process an arbitrary number of path planning queries. Other sampling strategies [BL91, HLM99] assume that the initial and goal configurations are given, and incrementally build a roadmap until these two configurations are connected.

The results reported in [KLMR98, HLM99] bound the number of milestones generated by probabilistic-roadmap planners, under the assumption that the free space \mathcal{F} satisfies some geometric properties. One such property, called **expansiveness**, measures the difficulty caused by the presence of “narrow passages.” Let S be a subset of \mathcal{F} . The *lookout* of S is the set of all points in S that see a significant fraction of the volume of $\mathcal{F} \setminus S$ (the complement of S in \mathcal{F}). The lookout of S is “large” if its volume is a significant fraction of the volume of S . \mathcal{F} is said to be expansive if its subsets have large lookouts. If \mathcal{F} is expansive, the probability that a probabilistic-roadmap planner fails to find a free path between two given configurations, while one exists, goes to 0 exponentially in the number of milestones.

Recent research has focused on designing efficient sampling and connection strategies. For instance, the Gaussian sampling strategy produces a greater density of milestones near the boundary of the free space \mathcal{F} , whose connectivity is usually more difficult to capture by a roadmap than wide-open areas of \mathcal{F} [BOvS99]. Different methods to create milestones near the boundary of \mathcal{F} were obtained in [ABD⁺98]. A lazy-evaluation of the roadmap has been suggested in [BK00, SL02] while visibility has been exploited in [SL01]. Sampling and connection strategies

are reviewed in [SL02]. While some planners are better geared toward searching the whole \mathcal{F} (e.g., [KŠLO96]), others focus on answering single queries very efficiently (e.g., [BK00, SL02, LK01c]).

Probabilistic-roadmap techniques have also been used to compute collision-free trajectories taking dynamic constraints (e.g., bounded torques of actuators) into account [HKLR01, LK01c], and to plan manipulation and locomotion paths of humanoid robots under stability constraints [KNK⁺01]. The techniques have also been used for planning for nonholonomic systems [ŠO97, HKLR01] (see also Section 48.5.2).

Applications of probabilistic planning include the maintenance of aircraft engines, the riveting of aircraft fuselages, design automation (by ensuring correctness and maintainability of products from their CAD models), the programming of automotive assembly lines, the generation of aggressive maneuvers for autonomous helicopters, the generation of reconfiguration strategies for modular robots, the generation of motions in contact, and computer animation. Recent work has applied randomized path planning techniques to planning for flexible objects [LK01a] and to the computation of protein folding pathways and molecular motion [ADS02, ABG⁺02].

HEURISTIC ALGORITHMS

Several heuristic techniques have been proposed to speed up path planning. Some of them work well in practice, but they usually offer no performance guarantee.

Heuristic algorithms often search a regular grid defined over the configuration space and generate a path as a sequence of adjacent grid points [Don87]. The search can be guided by a *potential field*, a function over the free space that has a global minimum at the goal configuration. This function may be constructed as the sum of an attractive and a repulsive field [Kha86]. The attractive field has a single minimum at the goal and grows to infinity as the distance to the goal increases. The repulsive field is null at all configurations where the distance between the robot and the obstacles is greater than some predefined value, and grows to infinity as the robot gets closer to an obstacle. Evaluating the repulsive field requires an efficient distance computation algorithm. The search is usually done by following the steepest descent of the potential function. Several techniques deal with local minima [BL91]. Potentials free of local minima have been proposed [RK92], but their computation is likely to be at least as expensive as path planning.

One may also construct grids at variable resolution. Hierarchical space decomposition techniques such as octrees and boxtrees have been used to that purpose [BH95]. At any decomposition level, each grid cell is labeled EMPTY, FULL, or MIXED depending on whether it lies entirely in the free space, lies in the C -obstacle region, or overlaps both. Only the MIXED cells are decomposed further, until a search algorithm finds a sequence of adjacent FREE cells connecting the initial and goal configurations.

DISTANCE COMPUTATION

The efficient computation of distances between two bodies is a crucial element of many path planners. Various algorithms have been proposed to compute distances

between two convex bodies. A numerical descent technique is described in [GJK88] to compute the distance between two convex polyhedra; experience indicates that it runs in approximately linear time in the total complexity of the polyhedra. See [Chapters 34](#) and 37 for related techniques.

In robotics applications one often needs to compute the minimum distance between two sets of bodies, one representing the robot, the other the obstacles. The cost of computing the distance between every pair of bodies can be prohibitive. Simple bounding volumes, often coupled with hierarchical decomposition techniques, have been used to reduce computation time [Qui94, GLM96]. When motion is involved, incremental distance computation has been suggested for tracking the closest points on a pair of convex polyhedra [LC91]. It takes advantage of the fact that the closest features (faces, edges, vertices) change infrequently as the polyhedra move along finely discretized paths. See [Chapter 35](#).

OPEN PROBLEMS

1. Design algorithms for probabilistic-roadmap planners capable of efficiently sampling milestones in narrow passages of the free space.
2. Implement effective complete solutions, namely exact algorithmic solutions that run reasonably fast. The CGAL library ([Chapter 65](#)) of geometric algorithms provides infrastructure for such development [Hal02]. For example, an exact solver for translational motion planning in the plane has already been developed on top of CGAL [Fla00].
3. Design algorithms to compute distance between rigid and continuously deformable objects (e.g., power cables).

48.5 OTHER MOTION PLANNING PROBLEMS

There are many useful extensions of the basic path planning problem. Several are surveyed in [Chapter 47](#), e.g., shortest paths, coordinated motion planning (multi-robot case), time-varying workspaces (moving obstacles), and exploratory motion planning. Below we focus on the following extensions: manipulation planning, nonholonomic robots, uncertainty, and optimal planning.

GLOSSARY

Movable object: Body that can be grasped and moved by a robot.

Manipulation planning: Motion planning with movable objects.

Trajectory: Path parameterized by time.

Tangent space: Given a smooth manifold M and a point $p \in M$, the vector space $T_p(M)$ spanned by the tangents at p to all smooth curves passing through p and contained in M . The tangent space has the same dimension as M .

Nonholonomic robot: Robot whose permissible velocities at every configuration \mathbf{q} span a subset $\Omega(\mathbf{q})$ of the tangent space $T_{\mathbf{q}}(\mathcal{C})$ of lower dimension. Ω is called the *set of controls* of the robot.

Feasible path: A piecewise differentiable path of a nonholonomic robot whose tangent at every point belongs to the robot's set of controls, i.e., satisfies the nonholonomic velocity constraints.

Locally controllable robot: A nonholonomic robot is locally controllable if for every configuration \mathbf{q}_0 and any configuration \mathbf{q}_1 in a neighborhood U of \mathbf{q}_0 , there exists a feasible path connecting \mathbf{q}_0 to \mathbf{q}_1 which is entirely contained in U .

Uncertainty in control and sensing: Distributions of control and position sensing errors over multiple executions.

Landmark: Workspace feature that the robot may reliably sense and use to precisely localize itself. The region of configuration space from which the robot can sense a landmark is called a *landmark area*.

Kinodynamic planning: Find a minimal-time trajectory between two given configurations of a robot, given the robot's dynamic equation of motion.

48.5.1 MANIPULATION PLANNING

Many robot tasks consist of achieving arrangements of physical objects. Such objects, called movable objects, cannot move autonomously; they must be grasped by a robot. Planning with movable objects is called manipulation planning.

In [Wil91] the robot A and the movable object M are both convex polygons in a polygonal workspace. The goal is to bring A and M to specified positions. A can only translate. To grasp M , A must have one of its edges that exactly coincides with an edge of M . While A grasps M , they move together as one rigid object. An exact cell decomposition algorithm is given that runs in $O(n^2)$ time after $O(n^3 \log^2 n)$ preprocessing, where n is the total number of edges in the workspace, the robot, and the movable object. An extension of this problem allowing an infinite set of grasps is solved by an exact cell decomposition algorithm in [ALS95].

Heuristic algorithms have also been proposed. The planner in [KL94] first plans the path of the movable object M . During that phase, it verifies only that for every configuration taken by M there exists at least one collision-free configuration of the robot where it can hold M . In the second phase, the planner determines the points along the path of M where the robot must change grasps. It then computes the paths where the robot moves alone (transit paths) to (re)grasp M . The paths of the robot when it carries M (transfer paths) are obtained through inverse kinematics. This planner is not complete, but it has solved complex tasks in practice. Probabilistic roadmap methods have also been used for manipulation planning [NK00]. Finally, of special interest are the efforts on planning for closed kinematic chains using probabilistic methods, manipulation that frequently leads to closed chains formed by two manipulators and the manipulated object [CSL02].

48.5.2 NONHOLONOMIC ROBOTS

The trajectories of a nonholonomic robot are constrained by $p \geq 1$ nonintegrable scalar equality constraints:

$$G(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = (G^1(\mathbf{q}(t), \dot{\mathbf{q}}(t)), \dots, G^p(\mathbf{q}(t), \dot{\mathbf{q}}(t))) = (0, \dots, 0),$$

where $\dot{\mathbf{q}}(t) \in T_{\mathbf{q}(t)}(\mathcal{C})$ designates the velocity vector along the trajectory $\mathbf{q}(t)$. At every \mathbf{q} , the function $G_{\mathbf{q}} = G(\mathbf{q}, \cdot)$ maps the tangent space $T_{\mathbf{q}}(\mathcal{C})$ into \mathbb{R}^p . If $G_{\mathbf{q}}$ is smooth and its Jacobian has full rank (two conditions that are often satisfied), the constraint $G_{\mathbf{q}}(\dot{\mathbf{q}}) = (0, \dots, 0)$ constrains $\dot{\mathbf{q}}$ to be in a linear subspace of $T_{\mathbf{q}}(\mathcal{C})$ of dimension $m - p$. The nonholonomic robot may also be subject to scalar inequality constraints of the form $H^j(\mathbf{q}, \dot{\mathbf{q}}) > 0$. The subset of $T_{\mathbf{q}}(\mathcal{C})$ that satisfies all the constraints on $\dot{\mathbf{q}}$ is called the set $\Omega(\mathbf{q})$ of controls at \mathbf{q} . A feasible path is a piecewise differentiable path whose tangent lies everywhere in the control set.

A car-like robot is a classical example of a nonholonomic robot. It is constrained by one equality constraint (the linear velocity points along the car's main axis). Limits on the steering angle impose two inequality constraints. Other nonholonomic robots include tractor-trailers, airplanes, and satellites.

Given an arbitrary subset $U \subset \mathcal{C}$, the configuration $\mathbf{q}_1 \in U$ is said to be *U-accessible* from $\mathbf{q}_0 \in U$ if there exists a piecewise constant control $\dot{\mathbf{q}}(t)$ in the control set whose integral is a trajectory joining \mathbf{q}_0 to \mathbf{q}_1 that lies fully in U . Let $A_U(\mathbf{q}_0)$ be the set of configurations *U-accessible* from \mathbf{q}_0 . The robot is said to be **locally controllable** at \mathbf{q}_0 iff for every neighborhood U of \mathbf{q}_0 , $A_U(\mathbf{q}_0)$ is also a neighborhood of \mathbf{q}_0 . It is locally controllable iff this is true for all $\mathbf{q}_0 \in \mathcal{C}$. Car-like robots and tractor-trailers that can go forward and backward are locally controllable [BL93].

Let X and Y be two smooth vector fields on \mathcal{C} . The Lie bracket of X and Y , denoted by $[X, Y]$, is the smooth vector field on \mathcal{C} defined by $[X, Y] = dY \cdot X - dX \cdot Y$, where dX and dY , respectively, denote the $m \times m$ matrices of the partial derivatives of the components of X and Y w.r.t. the configuration coordinates in a chart placed on \mathcal{C} . To get a better intuition of the Lie bracket, imagine a trajectory starting at an arbitrary configuration \mathbf{q}_s and obtained by concatenating four subtrajectories: the first is the integral curve of X during time δt ; the second, third, and fourth are the integral curves of Y , $-X$, and $-Y$, respectively, each during the same δt . Let \mathbf{q}_f be the final configuration reached. A Taylor expansion yields:

$$\lim_{\delta t \rightarrow 0} \frac{\mathbf{q}_f - \mathbf{q}_s}{\delta t^2} = [X, Y].$$

Hence, if $[X, Y]$ is not a linear combination of X and Y , the above trajectory allows the robot to move away from \mathbf{q}_s in a direction that is not contained in the vector subspace defined by $X(\mathbf{q}_s)$ and $Y(\mathbf{q}_s)$. But the motion along this new direction is an order of magnitude slower than along any direction $\alpha X(\mathbf{q}_s) + \beta Y(\mathbf{q}_s)$.

The **control Lie algebra** associated with the control set Ω , denoted by $L(\Omega)$, is the space of all linear combinations of vector fields in Ω closed by the Lie bracket operation. The following result derives from the Controllability Rank Condition Theorem [BL93]:

A robot is locally controllable if, for every $\mathbf{q} \in \mathcal{C}$, $\Omega(\mathbf{q})$ is symmetric with respect to the origin of $T_{\mathbf{q}}(\mathcal{C})$ and the set $\{X(\mathbf{q}) \mid X \in L(\Omega(\mathbf{q}))\}$ has dimension m .

The minimal length of the Lie brackets required to construct $L(\Omega)$, when these brackets are expressed with vectors in Ω , is called the *degree of nonholonomy* of the robot. The degree of nonholonomy of a car-like robot is 2. Except at some singular configurations, the degree of nonholonomy of a tractor towing a chain of s trailers is $2+s$ [LR96]. Intuitively, the higher the degree of nonholonomy, the more complex (and the slower) the robot's maneuvers to perform some motions.

PLANNING FOR CONTROLLABLE ROBOTS

Let A be a locally controllable nonholonomic robot. A necessary and sufficient condition for the existence of a feasible free path of A between two given configurations is that they lie in the same connected component of the *open* free space. Indeed, local controllability guarantees that a possibly nonfeasible path can be decomposed into a finite number of subpaths, each short enough to be replaced by a feasible free subpath. Hence, deciding if there exists a free path for a locally controllable nonholonomic robot has the same complexity as deciding if there exists a path for the holonomic robot having the same geometry.

Transforming a nonfeasible free path τ into a feasible one can be done by recursively decomposing τ into subpaths. The recursion halts at every subpath that can be replaced by a feasible free subpath. Specific substitution rules (e.g., Reeds and Shepp curves) have been defined for car-like robots [LJTM94]. The complexity of transforming a nonfeasible free path τ into a feasible one is of the form $O(\epsilon^d)$, where ϵ is the smallest clearance between the robot and the obstacles along τ and d is the degree of nonholonomy of the robot (see [LJTM94] for the case $d = 2$).

The algorithm in [BL93] directly constructs a nonholonomic path for a car-like or a tractor-trailer robot by searching a tree obtained by concatenating short feasible paths, starting at the robot's initial configuration. The planner is *asymptotically complete*, i.e., it is guaranteed to find a path if one exists, provided that the lengths of the short feasible paths are small enough. It can also find paths that minimize the number of cusps (changes of sign of the linear velocity).

PLANNING FOR NONCONTROLLABLE ROBOTS

Path planning for nonholonomic robots that are not locally controllable is much less understood. Research has almost exclusively focused on car-like robots that can only move forward. Results include:

- No obstacles: A complete synthesis of the shortest, no-cusp path for a point moving with a lower-bounded turning radius [SL93].
- Polygonal obstacles: An algorithm to decide whether there exists such a path between two configurations; it runs in time exponential in obstacle complexity [FW88].
- Convex obstacles: The algorithm in [ART95] computes a path in polynomial time under the assumptions that all obstacles are convex and their boundaries have a curvature radius greater than the minimum turning radius of the point.
- Other polynomial algorithms (e.g., [BL93]) require some sort of discretization and are only asymptotically complete.

OPEN PROBLEM

Establish a nontrivial lower bound on the complexity of planning for a nonholonomic robot that is not locally controllable.

48.5.3 UNCERTAINTY

In practice, robots deviate from planned paths due to errors in control and position sensing. A motion planning problem with uncertainty can be formulated as follows:

Input. The inputs are the initial region $I \subset \mathcal{C}$, in which the robot is known to be prior to moving, the goal region $G \subset \mathcal{C}$, in which it should terminate its motion, and the uncertainty in control and sensing. Uncertainty is specified in the form of regions. For instance, the uncertainty in position sensing is the set of actual robot configurations that are possible given the sensor readings.

Output. The output is a series of motion commands, if one exists, whose execution enables the robot to reach G from I . Each command is a velocity vector \mathbf{v} and a termination condition T . The vector \mathbf{v} specifies the desired behavior of the robot over time (with or without compliance). The condition T is a Boolean function of the sensor readings and time which causes the motion to stop as soon as it becomes true. A plan may contain conditional branchings.

This problem is NEXPTIME-hard for a point robot moving in 3-space among polyhedral obstacles [CR87].

PREIMAGE OF A GOAL

Given a goal G and a command (\mathbf{v}, T) , a preimage of G is any region $P \subset \mathcal{C}$ such that executing the command from anywhere in P makes the robot reach and stop in G [LPMT84]. One way to compute a (nonmaximal) preimage is to restrict the termination condition so that it recognizes G independently of the region from which the motion started [Erd86]. For example, one may shrink G to a subset K , called the *kernel* of G , such that whenever the robot is in K , all robot configurations consistent with the current sensor readings are in G . A preimage is then computed as the region from which the robot commanded along \mathbf{v} is guaranteed to reach K . This region is called the *backprojection* of K for \mathbf{v} . This preimage computation has been well studied in a polygonal configuration space with G a polygon [Lat91].

ONE-STEP PLANNING

In a polygonal configuration space, the kernel of a polygonal goal is either independent of the selected \mathbf{v} or changes at a number of critical orientations of \mathbf{v} that is linear in the workspace complexity [Lat91]. Moreover, the backprojection of a polygonal region K , when the orientation of \mathbf{v} varies, changes topology only at a quadratic number of critical directions. Its intersection with a polygonal initial region I of constant complexity also changes qualitatively at few directions of \mathbf{v} . Checking the containment of I by the backprojection at each such direction yields a one-step motion plan, if one exists, in amortized time $O(n^2 \log n)$, where n are the edges in \mathcal{C} [Bri95]. In [dBGH⁺95] the computational complexity of solving certain one-step planning problems is expressed also in terms of the size of the control error.

MULTI-STEP PLANNING

For multi-step planning, algebraic approaches that check the satisfiability of a first-order semialgebraic formula have been proposed. In [Can89] it is assumed that all possible trajectories have an algebraic description. The approach there is based on a two-player game interpretation of planning, where the robot is one player and nature the other. Each step of a plan contributes three quantifiers: one existential quantifier applies to the direction of motion, and corresponds to choosing this direction; another existential quantifier applies to time, and corresponds to choosing when to terminate the motion; one universal quantifier applies to the sensor readings and represents the unknown action of nature. The formula representing an r -step plan thus contains r quantifier alternations; checking its satisfiability takes doubly-exponential time in r , which is itself polynomial in the total complexity of the robot and the workspace.

LANDMARK-BASED PLANNING

Often a workspace contains features that can be reliably sensed and used to precisely localize the robot. Each such landmark feature induces a region in configuration space called the *landmark area* from which the robot can sense the corresponding feature.

The planner described in [LL95] considers a point robot among n circular obstacles and $O(n)$ circular landmark areas. It assumes perfect position sensing and motion control in landmark areas. Outside these areas, it assumes that the robot has no position sensing whatsoever and that directional errors in control are bounded by the angle θ . Given circular initial and goal regions I and G (with G intersecting at least one landmark area), the planner constructs a motion plan that enables the robot to move from landmark area to landmark area until it reaches the goal. It proceeds backward by computing the preimages of the landmark regions intersecting G , the preimages of the landmark regions intersected by these preimages, and so on, until a preimage contains I . The planner runs in $O(n^4 \log n)$ time; it is complete and generates plans that minimize the number of steps to be executed in the worst case.

48.5.4 OPTIMAL PLANNING

There has been considerable research on finding shortest paths (see [Chapter 27](#)), but minimal Euclidean length may not be the most suitable criterion in practice. One is often more interested in minimizing execution time, which requires dealing with the robot's dynamics.

OPTIMAL-TIME CONTROL PLANNING

The input is a (geometric) free path τ parameterized by $s \in [0, L]$, the distance traveled from the starting configuration. The problem is to find the time parametrization $s(t)$ that minimizes travel time along τ , while satisfying actuator limits.

The equation of motion of a robot arm with m dofs can be written as $M(\mathbf{q})\ddot{\mathbf{q}} + V(\dot{\mathbf{q}}, \mathbf{q}) + G(\mathbf{q}) = \Gamma$, where \mathbf{q} , $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$, respectively, denote the robot's configuration, velocity, and acceleration [Cra89]. M is the $m \times m$ inertia matrix of the robot, V the m -vector (quadratic in $\dot{\mathbf{q}}$) of centrifugal and Coriolis forces, and G the m -vector of gravity forces. Γ is the m -vector of the torques applied by the

joint actuators.

Using the fact that the robot follows τ , this equation can be rewritten in the form: $\mathbf{m}\ddot{s} + \mathbf{v}\dot{s}^2 + \mathbf{g} = \Gamma$, where \mathbf{m} , \mathbf{v} , and \mathbf{g} are derived from M , V , and G , respectively. Minimum-time control planning becomes a two-point boundary value problem: Find $s(t)$ that minimizes $t_f = \int_0^L ds/\dot{s}$, subject to $\Gamma = \mathbf{m}\ddot{s} + \mathbf{v}\dot{s}^2 + \mathbf{g}$, $\Gamma_{min} \leq \Gamma \leq \Gamma_{max}$, $s(0) = 0$, $s(t_f) = L$, and $\dot{s}(0) = \dot{s}(L) = 0$. Numerical techniques solve this problem by finely discretizing the path τ [BDG85].

MINIMAL-TIME TRAJECTORY PLANNING

Finding a minimal-time trajectory, called *kinodynamic motion planning*, is much more difficult. One approach is to first plan a geometric path and then iteratively deform this path to reduce travel time [SD91]. Each iteration requires checking the new path for collision and recomputing the optimal-time control. No bound has been established on the running time of this approach or the goodness of its outcome. Kinodynamic planning is NP-hard for a point robot under Newtonian mechanics in 3-space [DX95]. The approximation algorithm in [DXCR93] computes a trajectory ϵ -close to optimal in time polynomial in both $1/\epsilon$ and the workspace complexity.

Other optimality questions concern the layout of a robotic cell and in particular the optimal placement of robots inside the cell. Such problems bear resemblance to *facility location* problems; see, for example, an efficient solution to the problem of placing two robot arms in order to minimize the maximal horizontal stretch of an arm for a given collection of workpoints that the robots must reach [HSG02].

48.6 DATA STRUCTURES FOR MOVING OBJECTS

Robotics requires efficient algorithms to compute motions and/or to update properties of bodies as they move (e.g., distances to obstacles). Several data structures have been specifically proposed to represent moving bodies. The related study of *kinetic data structures* is described in [Chapter 50](#).

NONDIRECTIONAL DATA STRUCTURES

These data structures partition the space of possible motions into an arrangement of cells such that a given property remains satisfied over each cell. They are typically computed in a preprocessing step to speed up the treatment of subsequent queries.

For example, in the context of assembly sequencing (Section 48.3), a property of interest is how the parts in an assembly block one another for a certain family of motions. It yields the concepts of a nondirectional blocking graph and an interference diagram. In motion planning with uncertainty (Section 48.5.3), a similar concept is the nondirectional backprojection/preimage of a goal [Bri95, LL95]. As the direction of motion varies, the topology of a preimage changes only at critical values which define an arrangement of cells in the motion space. This arrangement, along with a preimage computed in each cell, forms the *nondirectional preimage*.

A related concept is used in [Gol93] to construct the possible orientations of a polygonal body after it has been squeezed by a parallel-jaw gripper (Section 48.2.3).

DYNAMIC MAINTENANCE OF KINEMATIC STRUCTURES

Several prototypes of highly flexible robots have been designed and constructed in recent years. Since the number of dofs in these new designs is far larger than in more traditional robots, they raise new algorithmic issues. Similar issues arise in computer simulation of large kinematic structures outside robotics, e.g., in molecular biology and in graphic animation of digital actors.

A basic problem in this domain can be phrased as follows. Given a linkage with many dofs, how can we efficiently maintain a data structure that allows us to quickly answer intersection (or range) queries as the bodies move. Several models for dynamic maintenance of such linkages are proposed in [HLM97], together with efficient maintenance algorithms. Tight results are given on the worst-case, amortized, and randomized complexity of this data structure problem. For the off-line version of the problem, NP-hardness is established and efficient approximation algorithms are provided.

Another basic problem is to efficiently detect collisions (cf. Chapter 35) of a kinematic chain with itself (“self collisions”), motivated primarily by Monte Carlo simulation of conformational change of polymers. Two variants of the problem have been addressed: (i) single joint, continuous motion—detecting self-collision while continuously modifying one degree of freedom of the chain [ST00]. This variant was shown to defy preprocessing that would lead to efficient query answering [SEO03]. (ii) Several joints, discrete modification—changing a small number of degrees of freedom and testing statically for self-collision at the new configuration [LHSL02]. A data structure that combines bounding volume hierarchy and a hierarchy of transformations over the links of the chain was shown to perform very well in practice, with guaranteed theoretical resource bounds.

48.7 SENSING

Sensing allows a robot to acquire information about its workspace and to localize itself. A wide variety of sensors are available and provide raw data of different types, such as time of flight, light intensity, color, or force. Preprocessing these data yield more directly usable information, e.g., geometric information, which can then be exploited to perform such tasks as model construction, object identification, and robot localization. Vision sensors are the most widely used sensors. Many textbooks focus on the role of geometry in computer vision, e.g., [Gri90]. Touch and force sensors are important to detect and characterize contacts among objects, for instance in manipulation tasks. Sensing is a wide domain of research with many subareas and challenging problems. Here we mention only a few selected topics.

MODEL BUILDING

Consider a mobile robot in an unknown workspace W . A first task for this robot is likely to be the construction of a geometric model (also called a map) of W . This requires the robot to perform a series of sensing operations at different locations. Each operation yields a partial model. The robot must patch together the succes-

sively obtained partial models to eventually form a complete map of the workspace. This problem is complicated by the fact that the robot has imperfect control and cannot accurately keep track of its position in a fixed coordinate system. See, e.g., [ZF96].

Recently model building has led to two families of methods, SLAM and NBV. In SLAM (for *Simultaneous Localization and Sensing*), probability distributions are computed and combined to best localize the robot(s) with respect to the partial map built so far and to patch this map with newly acquired data [DWDG00]. In NBV (for *Next Best View*), geometric visibility algorithms are used to compute where the robot should move next in order to acquire the “best” new data [GBL02b].

ROBOT LOCALIZATION

A robot often has to localize itself relative to its workspace W . A model of W is given and localization is done by matching sensory inputs against this model to infer the transform that defines the robot configuration. This problem usually arises for mobile robots. Other types of robots, such as robot arms, often have absolute references (e.g., mechanical stops) and internal sensors (e.g., joint encoders) that provide configurations more directly. Mobile robots have wheel encoders allowing dead-reckoning, but the absence of absolute reference on the one hand and slipping on the ground on the other hand usually necessitate sensor-based localization. GPS (Global Positioning System) has recently become a more widely available alternative, but it does not work in all environments.

Two kinds of sensor-based localization problems can be distinguished, *static* and *dynamic*. In the static problem, the robot is placed at an arbitrary unknown configuration and the problem is to compute this configuration. In the dynamic problem, the robot moves continuously and must regularly update its configuration. The second problem consists of refining an available estimate of the current configuration; here the computation must be done in real time. The static problem is usually more complex, but computation time is less restricted. A preprocessing approach to the static localization problem for a point robot equipped with a 360° range sensor is discussed in Section 47.3. Practical techniques for localization are also available, e.g., [TA96]. Probabilistic methods (particle filtering) have also been successfully applied to the dynamic localization problem for one or several robots [FBKT00]. Localization using wireless Ethernet has been explored in [LBM⁺02].

PURSUIT-EVASION

The problem here is for a team of robots (called pursuers) equipped with visual sensors to find a moving target in an environment of given geometry. The solution is a set of coordinated paths such that one pursuer is eventually guaranteed to see the target. In a polygonal environment with n edges and h holes, it has been shown that the minimum number of pursuers needed is $\Omega(\sqrt{h} + \log n)$ and $O(h + \log n)$. If $h = 0$, it is $\Theta(\log n)$. Computing the actual minimum number of pursuers is NP-hard. See [SY92, GLL⁺99, LSS02].

ADDITIONAL ISSUES IN SENSING

Sensor placement is the problem of computing the set of placements from which a sensor (or guard) can monitor a region within a given workspace [Bri95]. Another problem is to choose a minimal set of sensors and their placement so as to completely cover a given region. This induces a family of art-gallery type problems (see [Section 28.1](#)) that vary according to the type of data that the sensors provide. For the case of visual sensors with realistic physical constraints, a practical randomized solution has been proposed that produces a good approximation of the minimal necessary number of guards [GBL02a]. In the case where each point sees a sizable fraction of the gallery, bounds on the number of guards are given in [Val98, Val99]. Interestingly, the latter results were motivated by questions in randomized motion planning [KLMR98].

There has been considerable interest in recognizing and reconstructing shapes of objects using simple sensors. So-called *probes*, described in [Chapter 29](#), provide a convenient abstraction for the case where the robot takes a discrete number of measurements. There is also work on combining shape reconstruction with manipulation; see e.g., [BMP99, ME02a]. *Matching* and *aspect graphs* (Section 28.6.3) are two related topics that have been well studied, mainly in computer vision.

48.8 SOURCES AND RELATED MATERIAL

Craig's book [Cra89] provides an introduction to robot arm kinematics, dynamics, and control. For advanced kinematics see the book by Bottema and Roth [BR79]. Robot motion planning and its variants are discussed in Latombe's book [Lat91]. This book takes an algorithmic approach to a variety of advanced issues in robotics (not restricted to robot arms). The mechanics of robotic manipulation is covered in Mason's book [Mas01].

The proceedings series of the *International Symposium on Robotics Research* gives state-of-the-art presentations of robotics in general (e.g., [GH96] and subsequent volumes). The proceedings of the *Workshop on Algorithmic Foundations of Robotics* (WAFR)—see [GHLW95] and subsequent volumes—emphasize algorithmic issues in robotics.

Several computational geometry books contain sections on robotics or motion planning [O'R98, SA95, dBvK⁺00].

RELATED CHAPTERS

[Chapter 24: Arrangements](#)

[Chapter 27: Shortest paths](#)

[Chapter 28: Visibility](#)

[Chapter 29: Geometric reconstruction problems](#)

[Chapter 33: Computational real algebraic geometry](#)

[Chapter 35: Collision detection](#)

[Chapter 47: Algorithmic motion planning](#)

[Chapter 50: Motion](#)

[Chapter 59: Geometric applications of the Grassmann-Cayley algebra](#)

REFERENCES

- [ABD⁺98] N.M. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In P.K. Agarwal, L.E. Kavraki, and M.T. Mason, editors, *Robotics: The Algorithmic Perspective*. A.K. Peters, Wellesley, 1998.
- [ABG⁺02] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, and J.-C. Latombe. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. In *Proc. 6th Internat. Conf. Comput. Molecular Biology*, pages 12–21, 2002.
- [ADS02] N.M. Amato, K. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. In *Proc. 6th Internat. Conf. Comput. Molecular Biology*, pages 2–11, 2002.
- [AHLM00] S. Akella, W. Huang, K. Lynch, and M.T. Mason. Parts feeding on a conveyor with a one joint robot. *Algorithmica (Special Issue on Robotics)*, 26(3/4):313–344, 2000.
- [ALS95] R. Alami, J.-P. Laumond, and T. Siméon. Two manipulation planning algorithms. In Goldberg et al. [GHLW95], pages 109–125.
- [ART95] P.K. Agarwal, P. Raghavan, and H. Tamaki. Motion planning for a steering-constrained robot through moderate obstacles. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 343–352, 1995.
- [BDG85] J.E. Bobrow, S. Dubowsky, and J.S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Internat. J. Robot. Res.*, 4:3–17, 1985.
- [BDKL00] K.-F. Böhringer, B.R. Donald, L.E. Kavraki, and F. Lamiraux. Part orientation to one or two stable equilibria using programmable force fields. *IEEE Trans. Robot. Autom.*, 16:731–747, 2000.
- [BDM99] K.-F. Böhringer, B.R. Donald, and N. MacDonald. Programmable vector fields for distributed manipulation, with application to MEMS actuator arrays and vibratory part feeders. *Internat. J. Robot. Res.*, 18:168–200, 1999.
- [BG96] R.C. Brost and K.Y. Goldberg. A complete algorithm for designing planar fixtures using modular components. *IEEE Trans. Syst. Man Cybern.*, 12:31–46, 1996.
- [BGO⁺98] R.-P. Berretty, K.Y. Goldberg, M.H. Overmars, and A.F. van der Stappen. Computing fence designs for orienting parts. *Comput. Geom. Theory Appl.*, 10:249–262, 1998.
- [BH95] M. Barbehenn and S. Hutchinson. Efficient search and hierarchical motion planning by dynamically maintaining single-source shortest paths trees. *IEEE Trans. Robot. Autom.*, 11:198–214, 1995.
- [BK00] R. Bohlin and L.E. Kavraki. Path planning using lazy prm. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 521–528, 2000.
- [BKL⁺97] J. Barraquand, L.E. Kavraki, J.-C. Latombe, T.-Y. Li, R. Motwani, and P. Raghavan. A random sampling framework for path planning in large-dimensional configuration spaces. *Internat. J. Robot. Res.*, 16:759–774, 1997.
- [BL91] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Internat. J. Robot. Res.*, 10:628–649, 1991.
- [BL93] J. Barraquand and J.-C. Latombe. Nonholonomic multibody mobile robots: Controllability and motion planning in the presence of obstacles. *Algorithmica*, 10:121–155, 1993.
- [BMP99] A. Bicchi, A. Marigo, and D. Prattichizzo. Dexterity through rolling: Manipulation of unknown objects. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1583–1588, Detroit, Michigan, 1999.

- [BOvS99] V. Boor, M.H. Overmars, and A.F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1018–1023, 1999.
- [BR79] O. Bottema and B. Roth. *Theoretical Kinematics*. North Holland, Amsterdam, 1979.
- [Bri95] A.J. Briggs. Efficient geometric algorithms for robot sensing and control. Report 95-1480, Dept. of Computer Science, Cornell Univ., Ithaca, 1995.
- [Can88] J.F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, 1988.
- [Can89] J.F. Canny. On computability of fine motion plans. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 177–182, 1989.
- [CI95] Y.-B. Chen and D.J. Ierardi. The complexity of oblivious plans for orienting and distinguishing polygonal parts. *Algorithmica*, 14:367–397, 1995.
- [CR87] J.F. Canny and J.H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.
- [Cra89] J.J. Craig. *Introduction to Robotics. Mechanics and Control*, 2nd edition. Addison-Wesley, Reading, 1989.
- [CSL02] J. Cortes, T. Siméon, and J.-P. Laumond. A random loop generator for planning the motions of closed kinematic chains using PRM methods. In *Proc. IEEE Internat. Conf. Robot. Autom.*, Washington, 2002.
- [dBGH⁺95] M. de Berg, L.J. Guibas, D. Halperin, M.H. Overmars, O. Schwarzkopf, M. Sharir, and M. Teillaud. Reaching a goal with directional uncertainty. *Theoret. Comput. Sci.*, 140:301–317, 1995.
- [dBvK⁺00] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*, 2nd edition. Springer-Verlag, Berlin, 2000.
- [Don87] B.R. Donald. A search algorithm for motion planning with six degrees of freedom. *Artif. Intell.*, 31:295–353, 1987.
- [DWDG00] H.F. Durrant-Whyte, M.W.M.G. Dissanayake, and P.W. Gibbens. Toward deployment of large scale simultaneous localisation and map building (SLAM) systems. In J.M. Hollerbach and D.E. Koditschek, editors, *Robotics Research—The 9th Internat. Sympo.*, pages 161–167, Springer-Verlag, New York, 2000.
- [DX95] B.R. Donald and P. Xavier. Provably good approximation algorithms for optimal kinodynamic planning: Robots with decoupled dynamics bounds. *Algorithmica*, 14:443–479, 1995.
- [DXCR93] B.R. Donald, P. Xavier, J.F. Canny, and J.H. Reif. Kinodynamic motion planning. *J. Assoc. Comput. Mach.*, 40:1048–1066, 1993.
- [EM88] M.A. Erdmann and M.T. Mason. An exploration of sensorless manipulation. *IEEE Trans. Robot. Autom.*, 4:369–379, 1988.
- [EMV93] M.A. Erdmann, M.T. Mason, and G. Vaněček, Jr. Mechanical parts orienting: The case of a polyhedron on a table. *Algorithmica*, 10:226–247, 1993.
- [Epp90] D. Eppstein. Reset sequences for monotonic automata. *SIAM J. Computing*, 19:500–510, 1990.
- [Erd86] M. Erdmann. Using backprojections for fine motion planning with uncertainty. *Internat. J. Robot. Res.*, 5:19–45, 1986.
- [FBKT00] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Efficient multi-robot localization based on Monte Carlo approximation. In J.M. Hollerbach and D.E. Koditschek, editors,

- Robotics Research—The 9th Internat. Sympos.*, pages 153–160, Springer-Verlag, New York, 2000.
- [FC92] C. Ferrari and J.F. Canny. Planning optimal grasps. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 2290–2295, 1992.
 - [Fla00] E. Flato. Robust and efficient construction of planar Minkowski sums. Master’s thesis, Dept. Comput. Sci., Tel-Aviv Univ., 2000. <http://www.cs.tau.ac.il/~flato>.
 - [FW88] S.J. Fortune and G. Wilfong. Planning constrained motions. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 445–459, 1988.
 - [GBL02a] H.H. González-Baños and J.-C. Latombe. A randomized art-gallery algorithm for sensor placement. *Internat. J. Robotics. Res.*, 21: 829–848, 2002.
 - [GBL02b] H.H. González-Baños and J.-C. Latombe. Navigation strategies for exploring indoor environments. *Internat. J. Robot. Res.*, 2002.
 - [GH96] G. Giralt and G. Hirzinger, editors. *Robotics Research*. Springer-Verlag, Berlin, 1996.
 - [GHH⁺98] L.J. Guibas, D. Halperin, H. Hirukawa, J.-C. Latombe, and R.H. Wilson. Polyhedral assembly partitioning using maximally covered cells in arrangements of convex polytopes. *Internat. J. Comput. Geom. Appl.*, 8:179–200, 1998.
 - [GHLW95] K.Y. Goldberg, D. Halperin, J.-C. Latombe, and R.H. Wilson, editors. *Algorithmic Foundations of Robotics*, A.K. Peters, Wellesley, 1995.
 - [GJK88] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing distance between complex objects in three-dimensional space. *IEEE Trans. Robot. Autom.*, 4:193–203, 1988.
 - [GLL⁺99] L.J. Guibas, J.-C. Latombe, S.M. LaValle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *Internat. J. Comput. Geom. Appl.*, 9:471–494, 1999.
 - [GLM96] S. Gottschalk, M.C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. In *Proc. ACM Conf. SIGGRAPH 96*, pages 171–180, 1996.
 - [Gol93] K.Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10:201–225, 1993.
 - [Gri90] W.E.L. Grimson. *Object Recognition by Computer. The Role of Geometric Constraints*. MIT Press, Cambridge, 1990.
 - [Hal02] D. Halperin. Robust geometric computing in motion. *Internat. J. Robot. Res.*, 21:219–232, 2002.
 - [HJW85] J.E. Hopcroft, D.A. Joseph, and S.H. Whitesides. On the movement of robot arms in 2-dimensional bounded regions. *SIAM J. Computing*, 14:315–333, 1985.
 - [HKLR01] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. In B.R. Donald, K.M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics*, pages 247–264. A.K. Peters, Wellesley, 2001.
 - [HLM97] D. Halperin, J.-C. Latombe, and R. Motwani. Dynamic maintenance of kinematic structures. In J.-P. Laumond and M.H. Overmars, editors, *Algorithmic Foundations of Robotics*, pages 155–170. A.K. Peters, Wellesley, 1997.
 - [HLM99] D. Hsu, J.-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Internat. J. Comput. Geom. Appl.*, 9(4–5):495–512, 1999.
 - [HLW00] D. Halperin, J.-C. Latombe, and R.H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26:577–601, 2000.

- [HS91] L.S. Homem de Mello and A.C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Trans. Robot. Autom.*, 7:228–240, 1991.
- [HS96] D. Halperin and M. Sharir. A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. *Discrete Comput. Geom.*, 16:121–134, 1996.
- [HSG02] D. Halperin, M. Sharir, and K.Y. Goldberg. The 2-center problem with obstacles. *J. Algorithms*, 42:109–134, 2002.
- [HW95] D. Halperin and R.H. Wilson. Assembly partitioning along simple paths: the case of multiple translations. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1585–1593, 1995.
- [Kav97] L.E. Kavraki. Part orientation with programmable vector fields: Two stable equilibria for most parts. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 20–25, Albuquerque, 1997.
- [Kha86] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Internat. J. Robot. Res.*, 5:90–98, 1986.
- [KK95] L.E. Kavraki and M.N. Kolountzakis. Partitioning a planar assembly into two connected parts is NP-complete. *Inform. Process. Lett.*, 55:159–165, 1995.
- [KL94] Y. Koga and J.-C. Latombe. On multi-arm manipulation planning. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 945–952, 1994.
- [KLMR98] L.E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot motion planning. *J. Comput. Syst. Sci.*, 57:50–60, 1998.
- [KMY92] D.G. Kirkpatrick, B. Mishra, and C.K. Yap. Quantitative Steinitz’s theorem with applications to multifingered grasping. *Discrete Comput. Geom.*, 7:295–318, 1992.
- [KNK⁺01] J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue. Motion planning for humanoid robots under obstacle and dynamic balance constraints. In *Proc. IEEE Internat. Conf. Robot. Autom.*, Seoul, 2001.
- [Kol95] K. Kolarov. Algorithms for optimal design of robots in complex environment. In Goldberg et al. [GHLW95], pages 347–369.
- [KŠLO96] L.E. Kavraki, P. Švestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for fast path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580, 1996.
- [Lat91] J.-C. Latombe. *Robot Motion Planning*. Kluwer, Boston, 1991.
- [LBM⁺02] A. Ladd, K. Bekris, G. Marceau, A. Rudys, D. Wallach, and L.E. Kavraki. Using wireless internet for localization. In *Proc. IEEE/RJS Internat. Conf. Intell. Rob. Sys. (IROS)*. IEEE Press, 2002.
- [LC91] M.C. Lin and J.F. Canny. A fast algorithm for incremental distance computation. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1008–1014, 1991.
- [LJTM94] J.-P. Laumond, P.E. Jacobs, M. Taix, and R.M. Murray. A motion planner for non-holonomic mobile robots. *IEEE Trans. Robot. Autom.*, 10:577–593, 1994.
- [LK01a] F. Lamiraux and L.E. Kavraki. Planning paths for elastic objects under manipulation constraints. *Internat. J. Robot. Res.*, 20:188–208, 2001.
- [LK01b] F. Lamiraux and L.E. Kavraki. Positioning of symmetric and nonsymmetric parts using radial and constant fields: Computation of all equilibrium configurations. *Internat. J. Robot. Res.*, 20:635–659, 2001.
- [LK01c] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *Internat. J. Robot. Res.*, 20:278–300, 2001.

- [LL95] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. *Algorithmica*, 13:472–501, 1995.
- [LMC01] J.E. Luntz, W. Messner, and H. Choset. Distributed manipulation using discrete actuator arrays. *Internat. J. Robot. Res.*, 20:553–582, 2001.
- [LP83] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. Comput.*, 32:108–120, 1983.
- [LPMT84] T. Lozano-Pérez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *Internat. J. Robot. Res.*, 3:3–24, 1984.
- [LR96] J.-P. Laumond and J.J. Risler. Nonholonomic systems: controllability and complexity. *Theoret. Comput. Sci.*, 157:101–114, 1996.
- [LSHL02] I. Lotan, F. Schwarzer, D. Halperin, and J.-C. Latombe. Efficient maintenance and self-collision testing for kinematic chains. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 43–52, 2002.
- [LSS02] S.M. LaValle, B. Simov, and G. Slutzki. An algorithm for searching a polygonal region with a flashlight. *Internat. J. Comput. Geom. Appl.*, 12(1-2):87–113, 2002.
- [Mas82] M.T. Mason. *Manipulation by grasping and pushing operations*. Ph.D. thesis, MIT, Artificial Intelligence Lab., 1982.
- [Mas01] M.T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, Cambridge, 2001.
- [Mat95] R. Matikalli. *Mechanics Based Assembly Planning*. Ph.D. thesis, Carnegie Mellon Univ., 1995.
- [ME02a] M. Moll and M.A. Erdmann. Dynamic shape reconstruction using tactile sensors. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1636–1641, 2002.
- [ME02b] M. Moll and M.A. Erdmann. Manipulation of pose distributions. *Internat. J. Robot. Res.*, 21:277–292, 2002.
- [MGEF02] M. Moll, K.Y. Goldberg, M.A. Erdmann, and R. Fearing. Aligning parts for micro assemblies. *Assembly Automation*, 22:46–54, 2002.
- [Mis91] B. Mishra. Workholding: Analysis and planning. In *Proc. IEEE/SRJ Internat. Conf. Intelligent Robots Syst.*, pages 53–57, 1991.
- [Mis95] B. Mishra. Grasp metrics: Optimality and complexity. In Goldberg et al. [GHLW95], pages 137–165.
- [MNP90] X. Markenscoff, L. Ni, and C.H. Papadimitriou. The geometry of grasping. *Internat. J. Robot. Res.*, 9:61–74, 1990.
- [MSS87] B. Mishra, J.T. Schwartz, and M. Sharir. On the existence and synthesis of multifinger positive grips. *Algorithmica*, 2:541–558, 1987.
- [MZG⁺96] B. Mirtich, Y. Zhuang, K.Y. Goldberg, J.J. Craig, R. Zanutta, B. Carlisle, and J.F. Canny. Estimating pose statistics for robotic part feeders. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1140–1146, 1996.
- [Nat88] B.K. Natarajan. On planning assemblies. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 299–308, 1988.
- [Nat89] B.K. Natarajan. Some paradigms for the automated design of parts feeders. *Internat. J. Robot. Res.*, 8:98–109, 1989.
- [Ngu88] V.D. Nguyen. Constructing force-closure grasps. *Internat. J. Robot. Res.*, 7:3–16, 1988.
- [NK00] C.L. Nielsen and L.E. Kavraki. A two level fuzzy PRM for manipulation planning. In *Proc. IEEE/RSJ Internat. Conf. Intelligent Robots Syst.*, pages 1716–1722, Japan, 2000.

- [O'R98] J. O'Rourke. *Computational Geometry in C*, 2nd edition. Cambridge University Press, 1998.
- [ORSW95] M.H. Overmars, A.S. Rao, O. Schwarzkopf, and C. Wentink. Immobilizing polygons against a wall. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 29–38, 1995.
- [PSS⁺97] J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *Internat. J. Robot. Res.*, 16:11–35, 1997.
- [Qui94] S. Quinlan. Efficient distance computation between non-convex objects. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 3324–3329, 1994.
- [RB98] E. Rimon and J. Burdick. Mobility of bodies in contact—I: A 2nd order mobility index for multiple-finger grasps, *IEEE Trans. Robot. Autom.*, 14(5), 1998.
- [RG95] A.S. Rao and K.Y. Goldberg. Manipulating algebraic parts in the plane. *IEEE Trans. Robot. Autom.*, 11:598–602, 1995.
- [RK92] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.*, 8:501–518, 1992.
- [RMC00] D. Reznik, E. Moshkovich, and J.F. Canny. Building a universal planar manipulator. In K.-F. Böhringer and H. Choset, editors, *Distributed Manipulation*, pages 147–171. Kluwer Academic Publishers, Boston, 2000.
- [Rot94] B. Roth. Connections between robotic and classical mechanisms. In T. Kanade and R. Paul, editors, *Robotics Research 6*, pages 225–236. The Internat. Foundation for Robotics Research, 1994.
- [SA95] M. Sharir and P.K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [SD91] Z. Shiller and S. Dubowsky. On computing time-optimal motions of robotic manipulators in the presence of obstacles. *IEEE Trans. Robot. Autom.*, 7:785–797, 1991.
- [SEO03] M. Soss, J. Erickson, and M.H. Overmars. Preprocessing chains for fast dihedral rotations is hard or even impossible. *Comput. Geom. Theory Appl.*, 26:235–246, 2003.
- [SK01] A. Sudsang and L.E. Kavraki. A geometric approach to designing a programmable force field with a unique stable equilibrium for parts in the plane. In *Proc. IEEE Internat. Conf. Robot. Autom. (ICRA)*, pages 1079–1085, Seoul, 2001.
- [SL93] P. Souères and J.-P. Laumond. Shortest path synthesis for a car-like robot. In *Proc. 2nd European Control Conf.*, 1993.
- [SL01] T. Siméon and J.-P. Laumond. Notes on visibility roadmaps and path planning. In B.R. Donald, K.M. Lynch, and D. Rus, editors, *Algorithmic and Computational Robotics*, pages 317–328. A.K. Peters, Wellesley, 2001.
- [SL02] G. Sánchez and J.-C. Latombe. On delaying collision checking in prm planning: Application to multi-robot coordination. *Internat. J. Robot. Res.*, 21:5–26, 2002.
- [ŠO97] P. Švestka and M.H. Overmars. Motion planning for car-like robots, a probabilistic learning approach. *Internat. J. Robot. Res.*, 16:119–143, 1997.
- [SS94] J. Snoeyink and J. Stolfi. Objects that cannot be taken apart with two hands. *Discrete Comput. Geom.*, 12:367–384, 1994.
- [ST00] M. Soss and G.T. Toussaint. Geometric and computational aspects of polymer reconfiguration. *J. Math. Chemistry*, 27:303–318, 2000.
- [SY92] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM J. of Computing*, 21:863–888, 1992.

- [TA96] R. Talluri and J.K. Aggarwal. Mobile robot self-location using model-image feature correspondence. *IEEE Trans. Robot. Autom.*, 12:63–77, 1996.
- [Val98] P. Valtr. Guarding galleries where no point sees a small area. *Israel J. Mathematics*, 104:1–16, 1998.
- [Val99] P. Valtr. Guarding galleries with no bad points. *Discrete Comput. Geom.*, 21:193–200, 1999.
- [vSGO00] A.F. van der Stappen, K.Y. Goldberg, and M.H. Overmars. Geometric eccentricity and the complexity of manipulation plans. *Algorithmica*, 26:494–514, 2000.
- [vSWOO0] A.F. van der Stappen, C. Wentink, and M.H. Overmars. Computing immobilizing grasps of polygonal parts. *Internat. J. Robot. Res.*, 19:467–479, 2000.
- [WGPB96] J. Wiegley, K.Y. Goldberg, M. Peshkin, and M. Brokowski. A complete algorithm for designing passive fences to orient parts. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 1133–1139, 1996.
- [Wil91] G.T. Wilfong. Motion planning in the presence of movable objects. *Ann. Math. Artif. Intell.*, 3:131–150, 1991.
- [WKL⁺95] R.H. Wilson, L.E. Kavraki, J.-C. Latombe, and T. Lozano-Pérez. Two-handed assembly sequencing. *Internat. J. Robot. Res.*, 14:335–350, 1995.
- [WL95] R.H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artif. Intell.*, 71:371–396, 1995.
- [YB00] M. Yim and A. Berlin. Two approaches to distributed manipulation. In K.-F. Böhringer and H. Choset, editors, *Distributed Manipulation*, pages 237–261, Kluwer Academic, Boston, 2000.
- [ZF96] Z. Zhang and O. Faugeras. A 3d world model builder with a mobile robot. *Internat. J. Robot. Res.*, 11:269–285, 1996.
- [ZG95] Y. Zhuang and K.Y. Goldberg. On the existence of modular fixtures. *Internat. J. Robot. Res.*, 15(5), 1995.

49 COMPUTER GRAPHICS

David Dobkin and Seth Teller

INTRODUCTION

Computer graphics is often cited as a prime application area for the techniques of computational geometry. The histories of the two fields have a great deal of overlap, with similar methods (e.g., sweep-line and area subdivision algorithms) arising independently in each. Both fields have often focused on similar problems, although with different computational models. For example, hidden surface removal (visible surface identification) is a fundamental problem in both fields. At the same time, as the fields have matured, they have brought different requirements to similar problems. Here, we aim to highlight both similarities and differences between the fields.

Computational geometry is fundamentally concerned with the efficient quantitative representation and manipulation of ideal geometric entities to produce exact results. Computer graphics shares these goals, in part. However, graphics practitioners also model the interaction of objects with light and with each other, and the media through which these effects propagate. Moreover, graphics researchers and practitioners: (1) typically use finite precision (rather than exact) representations for geometry; (2) rarely formulate closed-form solutions to problems, instead employing sampling strategies and numerical methods; (3) often design into their algorithms explicit tradeoffs between running time and solution quality; (4) often analyze algorithm performance by defining as primitive operations those that have been implemented in hardware and (5) implement most algorithms they propose.

49.1 RELATIONSHIP TO COMPUTATIONAL GEOMETRY

In this section we elaborate these five contacts and contrasts.

GEOMETRY VS. RADIOMETRY AND PSYCHOPHYSICS

One fundamental computational process in graphics is *rendering*: the synthesis of realistic images of physical objects. This is done through the application of a simulation process to quantitative models of light, materials, and transmission media to predict (i.e., *synthesize*) appearance. Of course, this process must account for the shapes of and spatial relationships among objects and the viewer, as must computational geometric algorithms. In graphics, however, objects are imbued further with material properties, such as *reflectance* (in its simplest form, color), *refractive index*, *opacity*, and (for light sources) *emissivity*. Moreover, physically justifiable graphics algorithms must model *radiometry*: quantitative representations of light sources and the electromagnetic radiation they emit (with associated attributes of intensity, wavelength, polarization, phase, etc.), and the psychophysical aspects of the human visual system. Thus rendering is a kind of radiometrically

and psychophysically “weighted” counterpart to computational geometry problems involving interactions among objects.

CONTINUOUS IDEAL VS. DISCRETE REPRESENTATIONS

Computational geometry is largely concerned with ideal objects (points, lines, circles, spheres, hyperplanes, polyhedra), continuous representations (effectively infinite precision arithmetic), and exact combinatorial and algebraic results. Graphics algorithms (and their implementations) model such objects as well, but do so in a discrete, finite-precision computational model. For example, most graphics algorithms use a floating-point or fixed-point coordinate representation. Thus, one can think of many computer graphics computations as occurring on a (2D or 3D) sample grid. However, a practical difficulty is that the grid spacing is not constant, causing certain geometric predicates (e.g., sidedness) to change under simple transformations such as scaling or translation (see [Chapter 41](#)).

An analogy can be made between this distinct choice of coordinates, and the way in which geometric objects—*infinite collections of points*—are represented by geometers and graphics researchers. Both might represent a sphere similarly—say, by a center and radius. However, an algorithm to render the sphere must select a finite set of sample points on its surface. These sample points typically arise from the placement of a synthetic camera and from the locations of display elements on a two-dimensional display device, for example pixels on a computer monitor or ink dots on a page in a computer printer. The colors computed at these (zero-area) sample points, through some radiometric computation, then serve as an assignment to the discrete value of each (finite-area) display element.

CLOSED-FORM VS. NUMERICAL SOLUTION METHODS

Rarely does a problem in graphics demand a closed-form solution. Instead, graphists typically rely on numerical algorithms to estimate solution values in an iterative fashion. Numerical algorithms are chosen by reason of efficiency, or of simplicity. Often, these are antagonistic goals. Aside from the usual dangers of quantization into finite-precision arithmetic (Chapter 41), other types of error may arise from numerical algorithms. First, using a point-sampled value to represent a finite-area function’s value leads to discretization errors—differences between the reconstructed (interpolated) function, which may be piecewise-constant, piecewise-linear, piecewise-polynomial, etc., and the piecewise-continuous (but unknown) true function. These errors may be exacerbated by a poor choice of sampling points, by a poor piecewise function representation or basis, or by neglect of boundaries along which the true function or its derivative have strong discontinuities. Also, numerical algorithms may suffer bias and converge to incorrect solutions (e.g., due to the misweighting, or omission, of significant contributions).

TRADING SOLUTION QUALITY FOR COMPUTATION TIME

Many graphics algorithms recognize sources of error and seek to bound them by various means. Moreover, for efficiency’s sake an algorithm might deliberately introduce error. For example, during rendering, objects might be crudely approximated to

speed the geometric computations involved. Alternatively, in a more general illumination computation, many instances of combinatorial interactions (e.g., reflections) between scene elements might be ignored except when they have a significant effect on the computed image or radiometric values. Graphics practitioners have long sought to exploit this intuitive tradeoff between solution quality and computation time.

THEORY VS. PRACTICE

Graphics algorithms, while often designed with theoretical concerns in mind, are typically intended to be of practical use. Thus, while computational geometers and computer graphicists have a substantial overlap of interest in geometry, graphicists develop computational strategies that can feasibly be implemented on modern machines. Also, while computational geometric algorithms often assume “generic” inputs, in practice geometric degeneracies do occur, and inputs to graphics algorithms are at times highly degenerate (for example, comprised entirely of isothetic rectangles).

Thus, algorithmic strategies are shaped not only by challenging inputs that arise in practice, but also by the technologies available at the time the algorithm is proposed. The relative bandwidths of CPU, bus, memory, network connections, and tertiary storage have major implications for graphics algorithms involving interaction or large amounts of simulation data, or both. For example, in the 1980s the decreasing cost of memory, and the need for robust processing of general datasets, brought about a fundamental shift in most practitioners’ choice of computational techniques for resolving visibility (from combinatorial, object-space algorithms to brute force, screen-space algorithms). The increasing power of general-purpose processors, the emergence of sophisticated, robust visibility algorithms, and the wide availability of dedicated, programmable low-level graphics hardware may bring about yet another fundamental shift.

TOWARD A MORE FRUITFUL OVERLAP

Given such substantial overlap, there is potential for fruitful collaboration between geometers and graphicists [CAA⁺96]. One mechanism for spurring such collaboration is the careful posing of models and open problems to both communities. To that end, these are interspersed throughout the remainder of this chapter.

49.2 GRAPHICS AS A COMPUTATIONAL PROCESS

This section gives an overview of three fundamental graphics operations: *acquisition* of some *representation* of model data, its associated attributes and illumination sources; *rendering*, or simulating the appearance of static scenes; and simulating the *behavior* of dynamic scenes either in isolation or under the influence of user *interaction*.

GLOSSARY

Rendering problem: Given quantitative descriptions of surfaces and their properties, light sources, and the media in which all these are embedded, rendering is the application of a computational model to predict appearance; that is, rendering is the synthesis of images from simulation data. Rendering typically involves for each surface a *visibility* computation followed by a *shading* computation.

Visibility: Determining which pairs of a set of objects in a scene share an unobstructed line of sight.

Shading: The determination of radiometric values on a surface (eventually interpreted as colors) as viewed by the observer.

Simulation: The representation of a natural process by a computation.

Psychophysics: The study of the human visual system's response to electromagnetic stimuli.

REPRESENTATION: GEOMETRY, LIGHT, AND FORCES

Every computational process requires some representation in a form amenable to simulation. In graphics, the quantities to be represented span shape, appearance, and illumination. In simulation or interactive settings forces must also be represented; these may arise from the environment, from interactions among objects, or from the user's actions.

The graphics practitioner's choice of representation has significant implications. For example, how is the data comprising the representation to be acquired? For efficient manipulation or increased spatial or temporal coherence, the representation might have to include, or be amenable to, spatial indexing. A number of intrinsic (winged-edge, quad-edge, facet-edge, etc.) and extrinsic (quadtree, octree, k -d tree, BSP tree, B-rep, CSG, etc.) data structures have been developed to represent geometric data. Continuous, implicit functions have been used to model shape, as have discretized volumetric representations, in which data types or densities are associated with spatial "voxels." A subfield of modeling, Solid Modeling ([Chapter 56](#)), represents shape, mass, material, and connectivity properties of objects, so that, for example, complex object assemblies may be defined for use in Computer-Aided Machining environments ([Chapter 55](#)). Some of these data structures can be adaptively subdivided, and made persistent (that is, made to exist in memory and in nonvolatile storage; see [Chapter 34](#)), so that models with wide-scale variations, or simply enormous data size, may be handled. None of these data structures is universal; each has been brought to bear in specific circumstances, depending on the nature of the data (manifold vs. nonmanifold, polyhedral vs. curved, etc.) and the problem at hand. We forego here a detailed discussion of representational issues; see [Chapters 53](#) and [56](#).

The data structures alluded to above represent "macroscopic" properties of scene geometry—shape, gross structure, etc. Representing material properties, including reflectance over each surface, and possibly surface microstructure (such as roughness) and substructure (as with layers of skin or other tissue), is another fundamental concern of graphics. For each material, computer graphics researchers craft and employ quantitative descriptions of the interaction of radiant energy

and/or physical forces with objects having these properties. Examples include human-made objects such as machine parts, furniture, and buildings; organic objects such as flora and fauna; naturally occurring objects such as molecules, terrains, and galaxies; and wholly synthetic objects and materials. Analogously, suitable representations of radiant energy and physical forces also must be crafted in order that the simulation process can model such effects as erosion [DPH96].

ACQUISITION

In practice, algorithms require input. Realistic scene generation can demand complex geometric and radiometric models—for example, of scene geometry and reflectance properties, respectively. Nongeometric scene generation methods can use sparse or dense collections of images of real scenes. Geometric and image inputs must arise from some source; this *model acquisition* problem is a core problem in graphics. Models may be generated by a human designer (for example, using Computer-Aided Design packages), generated procedurally (for example, by applying recursive rules), or constructed by machine-aided manipulation of image data (for example, generating 3D topographical maps of terrestrial or extra-terrestrial terrain from multiple photographs), or other machine-sensing methods (e.g., [CL96]). Methods for completely automatic (i.e., not human-assisted) acquisition of large-scale geometric models are still in their infancy.

RENDERING

We partition the simulation process of *rendering* into *visibility* and *shading* sub-components, which are treated in separate subsections below.

For static scenes, and with more difficulty when conditions change with time, rendering can be factored into geometrically and radiometrically view-independent tasks (such as spatial partitioning for surface intervisibility, and the computation of diffuse illumination) and their view-dependent counterparts (culling and specular illumination, respectively). View-independent tasks can be cast as precomputations, while at least some view-dependent tasks cannot occur until the instantaneous viewpoint is known. These distinctions have been blurred by the development of data structures that organize lazily-computed, view-dependent information for use in interactive settings [TBD96].

INTERACTION (SIMULATION OF DYNAMICS)

Graphics brings to bear a wide variety of simulation processes to predict behavior. For example, one might detect collisions to simulate a pair of tumbling dice, or simulate frictional forces in order to provide haptic (touch) feedback through a mechanical device to a researcher manipulating a virtual object [LMC94]. Increasingly, graphics researchers are incorporating spatialized sounds into simulations as well. These physically-based simulations are integral to many graphics applications. However, the generation of synthetic imagery is the most fundamental operation in graphics. The next section describes this “rendering” problem.

When datasets become extremely large, some kind of hierarchical, persistent spatial database is required for efficient storage and access to the data [FKST96],

and simplification algorithms are necessary to store and display complex objects with varying fidelity (see, e.g., [CVM⁺96, HDD⁺92]).

We first discuss algorithmic aspects of model acquisition, a fundamental first step in graphics (Section 49.3). We next introduce rendering, with its intertwined operations of visibility determination, shading, and sampling (Section 49.4). We then pose several challenges for the future, listing problems of current or future interest in computer graphics on which computational geometry may have a substantial impact (Section 49.5). Finally, we list further references (Section 49.6).

49.3 ACQUISITION

Model acquisition is fundamental in achieving realistic, complex simulations. In some cases, the required model information may be “authored” manually, for example by a human user operating a computer-aided design application. Clearly manual authoring can produce only a limited amount of data. For more complex inputs, simulation designers have crafted “procedural” models, in which code is written to generate model geometry and attributes. However, such models often have limited expressiveness. To achieve both complexity and expressiveness, practitioners employ sensors such as cameras and range scanners to “capture” representations of real-world objects.

GLOSSARY

Model capture: Acquiring a data representation of a real-world object’s shape, appearance, or other properties.

GEOMETRY CAPTURE

In crafting a geometry capture method, the graphics practitioner must choose a sensor, for example a (passive) camera or multi-baseline stereo camera configuration, or an (active) laser range-finder. Regardless of sensor choice, data fusion from several sensors requires intrinsic and extrinsic sensor *calibration* and *registration* of multiple sensor observations. The fundamental algorithmic challenges here include handling noisy data, and solving the *data association* problem, i.e., determining which features match or correspond across sensor observations. When the device output (e.g., a point cloud) is not immediately useful as a geometric model, an intermediate step is required to infer geometric structure from the unorganized input [HDD⁺92, AB99]. These problems are particularly challenging in an interactive context, for example when merging range scans acquired at video rate [RHHL02]. In some applications, the datasize becomes enormous, as in the “Digital Michelangelo” project [LPC⁺00] or in GIS (geographical information systems) applied over large land areas (see Chapter 59).

One thrust common to both computer graphics and computer vision includes attempts to recover 3D geometry from many cameras situated outside or within the object or scene of interest. These “volumetric stereo” algorithms must face representational issues: a voxel data structures grows in size as the cube of the scene’s linear dimension, whereas a boundary representation is more efficient but

requires additional a priori information.

Another class of challenges arises from hybrids of procedural and data-driven methods. For example, there exist powerful “grammars” that produce complex synthetic flora using recursive elaboration of simple shapes [MP96]. These methods have a high “amplification factor” in the sense that they can produce complex geometry from a small number of parameters. However, they are notoriously difficult to invert; that is, given a set of observations of a tree, it is apparently difficult to recover an L-system (a particular string rewriting system) that reproduces the tree.

APPEARANCE CAPTURE

Another aspect of capture arises in the process of acquiring texture properties or other “appearance” attributes of geometric models. A number of powerful procedural methods exist for texture generation [Per85] and 3D volumetric effects such as smoke, fire, and clouds [SF95]. Researchers are challenged to make these methods data-driven, i.e., to find the procedural parameters that reproduce observations.

Recently, appearance capture approaches have emerged that attempt to avoid explicit geometry capture. These “image-based” modeling techniques [MB95] gather typically dense collections of images of the object or scene of interest, then use the acquired data to reconstruct images from novel viewpoints (i.e., viewpoints not occupied by the camera). Outstanding challenges for developers of these methods include: crafting effective sampling and reconstruction strategies; achieving effective storage and compression of the input images, which are often highly redundant; and achieving classical graphics effects such as re-illumination under novel lighting conditions when the underlying object geometry is unknown or only approximately known. Acquisition strategies are also needed when capturing materials with complex appearance due to, for example, subsurface effects (e.g., veined marble) [LPC⁺00].

MOTION CAPTURE

Capturing geometry and appearance of static scenes populated by rigid bodies is challenging. Yet this problem can itself be generalized in two ways. First, scenes may be dynamic, i.e., dependent on the passage of time. Second, scene objects may be articulated, i.e., composed of a number of rigid or deformable subobjects, linked through a series of geometric transformations. Although the dimensionality of the observed data may be immense, the actual number of degrees of freedom can be significantly lower; the computational challenge lies in discovering and representing the reduced dimensions efficiently and without an unacceptable loss of fidelity to the original motion. Thus motion capture yields a host of problems: segmenting objects from one another and from outlier data; inference of object substructure and degrees of freedom; and scaling up to complex articulated assemblies. Some of these problems have been addressed in Computer Vision (see also [Chapter 51](#)), although in graphics the same problems arise when processing 3D range (in contrast to 2D image) data.

OPEN PROBLEMS

- Given time-dependent range or motion data of several moving human figures, segment the figures from one another and produce as output an articulated model of each figure.
-

49.4 RENDERING

Rendering is the process through which a computer image of a model (acquired or otherwise) is created. To render an image that is perceived by the human visual system as being accurate is often considered to be the fundamental problem of computer graphics (*photorealistic rendering*). To do so requires visibility computations to determine which portions of objects are not obscured. Also required are shading computations to model the photometry of the situation. Because the resultant image will be sampled on a discrete grid, we must also consider techniques for minimizing sampling artifacts from the resultant image.

GLOSSARY

Visibility computation: The determination of whether some set of surfaces, or sample points, is visible to a synthetic observer.

Shading computation: The determination of radiometric values on the surface (eventually interpreted as colors) as viewed by the observer.

Pixel: A picture element, for example on a raster display.

Viewport: A 2D array of pixels, typically comprising a rectangular region on a computer display.

View frustum: A truncated rectangular pyramid, representing the synthetic observer's field of view, with the synthetic eyepoint at the apex of the pyramid. The truncation is typically accomplished using *near* and *far* clipping planes, analogous to the "left, right, top, and bottom" planes that define the rectangular field of view. (If the synthetic eyepoint is placed at infinity, the frustum becomes a rectangular parallelepiped.) Only those portions of the scene geometry that fall inside the view frustum are rendered.

Rasterization: The transformation of a continuous scene description, through discretization and sampling, into a discrete set of pixels on a display device.

Ray casting: A hidden-surface algorithm in which, for each pixel of an image, a ray is cast from the synthetic eyepoint through the center of the pixel [App68]. The ray is parametrized by a variable t such that $t = 0$ is the eyepoint, and $t > 0$ indexes points along the ray increasingly distant from the eye. The first intersection found with a surface in the scene (i.e., the intersection with minimum positive t) locates the visible surface along the ray. The corresponding pixel is assigned the intrinsic color of the surface, or some computed value.

Depth-buffering: (also *z-buffering*) An algorithm that resolves visibility by storing a discrete depth (initialized to some large value) at each pixel [Cat74]. Only when a rendered surface fragment's depth is less than that stored at the pixel can the fragment's color replace that currently stored at the pixel.

Irradiance: Total power per unit area impinging on a surface element. Units: POWER PER RECEIVER AREA.

BRDF: The Bidirectional Reflectance Distribution Function, which maps incident radiation (at general position and angle of incidence) to reflected exiting radiation (at general position and angle of exiting). Unitless, in [0, 1].

BTDF: The Bidirectional Transmission Distribution Function, which maps incident radiation (at general position and angle of incidence) to transmitted exiting radiation (at general position and angle of exiting). Analogous to the BRDF.

Radiance: The fundamental quantity in image synthesis, which is conserved along a ray traveling through a nondispersing medium, and is therefore “the quantity that should be associated with a ray in ray tracing” [CW93]. Units: POWER PER SOURCE AREA PER RECEIVER STERADIAN.

Radiosity: A global illumination algorithm for ideal diffuse environments. Radiosity algorithms compute shading estimates that depend only on the surface normal and the size and position of all other surfaces and light sources, and that are independent of view direction. Also: a physical quantity, with units POWER PER SOURCE AREA.

Ray tracing: An image synthesis algorithm in which ray casting is followed, at each surface, by a recursive shading operation involving a spherical/hemispherical integral of irradiance at each surface point. Ray tracing algorithms are best suited to scenes with small light sources and specular surfaces.

Hybrid algorithm: A global illumination algorithm that models both diffuse and specular interactions (e.g., [SP89]).

VISIBILITY

LOCAL VISIBILITY COMPUTATIONS

Given a scene composed of modeling primitives (e.g., polygons, or spheres), and a viewing frustum defining an eyepoint, a view direction, and field of view, the visibility operation determines which scene points or fragments are *visible*—connected to the eyepoint by a line segment that meets the closure of no other primitive. The visibility computation is global in nature, in the sense that the determination of visibility along a single ray may involve all primitives in the scene. Typically, however, visibility computations can be organized to involve coherent subsets of the model geometry.

In practice, algorithms for visible surface identification operate under severe constraints. First, available memory may be limited. Second, the computation time allowed may be a fraction of a second—short enough to achieve interactive refresh rates under changes in viewing parameters (for example, the location or viewing direction of the observer). Third, visibility algorithms must be simple

enough to be practical, but robust enough to apply to highly degenerate scenes that arise in practice.

The advent of machine rendering techniques brought about a cascade of screen-space and object-space combinatorial hidden-surface algorithms, famously surveyed and synthesized in [SSS74]. However, a memory-intensive screen-space technique—*depth-buffering@-buffering*—soon won out due to its simplicity and the decreasing cost of memory. In depth-buffering, specialized hardware performs visible surface determination independently at each pixel. Each polygon to be rendered is rasterized, producing a collection of pixel coordinates and an associated depth for each. A polygon fragment is allowed to “write” its color into a pixel only if the depth of the fragment at hand is less than the depth stored at the pixel (all pixel depths are initialized to some large value). Thus, in a complex scene each pixel might be written many times to produce the final image, wasting computation and memory bandwidth. This is known as the **overdraw** problem.

Two decades of spectacular improvement in graphics hardware have ensued, and high-end graphics workstations now contain hundreds of increasingly complex processors that clip, illuminate, rasterize, and texture millions of polygons per second. This capability increase has naturally led users to produce ever more complex geometric models, which suffer from increasing overdraw. Object simplification algorithms, which represent complex geometric assemblages with simpler shapes, do little to reduce overdraw. Thus, visible-surface identification (hidden-surface elimination) algorithms have again come to the fore (Section 28.8.1).

GLOBAL VISIBILITY COMPUTATIONS

Real-time systems perform visibility computations from an instantaneous synthetic viewpoint along rays associated with one or more samples at each pixel of some viewport. However, visibility computations also arise in the context of global illumination algorithms, which attempt to identify *all* significant light transport among point and area emitters and reflectors, in order to simulate realistic visual effects such as diffuse and specular interreflection and refraction. A class of *global* visibility algorithms has arisen for these problems. For example, in radiosity computations, a fundamental operation is determining **area-area** visibility in the presence of **blockers**; that is, the identification of those (area) surface elements visible to a given element, and for those partially visible, all tertiary elements causing (or potentially causing) occlusion [HW91, HSA91].

CONSERVATIVE ALGORITHMS

Graphics algorithms often employ *quadrature* techniques in their innermost loops—for example, estimating the energy arriving at one surface from another by casting multiple rays and determining an energy contribution along each. Thus, any efficiency gains in this frequent process (e.g., omission of energy sources known not to contribute energy at the receiver, or omission of objects known not to be blockers) will significantly improve overall system performance. Similarly, occlusion culling algorithms (omission of objects known not to contribute pixels to the rendered image) can significantly reduce overdraw. Both techniques are examples of **conservative** algorithms, which overestimate some geometric set by combinatorial means, then perform a final sampling-based operation that produces a (discrete) solution or quadrature. Of course, the success of conservative algorithms in practice depends on two assumptions: first, that through a relatively simple computation, a

usefully tight bound can be attained on whatever set would have been computed by a more sophisticated (e.g., exact) algorithm; and second, that the aggregate time of the conservative algorithm and the sampling pass is less than that of an exact algorithm for input sizes encountered in practice.

This idea can be illustrated as follows. Suppose the task is to render a scene of n polygons. If visible fragments must be rendered *exactly*, any correct algorithm must expend at least kn^2 time, since n polygons (e.g., two slightly misaligned combs, each with $n/2$ teeth) can cause $O(n^2)$ visible fragments to arise. But a conservative algorithm might simply render all n polygons, incurring some overdraw (to be resolved by a depth-buffer) at each pixel, but expending only time linear in the size of the input.

This highlights an important difference between computational geometry and computer graphics. Standard computational geometry cost measures would show that the $O(n^2)$ algorithm is optimal in an output-sensitive model (Section 28.8.1). In computer graphics, hardware considerations motivate a fundamentally different approach: rendering a (judiciously chosen) superset of those polygons that will contribute to the final image. A major open problem is to unify these approaches by finding a cost function that effectively models such considerations (see below).

HARDWARE TRENDS

In recent years, several hybrid object-space/screen-space visibility algorithms have emerged (e.g., [GKM93]). As general-purpose processors continue to become faster, such hybrid algorithms have become more widely used. In certain situations, these algorithms operate entirely in object space, without relying on special-purpose graphics hardware [CT96]. Specialized hardware for hierarchical visibility determination as envisioned in [GKM93], and programmable hardware capable of dedicated higher-level visibility operations such as ray-object intersection and spatial index traversal [PBMH02], will become increasingly available in the future, perhaps bringing about another shift in the algorithmic techniques of choice.

SHADING

Through sampling and visibility operations, a visible surface point or fragment is identified. This point or fragment is then *shaded* according to a *local* or *global* illumination algorithm. Given scene light sources and material reflection and transmission properties, and the propagative media comprising and surrounding the scene objects, the shading operation determines the color and intensity of the incident and exiting radiation at the point to be shaded. Shading computations can be characterized further as *view-independent* (modeling only purely diffuse interactions, or directional interactions with no dependence on the instantaneous eye position) or *view-dependent*.

Most graphics workstations perform a local shading operation in hardware, which, given a point light source, a surface point, and an eye position, evaluates the energy reaching the eye via a single reflection from the surface. This local operation is implemented in the software and hardware offered by most workstations. However, this simple model cannot produce realistic lighting cues such as shadows, reflection, and refraction. These require more extensive, global computations as described below.

SHADING AS RECURSIVE WEIGHTED INTEGRATION

Most generally, the shading operation computes the energy leaving a differential surface element in a specified differential direction. This energy depends on the surface's emittance and on the product of the surface's reflectance with the total energy incident from all other surfaces. This relation is known as the *Rendering Equation* [Kaj86], which states intuitively that each surface fragment's appearance, as viewed from a given direction, depends on any light it emits, plus any light (gathered from other objects in the scene) that it reflects in the direction of the observer. Thus, shading can be cast as a recursive integration; to shade a surface fragment F , shade all fragments visible to F , then sum those fragments' illumination upon F (appropriately weighted by the BRDF or BTDF) with any direct illumination of F . Effects such as diffuse illumination, motion blur, Fresnel effects, etc., can be simulated by supersampling in space, time, and wavelength, respectively, and then averaging [CPC84].

Of course, a base case for the recursion must be defined. Classical ray tracers truncate the integration when a certain recursion depth is reached. If this maximum depth is set to 1, ray casting (the determination of visibility for eye rays only) results. More common is to use a small constant greater than one, which leads to "Whitted" or "classical" ray tracing [Whi80]. For efficiency, practitioners also employ a thresholding technique: when multiple reflections cause the weight with which a particular contribution will contribute to the shading at the root to drop below a specified threshold, the recursion ceases. These termination conditions can, under some conditions, cause important energy-bearing paths to be overlooked. For example, a bright light source (such as the sun) filtering through many parts of a house to reach an interior space may be incorrectly discounted by this condition.

In recent years, a hardware trend has developed in support of "programmable shading," in which a (typically short, straight-line) program can be downloaded into graphics hardware for application to every vertex or pixel processed.¹ This trend has spurred research into, for example, ways to "factor" complex shading calculations into suitable components for mapping to hardware.

ALIASING

From a purely physical standpoint, the amount of energy leaving a surface in a particular direction is the product of the spherical integral of incoming energy and the bidirectional reflectance (and transmittance, as appropriate) in the exiting direction. From a psychophysical standpoint, the perceived color is an inner product of the energy distribution incident on the retina with the retina's spectral response function. We do not explore psychophysical considerations further here.

Global illumination algorithms perform an integration of irradiance at each point to be shaded. Ray tracing and radiosity are examples of global illumination algorithms. Since no closed-form solutions for global illumination are known for general scenes, practitioners employ sampling strategies. Graphics algorithms typically attempt "reconstruction" of some illumination function (e.g., irradiance, or radiance), given some set of samples of the function's values and possibly other information, for example about light source positions, etc. However, such reconstruction is subject to error for two reasons.

First, the well-known phenomenon of **aliasing** occurs when insufficient samples are taken to find all high-frequency terms in a sampled signal. In image processing,

¹Current manufacturers include NVIDIA <http://nvidia.com/> and ATI <http://ati.com/>.

samples arise from measurements, and reconstruction error arises from samples that are too widely spaced. However, in graphics, the sample values arise from a simulation process, for example, the evaluation of a local illumination equation, or the numerical integration of irradiance. Thus, reconstruction error can arise from simulation errors in generating the samples. This second type of error is called ***biasing***.

For example, classical ray tracers [Whi80] may suffer from biasing in three ways. First, at each shaded point, they compute irradiance only: from direct illumination by point lights; along the reflected direction; and along the refracted direction. Significant “indirect” illumination that occurs along any direction other than these is not accounted for. Thus, indirect reflection and focusing effects are missed. Classical ray tracers also suffer biasing by truncating the depth of the recursive ray tree at some finite depth d ; thus, they cannot find significant paths of energy from light source to eye of length greater than d . Third, classical ray tracers truncate ray trees when their weight falls below some threshold. This can fail to account for large radiance contributions due to bright sources illuminating surfaces of low reflectance.

SAMPLING

Sampling patterns can arise from a regular grid (e.g., pixels in a viewport) but these lead to a stair-stepping kind of aliasing. One solution is to ***supersample*** (i.e., take multiple samples per pixel) and average the results. However, one must take care to supersample in a way that does not align with the scene geometry or some underlying attribute (e.g., texture) in a periodic, spatially varying fashion; otherwise aliasing (including Moiré patterns) will result.

DISCREPANCY

The quality of sampling patterns can be evaluated with a measure known as ***discrepancy*** ([Chapter 44](#)). For example, if we are sampling in a pixel, features interacting with the pixel can be modeled by line segments (representing parts of edges of features) crossing the pixel. These segments divide the pixel into two regions. A good sampling strategy will ensure that the proportion of sample points in each region approximates the proportion of pixel area in that region. The difference between these quantities is the discrepancy of the point set with respect to the line segment. We define the discrepancy of a set of samples (in this case) as the maximum discrepancy with respect to all line segments. Other measures of discrepancy are possible, as described below. See also [Chapter 13](#).

Sampling patterns are used to solve integral equations. The advantage of using a low-discrepancy set is that the solution will be more accurately approximated, resulting in a better image. These differences are expressed in solution convergence rates as a function of the number of samples. For example, truly random sampling has a discrepancy that grows as $O(N^{-\frac{1}{2}})$ where N is the number of samples. There are other sampling patterns (e.g., the *Hammersley points*) that have discrepancies growing as $O(N^{-1} \log^{k-1} N)$. Sometimes one wishes to combine values obtained by different sampling methods [VG95]. The search for good sampling patterns, given a fixed number of samples, is often done by running an optimization process which aims to find sets of ever-decreasing discrepancy. A crucial part of any such process is the ability to quickly compute the discrepancy of a set of samples.

COMPUTING THE DISCREPANCY

There are two common questions that arise in the study of discrepancy: first, given fixed N , how to construct a good sampling pattern in the model described above; second, how to construct a good sampling pattern in an alternative model.

For concreteness, consider the problem of finding low discrepancy patterns in the unit square, modeling an individual pixel. As stated above, the geometry of objects is modeled by edges that intersect the pixel dividing it into two regions, one where the object exists and one where it does not. An ideal sampling method would sample the regions in proportion to their relative areas.

We model this as a discrepancy problem as follows. Let S be a sample set of points in the unit square. For a line l (actually, a segment arising from a polygon boundary in the scene being rendered), define the two regions S^+ and S^- into which l divides S . Ideally, we want a sampling pattern that has the same fraction of samples in the region S^+ as the area of S^+ . Thus, in the region S^+ , the discrepancy with respect to l is

$$|\#\left(S \cap S^+\right) / \#\left(S\right) - \text{Area}(S^+)|,$$

where $\#(\cdot)$ denotes the cardinality operator. The discrepancy of the sample set S with respect to a line l is defined as the larger of the discrepancies in the two regions. The discrepancy of set S is then the maximum, over *all* lines l , of the discrepancy of S with respect to l .

Finding the discrepancy in this setting is an interesting computational geometry problem. First, we observe that we do not need to consider all lines. Rather, we need consider only those lines that pass through two points of S , plus a few lines derived from boundary conditions. This suggests the $O(n^3)$ algorithm of computing the discrepancy of each of the $O(n^2)$ lines separately. This can be improved to $O(n^2 \log n)$ by considering the fan of lines with a common vertex (i.e., one of the sample points) together. This can be further improved by appealing to duality. The traversal of this fan of lines is merely a walk in the arrangement of lines in dual space that are the duals of the sample points. This observation allows us to use techniques similar to those in [Chapter 24](#) to derive an algorithm that runs asymptotically as $O(n^2)$. Full details are given in [DEM93].

There are other discrepancy models that arise naturally. A second obvious candidate is to measure the discrepancy of sample sets in the unit square with respect to axis-oriented rectangles. Here we can achieve a discrepancy of $O(n^2 \log n)$, again using geometric methods. We use a combination of techniques, appealing to the incremental construction of 2D convex hulls to solve a basic problem, then using the sweep paradigm to extend this incrementally to a solution of the more general problem. The sweep is easier in the case in which the rectangle is anchored with one vertex at the origin, yielding an algorithm with running time $O(n \log^2 n)$.

The model given above can be generalized to compute **bichromatic discrepancy**. In this case, we have sample points that are colored either black or red. We can now define the discrepancy of a region as the difference between its number of red and black points. Alternatively, we can look for regions (of the allowable type) that are most nearly monochromatic in red while their complements are nearly monochromatic in black. This latter model has application in computational learning theory. For example, red points may represent situations in which a concept is true, black situations where it is false. The minimum discrepancy rectangle is now a classifier of the concept. This is a popular technique for computer-assisted medical diagnosis.

The relevance of these algorithms to computational geometry is that they will lead to faster algorithms for testing the “goodness” of sampling patterns, and thus eventually more efficient algorithms with bounded sampling error. Also, algorithms for computing the discrepancy relative to a particular set system are directly related to the system’s VC-dimension (see [Section 44.1](#)).

OPEN PROBLEMS

An enormous literature of adaptive, backward, forward, distribution, etc. ray tracers has evolved to address sampling and bias errors. However, the fundamental issues can be stated simply. (Each of the problems below assumes a geometric model consisting of n polygons.)

A related *inverse* problem arises in machine vision, now being adopted by computer graphics practitioners as a method for acquiring large-scale geometric models from imagery.

The problems below are open for both the unit cube and unit ball in all dimensions.

1. The set of visible fragments can have complexity $\Omega(n^2)$ in the worst case. However, the complexity is lower for many scenes. If k is the number of edge incidences (vertices) in the projected visible scene, the set of visible fragments can be computed in optimal output-sensitive $O(nk^{1/2} \log n)$ time [SO92]. Although specialized results have been obtained, optimality has not been reached in many cases. See [Table 28.8.1](#).
2. Give a spatial partitioning and ray casting algorithm that runs in amortized nearly-constant time (that is, has only a weak asymptotic dependence on total scene complexity). Identify a useful “density” parameter of the scene (e.g., the largest number of simultaneously visible polygons), and express the amortized cost of a ray cast in terms of this parameter.
3. Give an output-sensitive algorithm which, for specified viewing parameters, determines the set of “contributing” polygons—i.e., those which contribute their color to at least one viewport pixel.
4. Give an output-sensitive algorithm which, for specified viewing parameters, approximates the visible set to within ϵ . That is, produce a superset of the visible polygons of size (alternatively, total projected area) at most $(1 + \epsilon)$ times the size (resp., projected area) of the true set. Is the lower bound for this problem asymptotically smaller than that for the exact visibility problem?
5. For machine-dependent parameters A and B describing the transform (per-vertex) and fill (per-pixel) costs of some rendering architecture, give an algorithm to compute a superset S of the visible polygon set minimizing the rendering cost on the specified architecture.
6. In a local illumination computation, identify those polygons (or a superset) visible from the synthetic observer, and construct, for each visible polygon P , an efficient function $V(p)$ that returns 1 iff point $p \in P$ is visible from the viewpoint.

7. In a global illumination computation, identify all pairs (or a superset) of intervisible polygons, and for each such pair P, Q , construct an efficient function $V(p, q)$ that returns 1 iff point $p \in P$ is visible from point $q \in Q$.
 8. **Image-based rendering** [MB95]: Given a 3D model, generate a minimal set of images of the model such that for all subsequent query viewpoints, the correct image can be recovered by combination of the sample images.
 9. Given a geometric model M , a collection of light sources L , a synthetic viewpoint E , and a threshold ϵ , identify all optical paths to E bearing radiance greater than ϵ .
 10. Given a geometric model M , a collection of light sources L , and a threshold ϵ , identify *all* optical paths bearing radiance greater than ϵ .
 11. An observation of a real object comprises the *product* of irradiance and reflection (BRDF). How can one deduce the BRDF from such observations?
 12. Given N , generate a minimum-discrepancy pattern of N samples.
 13. Given a low-discrepancy pattern of K points, generate a low (or lower) discrepancy pattern of $K + 1$ points.
-
-

49.5 FURTHER CHALLENGES

We have described several core problems of computer graphics and illustrated the impact of computational geometry. We have only scratched the surface of a highly fruitful interaction; the possibilities are expanding, as we describe below. These computer graphics problems all build on the combinatorial framework of computational geometry and so have been, and continue to be, ripe candidates for application of computational geometry techniques. Numerous other problems remain whose combinatorial aspects are perhaps less obvious, but for which interaction may be equally fruitful.

INDEX AND SEARCH

The proliferation of geometric models leads to a problem analogous to that in document storage: how to index models so that they can be efficiently found later. In particular, we might wish to define the Google of 3D models. Searching by name is of limited utility, since in many cases a model's author may not have named it suggestively, or as expected by the seeker. Searching by attributes or appearance is likely to be more fruitful or at the least, a necessary adjunct to searching by name. Perhaps the most successful search mechanisms to date are those relying on geometric “shape signatures” of objects, along with name and attribute metadata where available [FMK⁺03]. One promising class of signatures related to the medial axis transform is the “shock graph” [LK01]. A first step toward building such a system appears in [OFCD02].

TRANSMISSION AND LEVEL OF DETAIL

Fast network connectivity is not yet universally deployed, and the number and size of available models is growing inexorably with time. Thus in many contexts it is important to store, transmit, and display geometric information efficiently. A variety of techniques have been developed for “progressive” [Hop97] or “multi-resolution” geometry representation [GSS99], as well as for automated level-of-detail generation from source objects [GH97]. For specific model classes, e.g., terrain, efficient algorithms have been developed for varying the fidelity of the display across the field of view [dBD98]. Finally, some practitioners have proposed techniques to choose levels of detail, within some time rendering budget, to optimize some image quality criterion [FKST96].

OPEN PROBLEM

- *Robust simplification.* Cheng et al. [SWCP02] recently gave a method for computing levels of details that preserve the genus of the original surface. Combine their techniques with techniques for robust computation to derive a robust and efficient scheme for simplification that can be easily implemented. See [Chapter 54](#).

INTERACTION

In addition to off-line or batch computations, graphics practitioners develop on-line computations which involve a user in an interactive loop with input and output (display) stages, such as scientific visualization. For responsiveness, such applications may have to produce many outputs per second: rendering applications typically must maintain 10Hz or faster, whereas haptic or force-feedback applications may operate at 1KHz. Modern applications must also cope with large datasets, only parts of which may be memory-resident at any moment. Thus effective techniques for indexing, searching, and transmitting model data are required. For out-of-core data, predictive fetching strategies are required to avoid high-latency “hiccups” in the user’s display.

Beyond seeing and feeling virtual representations of an object, new “3D printing” techniques have emerged for rapid prototyping applications that create real, physical models of objects. Computational geometry algorithms are required to plan the slicing or deposition steps needed. Also, “augmented reality” (AR) methods attempt to provide synthetically generated image overlays onto real scenes, for example using head-mounted displays or hand-held projectors. AR methods require good, low-latency 6-DOF tracking of the user’s head or device position and orientation in extended environments.

An exciting new class of “pervasive computing” and “mobile computing” applications attempts to move computation away from the desktop and out into the extended work, home, or outdoor environment. These applications are by nature integrative, encompassing geometric and functional models, position and orientation tracking, proximity data structures, ad hoc networks, and distributed self-calibration algorithms [PMBT01].

OPEN PROBLEM

- *Collision detection and force feedback.* Imagine that every object has an associated motion, and that some objects (e.g., virtual probes) are interactively controlled. Suppose further that when pairs of objects intersect, there is a reaction (due, e.g., to conservation of momentum). Here we wish to render frames and generate haptic feedback while accounting for such physical considerations. Are there suitable data structures and algorithms within computational geometry to model and solve this problem (e.g., [LMC94, MC95])?
-

DYNAMICS

When simulations include objects that affect each other through force exchange or collision, they must efficiently identify the actual interactions. Usually there is significant temporal coherence, i.e., the set of objects near a given object changes slowly over time. A number of techniques have been proposed to track moving objects in a spatial index or closest-pair geometric data structure in order to detect collisions efficiently [MC95, LMC94, BGH99]. The “object” of interest may be the geometric representation of a user, for example of a finger or hand probing a virtual scene. Recently, some authors have proposed synthesizing sound information to accompany the visual simulation outputs [OSG02].

We have focused this chapter on problems in which the parameters are static; that is, the geometry is unchanging, and nothing is moving (except perhaps the synthetic viewpoint). Now, we briefly describe situations where this is not the case and deeper analysis is required. In these situations it is likely that computational geometry can have a tremendous impact; we sketch some possibilities here.

Each of the static assumptions above may be relaxed, either alone or in combination. For example, objects may evolve with time; we may be interested in transient rather than steady-state solutions; material properties may change over time; object motions may have to be computed and resolved; etc. It is a challenge to determine how techniques of computational geometry can be modified to address state-of-the-art and future computer graphics tasks in dynamic environments.

Among the issues we have not addressed where these considerations are important are the following.

Model changes over time. In a realistic model, even unmoving objects change over time, for example becoming dirty or scratched. In some environments, objects rust or suffer other corrosive effects. Sophisticated geometric representations and algorithms are necessary to capture and model such phenomena [DPH96]. See [Chapter 50](#).

Inverse processes. Much of what we have described is a feed-forward process in which one specifies a model and a simulation process and computes a result. Of equal importance in design contexts is to specify a result and a simulation process, and compute a set of initial conditions that would produce the desired result. For example, one might wish to specify the appearance of a stage, and deduce the intensities of scores or hundreds of illuminating light sources that would result in this appearance [SDSA93]. Or, one might wish to solve an inverse kinematics problem in which an object with multiple parts and numerous degrees of freedom is specified.

Given initial and final states, one must compute a smooth, minimal energy path between the states, typically in an underconstrained framework. This is a common problem in robotics (see [Section 47.1](#)). However, the configurations encountered in graphics tend to have very high complexity. For example, convincingly simulating the motion of a human figure requires processing kinematic models with hundreds of degrees of freedom.

External memory algorithms. Computational geometry assumes a realm in which all data can be stored in RAM and accessed at no cost (or unit cost per word). Increasingly often, this is not the case in practice. For example, many large databases cannot be stored in main memory. Only a small subset of the model contributes to each generated image, and algorithms for efficiently identifying this subset, and maintaining it under small changes of the viewpoint or model, form an active research area in computer graphics. Given that motion in virtual environments is usually smooth, and that hard real-time constraints preclude the use of purely reactive, synchronous techniques, such algorithms must be *predictive* and *asynchronous* in nature [FKST96]. Achieving efficient algorithms for appropriately shuttling data between secondary (and tertiary) storage and main memory is an interesting challenge for computational geometry.

49.6 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys:

- [Dob92]: A survey article on computational geometry and computer graphics.
- [Dor94]: Survey of object-space hidden-surface removal algorithms.
- [Yao92, LP84]: Surveys of computational geometry.
- [CCSD03]: Survey of visibility for walkthroughs.

RELATED CHAPTERS

- [Chapter 13: Geometric discrepancy theory and uniform distribution](#)
- [Chapter 25: Triangulations and mesh generation](#)
- [Chapter 26: Polygons](#)
- [Chapter 28: Visibility](#)
- [Chapter 35: Collision detection](#)
- [Chapter 37: Ray shooting and lines in space](#)
- [Chapter 50: Algorithms for tracking moving objects](#)
- [Chapter 53: Splines and geometric modeling](#)
- [Chapter 54: Surface simplification and 3D geometry compression](#)
- [Chapter 56: Solid modeling](#)

REFERENCES

- [AB99] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22:481–504, 1999.
- [App68] A. Appel. Some techniques for shading machine renderings of solids. In *Proc. SJCC*, pages 37–45. Thompson Books, Washington, 1968.
- [BGH99] J. Basch, L.J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Algorithms*, 31:1–28, 1999.
- [Cat74] E.E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. Ph.D. thesis, Univ. Utah, TR UTEC-CSc-74-133, 1974.
- [CAA⁺96] B. Chazelle, N. Amenta, Te. Asano, G. Barequet, M. Bern, J.-D. Boissonnat, J.F. Canny, K.L. Clarkson, D.P. Dobkin, B.R. Donald, S. Drysdale, H. Edelsbrunner, D. Eppstein, A.R. Forrest, S.J. Fortune, K.Y. Goldberg, M.T. Goodrich, L.J. Guibas, P. Hanrahan, C. Hoffmann, D. Huttenlocher, H. Imai, D.G. Kirkpatrick, D.T. Lee, K. Mehlhorn, V.J. Milenkovic, J.S.B. Mitchell, M.H. Overmars, R. Pollack, R. Seidel, M. Sharir, J. Snoeyink, G.T. Toussaint, S. Teller, H. Voelcker, E. Welzl, and C.K. Yap. Application Challenges to Computational Geometry: CG Impact Task Force Report. Tech. Rep. TR-521-96, Princeton CS Dept., 1996. <http://graphics.lcs.mit.edu/~seth/pubs/taskforce/techrep.html>
- [CVM⁺96] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P.K. Agarwal, F.P. Brooks, Jr., and W.V. Wright. Simplification envelopes. In *Proc. ACM Conf. SIGGRAPH 96*, pages 119–128, 1996.
- [CCSD03] D. Cohen-Or, Y. Chrysanthou, C. Silva, and F. Durand. A survey of visibility for walkthrough applications. *IEEE Trans. Visualization Comput. Graphics*, 9:412–431, 2003.
- [CW93] M.F. Cohen and J.R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, Cambridge, 1993.
- [CPC84] R.L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In *Proc. ACM Conf. SIGGRAPH 84*, volume 18, pages 137–45, 1984.
- [CT96] S. Coorg and S. Teller. Temporally coherent conservative visibility. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, 1996.
- [CL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. ACM Conf. SIGGRAPH 96*, pages 303–312, 1996.
- [dBd98] M. de Berg and K. Dobrindt. On levels of detail in terrains. *Graphical Models Image Proc.*, 60:1–12, 1998.
- [DEM93] D.P. Dobkin, D. Eppstein, and D. Mitchell. Computing the discrepancy with applications to supersampling patterns. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 47–52, 1993.
- [Dob92] D.P. Dobkin. Computational geometry and computer graphics. *Proc. IEEE*, 80:1400–1411, 1992.
- [DPH96] J. Dorsey, H. Pedersen, and P. Hanrahan. Flow and changes in appearance. In *Proc. ACM Conf. SIGGRAPH 96*, pages 411–420, 1996.
- [Dor94] S.E. Dorward. A survey of object-space hidden surface removal. *Internat. J. Comput. Geom. Appl.*, 4:325–362, 1994.

- [FMK⁺03] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D.P. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Trans. Graph.*, 22:83–105, 2003.
- [FKST96] T. Funkhouser, D. Khorramabadi, C. Séquin, and S. Teller. The UCB system for interactive visualization of large architectural models. *Presence*, 5:13–44, Winter 1996.
- [GH97] M. Garland and P.S. Heckbert. Surface simplification using quadric error metrics. In *Proc. ACM Conf. SIGGRAPH 97*, pages 209–216, 1997.
- [GKM93] N. Greene, M. Kass, and G.L. Miller. Hierarchical Z-buffer visibility. In *Proc. ACM Conf. SIGGRAPH 93*, pages 231–238, 1993.
- [GSS99] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proc. ACM Conf. SIGGRAPH 99*, pages 325–334, 1999.
- [HW91] E. Haines and J.R. Wallace. Shaft culling for efficient ray-traced radiosity. In *Proc. 2nd Eurographics Workshop Rendering*, pages 122–138, 1991.
- [HSA91] P. Hanrahan, D. Salzman, and L.J. Aupperle. A rapid hierarchical radiosity algorithm. In *Proc. ACM Conf. SIGGRAPH 91*, pages 197–206, 1991.
- [HDD⁺92] H. Hoppe, T.D. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. ACM Conf. SIGGRAPH 92*, pages 71–78, 1992.
- [Hop97] H. Hoppe. View-dependent refinement of progressive meshes. In *Proc. ACM Conf. SIGGRAPH 97*, pages 189–198, 1997.
- [HDD⁺92] H. Hoppe, T.D. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. ACM Conf. SIGGRAPH 92*, pages 71–78, 1992.
- [Kaj86] J.T. Kajiya. The rendering equation. In *Proc. ACM Conf. SIGGRAPH 86*, pages 143–150, 1986.
- [LP84] D.T. Lee and F.P. Preparata. Computational geometry: A survey. *IEEE Trans. Comput.*, 33:1072–1101, 1984.
- [LPC⁺00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *Proc. ACM Conf. SIGGRAPH 00*, pages 131–144, 2000.
- [LK01] F.F. Leymarie and B.B. Kimia. The shock scaffold for representing 3D shape. In *Proc. 4th Internat. Workshop Visual Form*, pages 216–229. Springer-Verlag, Berlin, 2001.
- [LMC94] M.C. Lin, D. Manocha, and J.F. Canny. Fast contact determination in dynamic environments. In *Proc. Internat. Conf. Robot. Autom.*, pages 602–609, 1994.
- [MB95] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proc. ACM Conf. SIGGRAPH 95*, pages 39–46, 1995.
- [MP96] R. Mech and P. Prusinkiewicz. Visual models of plants interacting with their environment. In *Proc. ACM Conf. SIGGRAPH 96*, pages 397–410, 1996.
- [MC95] B. Mirtich and J.F. Canny. Impulse-based simulation of rigid bodies. In *1995 Sympos. Interactive 3D Graphics*, pages 181–188, 1995.
- [OSG02] J. O’Brien, C. Shen, and C. Gatchalian. Natural phenomena: Synthesizing sounds from rigid-body simulations. *Proc. ACM SIGGRAPH Sympos. Computer Animation*, 2002.
- [OFCD02] R. Osada, T. Funkhouser, B. Chazelle, and D.P. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21:807–832, 2002.

- [Per85] K. Perlin. An image synthesizer. In *Proc. ACM Conf. SIGGRAPH 85*, pages 287–296, 1985.
- [PMBT01] N. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *Proc. 7th ACM Internat. Conf. Mobile Comput. Network*, pages 1–14, 2001.
- [PBMH02] T.J. Purcell, I. Buck, R.M. William, and P. Hanrahan. Ray tracing on programmable graphics hardware. *ACM Trans. Graph.*, 21:703–712, 2002.
- [RHHL02] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. In *Proc. ACM Conf. SIGGRAPH 02*, pages 438–446, 2002.
- [SWCP02] T.K. Dey, S.W. Cheng and S.H. Poon. Hierarchy of surface models and irreducible triangulation. In *Proc. Internat. Sympos. Algorithms Comput.*, pages 286–295, 2002.
- [SDSA93] C. Schoeneman, J. Dorsey, B. Smits, and J. Arvo. Painting with light. *Proc. ACM Conf. SIGGRAPH 93*, pages 143–146, 1993.
- [SO92] M. Sharir and M.H. Overmars. A simple output-sensitive algorithm for hidden surface removal. *ACM Trans. Graph.*, 11:1–11, 1992.
- [SP89] F.X. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *Proc. ACM Conf. SIGGRAPH 89*, pages 335–344, 1989.
- [SF95] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *Proc. ACM Conf. SIGGRAPH 95*, pages 129–136, 1995.
- [SSS74] I.E. Sutherland, R.F. Sproull, and R.A. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Comput. Surv.*, 6:1–55, 1974.
- [TBD96] S. Teller, K. Bala, and J. Dorsey. Conservative radiance envelopes for ray tracing. In *Proc. 7th Eurographics Workshop Rendering*, pages 105–114, 1996.
- [VG95] E. Veach and L.J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proc. ACM Conf. SIGGRAPH 95*, pages 419–428, 1995.
- [Whi80] T. Whitted. An improved illumination model for shading display. *Commun. ACM*, 23:343–349, 1980.
- [Yao92] F.F. Yao. Computational geometry. In *Algorithms and Complexity, Handbook of Theoretical Computer Science*, volume A, Elsevier Science, Amsterdam, pages 343–389, 1992.

50 MODELING MOTION

Leonidas J. Guibas

50.1 INTRODUCTION

Motion is ubiquitous in the physical world, yet its study is much less developed than that of another common physical modality, namely shape. While we have several standardized mathematical shape descriptions, and even entire disciplines devoted to that area—such as *Computer-Aided Geometric Design* (CAGD)—the state of formal motion descriptions is still in flux. This in part because motion descriptions span many levels of detail; they also tend to be intimately coupled to an underlying physical process generating the motion (dynamics). Thus, until recently, proper abstractions were lacking and there was only limited work on algorithmic descriptions of motion and their associated complexity measures. This chapter aims to show how an algorithmic study of motion is intimately tied to discrete and computational geometry. After a quick survey of earlier work (Sections 50.2 and 50.3), we devote the bulk of this chapter to discussing the framework of *Kinetic Data Structures* (Section 50.4) [Gui98, BGH99]. We also briefly discuss methods for querying moving objects (Section 50.5).

50.2 MOTION IN COMPUTATIONAL GEOMETRY

Dynamic computational geometry refers to the study of combinatorial changes in a geometric structure, as its defining objects undergo prescribed motions. For example, we may have n points moving linearly with constant velocities in \mathbb{R}^2 , and may want to know the time intervals during which a particular point appears on their convex hull, the steady-state form of the hull (after all changes have occurred), or get an upper bound on how many times the convex hull changes during this motion. Such problems were introduced and studied in [Ata85].

A number of other authors have dealt with geometric problems arising from motion, such as collision detection ([Chapter 35](#)) or minimum separation determination [GJS96, ST95, ST96]. For instance, [ST96] shows how to check in subquadratic time whether two collections of simple geometric objects (spheres, triangles) collide with each other under specified polynomial motions.

50.3 MOTION MODELS

An issue in the above research is that object motion(s) are assumed to be known in advance, sometimes in explicit form (e.g., points moving as polynomial functions

of time). Indeed, the proposed methods reduce questions about moving objects to other questions about derived static objects.

While most evolving physical systems follow known physical laws, it is also frequently the case that discrete events occur (such as collisions) that alter the motion law of one or more of the objects. Thus motion may be predictable in the short term, but becomes less so further into the future. Because of such discrete events, algorithms for modeling motion must be able to adapt in a dynamic way to motion model modifications. Furthermore, the occurrence of these events must be either predicted or detected, incurring further computational costs. Nevertheless, any truly useful model of motion must accommodate this *on-line* aspect of the temporal dimension, differentiating it from spatial dimensions, where all information is typically given at once.

In real-world settings, the motion of objects may be imperfectly known and better information may only be obtainable at considerable expense. The model of *data in motion* of [Kah91] assumes that upper bounds on the rates of change are known, and focuses on being selective in using sensing to obtain additional information about the objects, in order to answer a series of queries.

50.4 KINETIC DATA STRUCTURES

Suppose we are interested in tracking high-level attributes of a geometric system of objects in motion such as, for example, the convex hull of a set on n points moving in \mathbb{R}^2 . Note that as the points move continuously, their convex hull will be a continuously evolving convex polygon. At certain discrete moments, however, the combinatorial structure of the convex hull will change (that is, the circular sequence of a subset of the points that appear on the hull will change). In between such moments, tracking the hull is straightforward: its geometry is determined by the positions of the sequence of points forming the hull. How can we know when the combinatorial structure of the hull changes? The idea is that we can focus on certain elementary geometric relations among the n points, a set of *cached assertions*, which altogether certify the correctness of the current combinatorial structure of the hull. Furthermore, we can hope to choose these relations in such a way so that when one of them fails because of point motion, both the hull and its set of certifying relations can be updated locally and incrementally, so that the whole process can continue.

GLOSSARY

Kinetic data structure: A kinetic data structure (KDS) for a geometric attribute is a collection of simple geometric relations that certifies the combinatorial structure of the attribute, as well as a set of rules for repairing the attribute and its certifying relations when one relation fails.

Certificate: A certificate is one of the elementary geometric relations used in a KDS.

Event: An event is the failure of a KDS certificate during motion. Events are classified as *external* when the combinatorial structure of the attribute changes, and *internal*, when the structure of the attribute remains the same, but its

certification needs to change.

Event queue: In a KDS, all certificates are placed in an event queue, according to their earliest failure time.

The inner loop of a KDS consists of repeated certificate failures and certification repairs, as depicted in Figure 50.4.1.

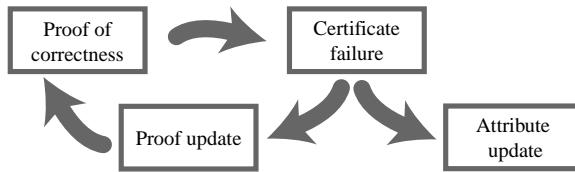


FIGURE 50.4.1
The inner loop of a kinetic data structure.

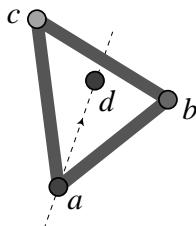
We remark that in the KDS framework, objects are allowed to change their motions at will, with appropriate notification to the data structure. When this happens all certificates involving the object whose motion has changed must re-evaluate their failure times.

CONVEX HULL EXAMPLE

Suppose we have four points a , b , c , and d in \mathbb{R}^2 , and wish to track their convex hull. For the convex hull problem, the most important geometric relation is the CCW predicate: $\text{CCW}(a, b, c)$ asserts that the triangle abc is oriented counterclockwise. Figure 50.4.2 shows a configuration of four points and four CCW relations that hold among them. It turns out that these four relations are sufficient to prove that the convex hull of the four points is the triangle abc . Indeed the points can move and form different configurations, but as long as the four certificates shown remain valid, the convex hull must be abc .

Now suppose that points a , b , and c are stationary and only point d is moving, as shown in Figure 50.4.3. At some time t_1 the certificate $\text{CCW}(d, b, c)$ will fail, and at a later time t_2 $\text{CCW}(d, a, b)$ will also fail. Note that the certificate $\text{CCW}(d, c, a)$ will never fail in the configuration shown even though d is moving. So the certificates $\text{CCW}(d, b, c)$ and $\text{CCW}(d, a, b)$ schedule events that go into the event queue. At time t_1 , $\text{CCW}(d, b, c)$ ceases to be true and its negation, $\text{CCW}(c, b, d)$, becomes true. In this simple case the three old certificates, plus the new certificate $\text{CCW}(c, b, d)$, allow us to conclude that the convex hull has now changed to $abdc$.

If the certificate set is chosen judiciously, the KDS repair can be a local, incremental process—a small number of certificates may leave the cache, a small number may be added, and the new attribute certification will be closely related to the old one. A good KDS exploits the continuity or coherence of motion and change in the world to maintain certifications that themselves change only incrementally and locally as assertions in the cache fail.



Proof of correctness:

- $\text{CCW}(a, b, c)$
- $\text{CCW}(d, b, c)$
- $\text{CCW}(d, c, a)$
- $\text{CCW}(d, a, b)$

FIGURE 50.4.2
Determining the convex hull of the points.

Old proof	New proof
$\text{CCW}(a, b, c)$	$\text{CCW}(a, b, c)$
$\text{CCW}(d, b, c)$	$\text{CCW}(c, b, d)$
$\text{CCW}(d, c, a)$	$\text{CCW}(d, c, a)$
$\text{CCW}(d, a, b)$	$\text{CCW}(d, a, b)$

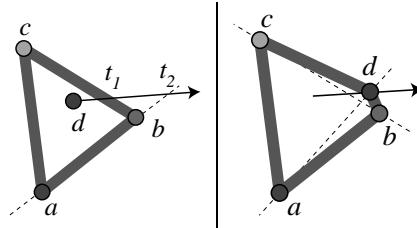


FIGURE 50.4.3
Updating the convex hull of the points.

PERFORMANCE MEASURES FOR KDS

Because a KDS is not intended to facilitate a terminating computation but rather an on-going process, we need to use somewhat different measures to assess its complexity. In classical data structures there is usually a tradeoff between operations that interrogate a set of data and operations that update the data. We commonly seek a compromise by building indices that make queries fast, but such that updates to the set of indexed data are not that costly as well. Similarly in the KDS setting, we must at the same time have access to information that facilitates or trivializes the computation of the attribute of interest, yet we want information that is relatively stable and not so costly to maintain. Thus, in the same way that classical data structures need to balance the efficiency of access to the data with the ease of its update, kinetic data structures must tread a delicate path between “knowing too little” and “knowing too much” about the world. A good KDS will select a certificate set that is at once economical and stable, but also allows a quick repair of itself and the attribute computation when one of its certificates fails.

GLOSSARY

responsiveness: A KDS is *responsive* if the cost, when a certificate fails, of repairing the certificate set and updating the attribute computation is small. By

“small” we mean polylogarithmic in the problem size—in general we consider small quantities that are polylogarithmic or $O(n^\epsilon)$ in the problem size.

efficiency: A KDS is *efficient* if the number of certificate failures (total number of events) it needs to process is comparable to the number of required changes in the combinatorial attribute description (external events), over some class of allowed motions. Technically, we require that the ratio of total events to external events is small. The class of allowed motions is usually specified as the class of *pseudo-algebraic* motions, in which each KDS certificate can flip between true and false at most a bounded number of times.

compactness: A KDS is *compact* if the size of the certificate set it needs is close to linear in the degrees of freedom of the moving system.

locality: A KDS is *local* if no object participates in too many certificates; this condition makes it easier to re-estimate certificate failure times when an object changes its motion law. (The existence of local KDSs is an intriguing theoretical question for several geometric attribute functions.)

CONVEX HULL, REVISITED

We now briefly describe a KDS for maintaining the convex hull of n points moving around in the plane [BGH99].

The key goal in designing a KDS is to produce a *repairable certification* of the geometric object we want to track. In the convex hull case it turns out that it is a bit more intuitive to look at the dual problem, that of maintaining the upper (and lower) envelope of a set of moving lines in the plane, instead of the convex hull of the primal points. For simplicity we focus only on the upper envelope part from now on; the lower envelope case is entirely symmetric. Using a standard divide-and-conquer approach, we partition our lines into two groups of size roughly $n/2$ each, and assume that recursive invocations of the algorithm maintain the upper envelopes of these groups. For convenience call the groups red and blue.

In order to produce the upper envelope of all the lines, we have to merge the upper envelopes of the red and blue groups and also certify this merge, so we can detect when it ceases to be valid as the lines move; see [Figure 50.4.4](#).

Conceptually, we can approach this problem by sweeping the envelopes with a vertical line from left to right. We advance to the next red (blue) vertex and determine if it is above or below the corresponding blue (red) edge, and so on. In this process we determine when red is above blue or vice versa, as well as when the two envelopes cross. By stitching together all the upper pieces, whether red or blue, we get a representation of the upper envelope of all the lines.

The certificates used in certifying the above merge are of three flavors:

- x -certificates ($<_x$) are used to certify to x -ordering among the red and blue vertices; these involve four original lines.
- y -certificates ($<_y$) are used to certify that a vertex is above or below an edge of the opposite color; these involve three original lines and are exactly the duals of the CCWcertificates discussed earlier.
- s -certificates ($<_s$) are slope comparisons between pairs of original lines; though these did not arise in our sweep description above, they are needed to make the KDS local [BGH99].

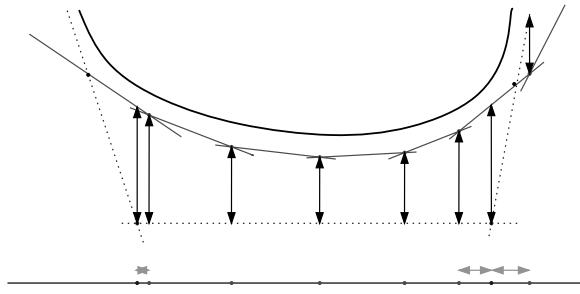


FIGURE 50.4.4

Merging the red and blue upper envelopes. In this example, the red envelope (solid line) is above the blue (dotted line), except at the extreme left and right areas. Vertical double-ended arrows represent y -certificates and horizontal double-ended arrows represent x -certificates, as described below.

Figure 50.4.5 shows examples of how these types of certificates can be used to specify x -ordering constraints and to establish intersection or non-intersection of the envelopes.

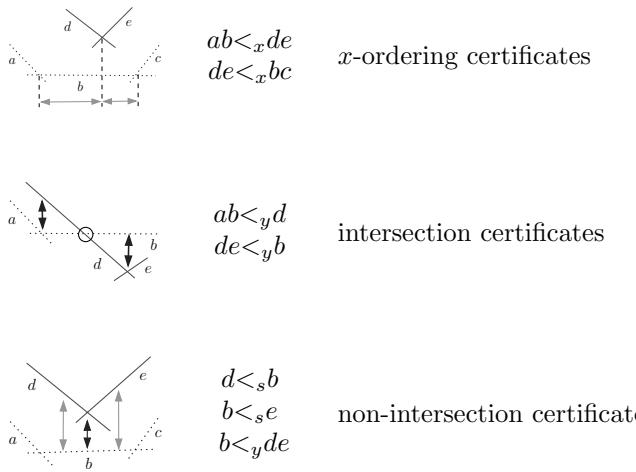


FIGURE 50.4.5

Using the different types of certificates to certify the red-blue envelope merge.

A total of $O(n)$ such certificates suffices to verify the correctness of the upper envelope merge.

Whenever the motion of the lines causes one of these certificates to fail, a local, constant-time process suffices to update the envelope and repair the certification. Figure 50.4.6 shows an example where an y -certificate fails, allowing the blue envelope to poke up above the red.

It is straightforward to prove that this kinetic upper envelope algorithm is responsive, local, and compact, using the logarithmic depth of the hierarchical structure of the certification. In order to bound the number of events processed,

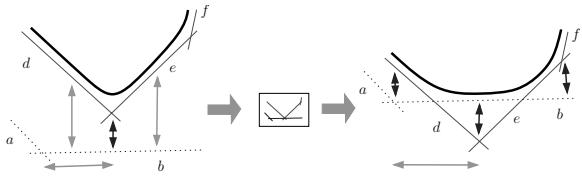


FIGURE 50.4.6

Envelope repair after a certificate failure. In the event shown lines b , d , and e become concurrent, producing a red-blue envelope intersection

however, we must assume that the line motions are polynomial or at least pseudo-algebraic. A proof of efficiency can be developed by extruding the moving lines into space-time surfaces. Using certain well-known theorems about the complexity of upper envelopes of surfaces [Sha94] and the overlays of such envelopes [ASS96] (cf. Chapter 47), it can be shown that in the worst case the number of events processed by this algorithm is near quadratic ($O(n^{2+\epsilon})$). Since the convex hull of even linearly moving points can change $\Omega(n^2)$ times [AGHV01], the efficiency result follows.

No comparable structure is known for the convex hull of points in dimensions $d \geq 3$.

EXTENT PROBLEMS

A number of the original problems for which kinetic data structures were developed are aimed at different measures of how “spread out” a moving set of points in \mathbb{R}^2 is—one example is the convex hull, whose maintenance was discussed in the previous subsection. Other measures of interest include the diameter, width, and smallest area or perimeter bounding rectangle for a moving set S of n points. All these problems can be solved using the kinetic convex hull algorithm above; the efficiency of the algorithms is $O(n^{2+\epsilon})$, for any $\epsilon > 0$. There are also corresponding $\Omega(n^2)$ lower bounds for the number of combinatorial changes in these measures. Surprisingly, the best upper bound known for maintaining the smallest enclosing disk containing S is still near-cubic. Extensions of these results to dimensions higher than two are also lacking.

These costs can be dramatically reduced if we consider approximate extent measures. If we are content with $(1 + \epsilon)$ approximations to the measures, then an approximate smallest orthogonal rectangle, diameter, and smallest enclosing disk can be maintained with a number of events that is a function ϵ only and not of n [AHP01]. For example, the bound of the number of approximate diameter updates in \mathbb{R}^2 under linear motion of the points is $O(1/\epsilon)$.

PROXIMITY PROBLEMS

The fundamental proximity structures in computational geometry are the Voronoi diagram and the Delaunay triangulation (Chapter 23). The edges of the Delaunay triangulation contain the closest pair of points, the closest neighbor to each point, as well as a wealth of other proximity information among the points. From the

kinetic point of view, these are nice structures, because they admit completely local certifications. Delaunay’s 1934 theorem [Del34] states that if a local empty sphere condition is valid for each $(d-1)$ -simplex in a triangulation of points in \mathbb{R}^d , then that triangulation must be Delaunay. This makes it simple to maintain a Delaunay triangulation under point motion: an update is necessary only when one of these empty sphere conditions fails. Furthermore, whenever that happens, a local retiling of space (of which the classic “edge-flip” in R^2 is a special case; cf. Section 25.3) easily restores Delaunayhood. Thus the KDS for Delaunay (and Voronoi) that follows from this theorem is both responsive and efficient—in fact, each KDS event is an external event in which the structure changes. Though no redundant events happen, an exact upper bound for the total number of such events in the worst-case is still elusive even in R^2 , where the best upper bound known is nearly cubic, while the best lower bound is only quadratic [AGMR98].

This principle of a set of easily checked local conditions that implies a global property has been used in kinetizing other proximity structures as well. For instance, in the *power diagram* [Aur87] of a set of disjoint balls, the two closest balls must be neighbors [GZ98]—and this diagram can be kinetized by a similar approach. Voronoi diagrams of more general objects, such as convex polytopes, have also been investigated. For example, in R^2 [GSZ00] shows how to maintain a compact Voronoi-like diagram among moving disjoint convex polygons; again, a set of local conditions is derived which implies the global correctness of this diagram. As the polygons move, the structure of this diagram allows one to know the nearest pair of polygons at all times.

In many applications the exact L_2 -distance between objects is not needed and more relaxed notions of proximity suffice. Polyhedral metrics (such as L_1 or L_∞) are widely used, and the normal unit ball in L_2 can be approximated arbitrarily closely by polyhedral approximants. It is more surprising, however, that if we partition the space around each point into a set of polyhedral cones and maintain a number of directional nearest neighbors to each point in each cone, then we can still capture the globally closest pair of points in the L_2 metric. By directional neighbors here we mean that we measure distance only along a given direction in that cone. This geometric fact follows from a packing argument and is exploited in [BGZ97] to give a different method for maintaining the closest pair of points in \mathbb{R}^d . The advantage of this method is that the kinetic events are changes of the sorted order of the points along a set of directions fixed *a priori*, and therefore the total number of events is provably quadratic.

TRIANGULATIONS AND TILINGS

Many areas in scientific computation and physical modeling require the maintenance of a triangulation (or more generally a simplicial complex) that approximates a manifold undergoing deformation. The problem of maintaining the Delaunay triangulation of moving points in the plane mentioned above is a special case. More generally, local re-triangulations are necessitated by collapsing triangles, and sometimes required in order to avoid undesirably “thin” triangles. In certain cases the number of nodes (points) may also have to change in order to stay sufficiently faithful to the underlying physical process; see, for example, [CDES01]. Because in general a triangulation meeting certain criteria is not unique or canonical, it becomes more difficult to assess the efficiency of kinetic algorithms for solving such

problems. The lower-bound results in [ABdB⁺99] indicate that one cannot hope for a subquadratic bound on the number of events in the worst case for the maintenance of *any* triangulation, even if a linear number of additional Steiner points is allowed.

There is large gap between the desired quadratic upper bound and the current state of the art. Even for maintaining an arbitrary triangulation of a set of n points moving linearly in the plane, the best-known algorithm processes $O(n^{7/3})$ events [ABG⁺00] in the worst case. The algorithm actually maintains a pseudotriangulation of the convex hull of the point set and then a triangulation of each pseudotriangle. Although there are only $O(n^2)$ events in the pseudotriangulation, some of the events change too many triangles because of high-degree vertices. Unless additional Steiner points are allowed, there are point configurations for which high-degree vertices are inevitable and therefore some of the events will be expensive. A more clever, global argument is needed to prove a near-quadratic upper bound on the total number of events in the above algorithm. Methods that choose to add additional points, on the other hand, have the burden of defining appropriate trajectories for these Steiner points as well. Finally, today no triangulation that guarantees certain quality on the shapes of triangles as well as a subcubic bound on the number of retiling events is known.

COLLISION DETECTION

Kinetic methods are naturally applicable to the problem of collision detection between moving geometric objects. Typically collisions occur at irregular intervals, so that fixed-time stepping methods have difficulty selecting an appropriate sampling rate to fit both the numerical requirements of the integrator as well as those of collision detection. A kinetic method based on the discrete events that are the failures of relevant geometric conditions can avoid the pitfalls of both oversampling and undersampling the system. For two moving convex polygons in the plane, a kinetic algorithm where the number of events is a function of the relative separation of the two polygons is given in [EGSZ99]. The algorithm is based on constructing certain outer hierarchies on the two polygons. Analogous methods for 3D polytopes were presented in [GXZ01], together with implementation data.

A tiling of the free space around objects can serve as a proof of non-intersection of the objects. If such a tiling can be efficiently maintained under object motion, then it can be the basis of a kinetic algorithm for collision detection. Several papers have developed techniques along these lines, including the case of two moving simple polygons in the plane [BEG⁺99], or multiple moving polygons [ABG⁺00, KSS00]. These developments all exploit deformable pseudotriangulations of the free space—tilings which undergo fewer combinatorial changes than, for example, triangulations. An example from [ABG⁺00] is shown in [Figure 50.4.7](#). The figure shows how the pseudotriangulation adjusts by local retiling to the motion of the inner quadrilateral. The approach of [ABG⁺00] maintains a canonical pseudotriangulation, while others are based on letting a pseudotriangulation evolve according to the history of the motion. It is unclear at this point which is best. An advantage of all these methods is that the number of certificates needed is close to the size of the min-link separating subdivision of the objects, and thus sensitive to how intertwined the objects are.

Deformable objects are more challenging to handle. Classical methods, such as

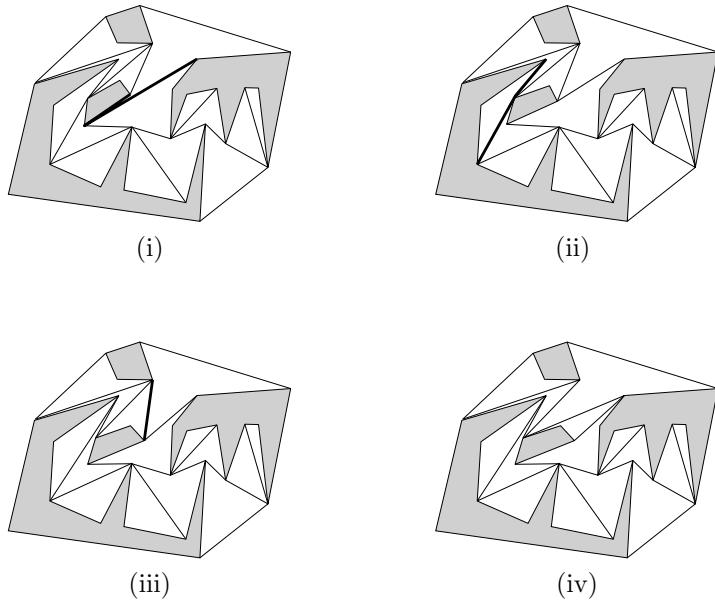


FIGURE 50.4.7

Snapshots of the mixed pseudotriangulation of [ABG⁺ 00]. As the center trapezoid-like polygon moves to the right, the edges corresponding to the next about-to-fail certificate are highlighted.

bounding volume hierarchies [GLM96], become expensive, as the fixed object hierarchies have to be rebuilt frequently. One possibility for mitigating this cost is to let the hierarchies themselves deform continuously, by having the bounding volumes defined implicitly in terms of object features. Such an approach was developed for flexible linear objects (such as rope or macromolecules), using combinatorially defined sphere hierarchies in [GNRZ02]. In that work a bounding sphere is defined not in the usual way, via its center and radius, but in an implicit combinatorial way, in terms of four feature points of the enclosed object geometry. As the object deforms these implicitly defined spheres automatically track their assigned features, and therefore the deformation. Of course the validity of the hierarchy has to be checked at each time step and repaired if necessary. What helps here is that the implicitly defined spheres change their combinatorial description rather infrequently, even under extreme deformation. An example is shown in Figure 50.4.8 where the rod shown is bent substantially, yet only the top-level sphere needs to update its description.

The pseudotriangulation-based methods above can also be adapted to deal with object deformation.

CONNECTIVITY AND CLUSTERING

Closely related to proximity problems is the issue of maintaining structures encoding connectivity among moving geometric objects. Connectivity problems arise frequently in *ad hoc* mobile communication and sensor networks, where the viability of links may depend on proximity or direct line-of-sight visibility among the sta-

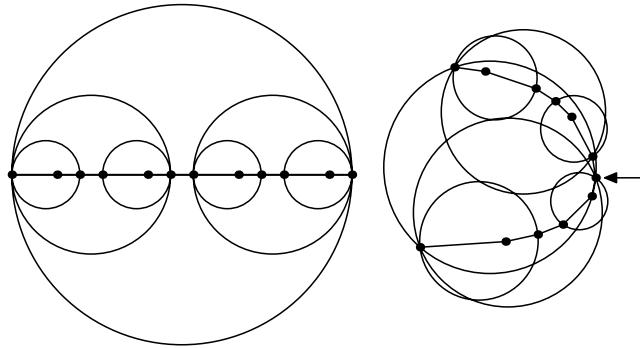


FIGURE 50.4.8

A thin rod bending from a straight configuration, and a portion of its associated bounding sphere hierarchy. The combinatorially defined sphere hierarchy is stable under deformation. Only the top level sphere differs between the two conformations.

tions desiring to communicate. With some assumptions, the communication range of each station can be modeled by a geometric region, so that two stations can establish a link if and only if their respective regions overlap. (See also Section 58.4.3.) There has been work on kinetically maintaining the connected components of the union of a set of moving geometric regions for the case of rectangles [HS99] and unit disks [GHSZ00].

Clustering mobile nodes is an essential step in many algorithms for establishing communication hierarchies, or otherwise structuring *ad hoc* networks. Nodes in close proximity can communicate directly, using simpler protocols; correspondingly, well-separated clusters can reuse scarce resources, such as the same frequency or time-division multiplexing communication scheme, without interference. Maintaining clusters of mobile nodes requires a tradeoff between the tightness, or optimality of the clustering, and its stability under motion. In [GGH⁺01b] a randomized clustering scheme is discussed based on an iterated leader-election algorithm that produces a number of clusters within a constant factor of the optimum, and in which the number of cluster changes is also asymptotically optimal. This scheme was used in [GGH⁺01a] to maintain a routing graph on mobile nodes that is always sparse and in which communication paths exist that are nearly as good as those in the full communication graph.

Another fundamental kinetic question is the maintenance of a minimum spanning tree (MST) among n mobile points in the plane, closely related to earlier work on parametric spanning trees [FBSE96] in a graph whose edge weights are functions of a parameter λ (λ is time in the kinetic setting). Since the MST is determined by the sorted order of the edge weights in the graph, a simple algorithm can be obtained by maintaining the sorted list of weights and some auxiliary data structures (such an algorithm is quadratic in the graph size, or $O(n^4)$ in our case). This was improved when the weights are linear functions of time to nearly $O(n^{11/6})$ (sub-quadratic) for planar graphs or other minor-closed families [AEGH98]. When the weights are the Euclidean distances between moving points, only approximation algorithms are known and the best event bounds are nearly cubic [BGZ97]. For many other optimization problems on geometric graphs, such as shortest paths for example, the corresponding kinetic questions are wide open.

VISIBILITY

The problem of maintaining the visible parts of the environment when an observer is moving is one of the classic questions in computer graphics and has motivated significant developments, such as binary space partition trees, the hardware depth buffer, etc. The difficulty of the question increases significantly when the environment itself includes moving objects; whatever visibility structures accelerate occlusion culling for the moving observer must now themselves be maintained under object motion.

Binary space partitions (BSP) are hierarchical partitions of space into convex tiles obtained by performing planar cuts (Chapter 28.8.2). Tiles are refined by further cuts until the interior of each tile is free of objects or contains geometry of limited complexity. Once a BSP tree is available, a correct visibility ordering for all geometry fragments in the tiles can be easily determined and incrementally maintained as the observer moves. A kinetic algorithm for visibility can be devised by maintaining a BSP tree as the objects move. The key insight is to certify the correctness of the BSP tree through certain combinatorial conditions, whose failure triggers localized tree rearrangements — most of the classical BSP construction algorithms do not have this property. In \mathbb{R}^2 , a randomized algorithm for maintaining a BSP of moving disjoint line segments is given in [AGMV00]. The algorithm processes $O(n^2)$ events, the expected cost per tree update is $O(\log n)$, and the expected tree size is $O(n \log n)$. The maintenance cost increases to $O(n \lambda_{s+2}(n) \log^2 n)$ [AEG98] for disjoint moving triangles in \mathbb{R}^3 (s is a constant depending on the triangle motion). Both of these algorithms are based on variants of vertical decompositions (many of the cuts are parallel to a given direction). It turns out that in practice these generate “sliver-like” BSP tiles that lead to robustness issues [Com99].

As the pioneering work on the visibility complex has shown [PV96], another structure that is well suited to visibility queries in \mathbb{R}^2 is an appropriate pseudotriangulation. Given a moving observer and convex moving obstacles, a full radial decomposition of the free space around the observer is quite expensive to maintain. One can build pseudotriangulations of the free space that become more and more like the radial decomposition as we get closer to the observer. Thus one can have a structure that compactly encodes the changing visibility polygon around the observer, while being quite stable in regions of the free space well occluded from the observer [OH02].

RESULT SUMMARY

We summarize in Table 50.4.1 the efficiency bounds on the main KDSs discussed above.

OPEN PROBLEMS

As mentioned above, we still lack efficient kinetic data structures for many fundamental geometric questions. Here is a short list of such open problems:

1. Find an efficient (and responsive, local, and compact) KDS for maintaining the convex hull of points moving in dimensions $d \geq 3$.

TABLE 50.4.1 Bounds on the number of combinatorial changes.

STRUCTURE	BOUNDS ON EVENTS	SOURCE
Convex hull	$\Omega(n^{2+\epsilon})$	[BGH99]
Pseudotriangulation	$O(n^2)$	[ABG ⁺ 00]
Triangulation (arb.)	$\Omega(n^{7/3})$	[ABG ⁺ 00]
MST	$O(n^{11/6} \log^{3/2} n)$	[AEGH98]
BSP	$\tilde{O}(n^2)$	[AGMV00, AEG98]

2. Find an efficient KDS for maintaining the smallest enclosing disk in $d \geq 2$. For $d = 2$, a goal would be an $O(n^{2+\epsilon})$ algorithm.
3. Establish tighter bounds on the number of Voronoi diagram events, narrowing the gap between quadratic and near-cubic.
4. Obtain a near-quadratic bound on the number of events maintaining an arbitrary triangulation of linearly moving points.
5. Maintain a kinetic triangulation with a guarantee on the shape of the triangles, in subcubic time.
6. Find a KDS to maintain the MST of moving points under the Euclidean metric achieving subquadratic bounds.

Beyond specific problems, there are also several important structural issues that require further research in the KDS framework. These include:

Recovery after multiple certificate failures. We have assumed up to now that the KDS assertion cache is repaired after each certificate failure. In many realistic scenarios, however, it is impossible to predict exactly when certificates will fail because explicit motion descriptions may not be available. In such settings we may need to sample the system and thus we must be prepared to deal with multiple (but hopefully few) certificate failures at each time step. A general area of research that this suggests is the study of how to efficiently update common geometric structures, such as convex hulls, Voronoi and Delaunay diagrams, arrangements, etc., after “small motions” of the defining geometric objects.

There is also a related subtlety in the way that a KDS assertion cache can certify the value, or a computation yielding the value, of the attribute of interest. Suppose our goal is to certify that a set of moving points in the plane, in a given circular order, always forms a convex polygon. A plausible certificate set for convexity is that all interior angles of the polygon are convex. See [Figure 50.4.9](#). In the normal KDS setting where we can always predict accurately the next certificate failure, it turns out that the above certificate set is sufficient, *as long as at the beginning of the motion the polygon was convex*. One can draw, however, nonconvex self-intersecting polygons all of whose interior angles are convex, as also shown in the same figure. The point here is that a standard KDS can offer a *historical* proof of the convexity of the polygon by relying on the fact that the certificate set is valid

and that the polygon was convex during the prior history of the motion. Indeed the counterexample shown cannot arise under continuous motion without one of the angle certificates failing first. On the other hand, if an oracle can move the points when ‘we are not looking,’ we can wake up and find all the angle certificates to be valid, yet our polygon need not be convex. Thus in this oracle setting, since we cannot be sure that no certificates failed during the time step, we must insist on *absolute* proofs — certificate sets that in any state of the world fully validate the attribute computation or value.

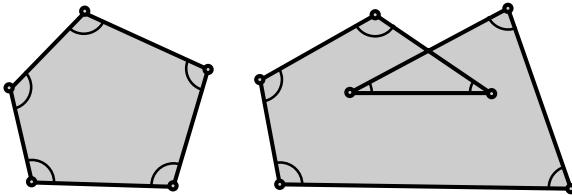


FIGURE 50.4.9
Certifying the convexity of a polygon.

Hierarchical motion descriptions. Objects in the world are often organized into groups and hierarchies and the motions of objects in the same group are highly correlated. For example, though not all points in an elastic bouncing ball follow exactly the same rigid motion, the trajectories of nearby points are very similar and the overall motion is best described as the superposition of a global rigid motion with a small local deformation. Similarly, the motion of an articulated figure, such as a man walking, is most succinctly described as a set of relative motions, say that of the upper right arm relative to the torso, rather than by giving the trajectory of each body part in world coordinates.

What both of these examples suggest is that there can be economies in motion description, if the motion of objects in the environment can be described as a superposition of terms, some of which can be shared among several objects. Such hierarchical motion descriptions can simplify certificate evaluations, as certificates are often local assertions concerning nearby objects, and nearby objects tend to share motion components. For example, in a simple articulated figure, we may wish to assert $\text{ccw}(A, B, C)$ to indicate that an arm is not fully extended, where \overline{AB} and \overline{BC} are the upper and lower parts of the arm, respectively. Evaluating this certificate is clearly better done in the local coordinate frame of the upper arm than in a world frame—the redundant motions of the legs and torso have already been factored out.

Motion sensitivity. As already mentioned, the motions of objects in the world are often highly correlated and it behooves us to find representations and data structures that exploit such motion coherence. It is also important to find mathematical measures that capture the degree of coherence of a motion and then use this as a parameter to quantify the performance of motion algorithms. If we do not do this, our algorithm design may be aimed at unrealistic worst-case behavior, without capturing solutions that exploit the special structure of the motion

data that actually arise in practice — as already discussed in a related setting in [dBK⁺97]. Thus it is important to develop a class of kinetic *motion-sensitive* algorithms, whose performance can be expressed as a function of how coherent the motions of the underlying objects are.

Noncanonical structures. The complexity measures for KDSs mentioned earlier are more suitable for maintaining *canonical* geometric structures, which are uniquely defined by the position of the data, e.g., convex hull, closest pair, and Delaunay triangulation. In these cases the notion of external events is well defined and is independent of the algorithm used to maintain the structure. On the other hand, as we already discussed, suppose we want to maintain a triangulation of a moving point set. Since the triangulation of a point set is not unique, the external events depend on the triangulation being maintained, and thus depend on the algorithm. This makes it difficult to analyze the efficiency of a kinetic triangulation algorithm. Most of the current approaches for maintaining noncanonical structures artificially impose canonicality and maintain the resulting canonical structure. But this typically increases the number of events. So it is entirely possible that methods in which the current form of the structure may depend on its past history can be more efficient. Unfortunately, we lack mathematical techniques for analyzing such history-dependent structures.

50.5 QUERYING MOVING OBJECTS

Continuous tracking of a geometric attribute may be more than is needed for some applications. There may be time intervals during which the value of the attribute is of no interest; in other scenarios we may be just interested to know the attribute value at certain discrete query times. For example, given n moving points in \mathbb{R}^2 , we may want to pose queries asking for all points inside a rectangle R at time t , for various values of R and t , or for an interval of time Δt , etc. Such problems can be handled by a mixture of kinetic and static techniques, including standard range-searching tools such as partition trees and range trees [dBvK⁺00]. They typically involve tradeoffs between evolving indices kinetically, or prebuilding indices for static snapshots. An especially interesting special case is when we want to be able answer queries about the near future faster than those about the distant future—a natural desideratum in many real-time applications.

A number of other classical range-searching structures, such as k -d-trees and R -trees have recently been investigated for moving objects [AHPP02, AGG02].

50.6 SOURCES AND RELATED MATERIALS

SURVEYS

Results not given an explicit reference above may be traced in these surveys.

[Gui98]: An early, and by now somewhat dated, survey of KDS work.

[AG⁺ar]: A report based on an NSF-ARO workshop, addressing several issues on modeling motion from the perspective of a variety of disciplines.

[Gui02]: A “popular-science” type article containing material related to the costs of sensing and communication for tracking motion in the real world.

RELATED CHAPTERS

[Chapter 22: Convex hull computations](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 25: Triangulations and mesh generation](#)

[Chapter 35: Collision detection](#)

REFERENCES

- [ABdB⁺99] P.K. Agarwal, J. Basch, M. de Berg, L.J. Guibas, and J. Hershberger. Lower bounds for kinetic planar subdivisions. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 247–254, 1999.
- [ABG⁺00] P.K. Agarwal, J. Basch, L.J. Guibas, J. Hershberger, and L. Zhang. Deformable free space tiling for kinetic collision detection. In *Proc. 4th Workshop Algorithmic Found. Robot.*, pages 83–96, 2000.
- [AEG98] P.K. Agarwal, J. Erickson, and L.J. Guibas. Kinetic BSPs for intersecting segments and disjoint triangles. In *Proc. 9th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 107–116, 1998.
- [AEGH98] P.K. Agarwal, D. Eppstein, L.J. Guibas, and M. Henzinger. Parametric and kinetic minimum spanning trees. In *Proc. 39th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 596–605, 1998.
- [AGG02] P.K. Agarwal, J. Gao, and L.J. Guibas. Kinetic medians and *kd*-trees. In *Proc. 10th Europ. Sympos. Algorithms*, pages 5–16, 2002.
- [AGHV01] P.K. Agarwal, L.J. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. *Discrete Comput. Geom.*, 26:353–374, 2001.
- [AGMR98] G. Albers, L.J. Guibas, J.S.B. Mitchell, and T. Roos. Voronoi diagrams of moving points. *Internat. J. Comput. Geom. Appl.*, 8:365–380, 1998.
- [AGMV00] P.K. Agarwal, L.J. Guibas, T.M. Murali, and J.S. Vitter. Cylindrical static and kinetic binary space partitions. *Comp. Geometry, Theory and Appl.*, 16:103–127, 2000.
- [AG⁺ar] P.K. Agarwal, L.J. Guibas, et al. Algorithmic issues in modeling motion. *ACM Comput. Surv.*, 34:550–572, 2002.
- [AHP01] P.K. Agarwal and S. Har-Peled. Maintaining approximate extent measures of moving points. In *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 148–157, 2001.
- [AHPP02] P.K. Agarwal, S. Har-Peled, and C.M. Procopiuc. Star-tree: An efficient self-adjusting index for moving points. In *Workshop Algorithms Engineering*, 2002.
- [ASS96] P.K. Agarwal, O. Schwarzkopf, and M. Sharir. The overlay of lower envelopes and its applications. *Discrete Comput. Geom.*, 15:1–13, 1996.

- [Ata85] M.J. Atallah. Some dynamic computational geometry problems. *Comput. Math. Appl.*, 11:1171–1181, 1985.
- [Aur87] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM J. Comput.*, 16:78–96, 1987.
- [BEG⁺99] J. Basch, J. Erickson, L.J. Guibas, J. Hershberger, and L. Zhang. Kinetic collision detection between two simple polygons. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 327–336, 1999.
- [BGH99] J. Basch, L.J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Algorithms*, 31:1–28, 1999.
- [BGZ97] J. Basch, L.J. Guibas, and L. Zhang. Proximity problems on moving points. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 344–351, 1997.
- [CDES01] H.L. Cheng, T.K. Dey, H. Edelsbrunner, and J. Sullivan. Dynamic skin triangulation. In *Proc. 12th SIAM Sympos. Discrete Algorithms*, pages 47–56, 2001.
- [Com99] J.L.D. Comba. *Kinetic vertical decomposition trees*. Ph.D. thesis, Stanford Univ., 1999.
- [dBK⁺97] M. de Berg, M.J. Katz, A.F. van der Stappen, and J. Vleugels. Realistic input models for geometric algorithms. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 294–303, 1997.
- [dBvK⁺00] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*, 2nd edition. Springer-Verlag, Berlin, 2000.
- [Del34] B.N. Delaunay. Sur la sphère vide. A la memoire de Georges Voronoi. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.
- [EGSZ99] J. Erickson, L.J. Guibas, J. Stolfi, and L. Zhang. Separation-sensitive collision detection for convex objects. In *Proc. 10th ACM-SIAM Symp. Discrete Algorithms*, pages 102–111, 1999.
- [FBSE96] D. Fernàndez-Baca, G. Slutzki, and D. Eppstein. Using sparsification for parametric minimum spanning tree problems. *Nordic J. Computing*, 3:352–366, 1996.
- [GGH⁺01a] J. Gao, L.J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. 2nd Annu. ACM Sympos. Ad-Hoc Networking Comput. (MobiHoc)*, pages 45–55, Oct. 2001.
- [GGH⁺01b] J. Gao, L.J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 190–198, 2001.
- [GHSZ00] L.J. Guibas, J. Hershberger, S. Suri, and L. Zhang. Kinetic connectivity for unit disks. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 331–340, 2000.
- [GJS96] P. Gupta, R. Janardan, and M. Smid. Fast algorithms for collision and proximity problems involving moving geometric objects. *Comput. Geom. Theory Appl.*, 6:371–391, 1996.
- [GLM96] S. Gottschalk, M.C. Lin, and D. Manocha. OBB-Tree: A hierarchical structure for rapid interference detection. In *Proc. ACM Conf. SIGGRAPH 96*, pages 171–180, 1996.
- [GNRZ02] L.J. Guibas, A. Nguyen, D. Russell, and L. Zhang. Collision detection for deforming necklaces. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 33–42, 2002.
- [GSZ00] L.J. Guibas, J. Snoeyink, and L. Zhang. Compact Voronoi diagrams for moving convex polygons. In *Proc. Scand. Workshop Alg. Data Struct.*, volume 1851 of *Lecture Notes Comput. Sci.*, pages 339–352. Springer-Verlag, Berlin, 2000.

- [Gui98] L.J. Guibas. Kinetic data structures—A state of the art report. In P.K. Agarwal, L.E. Kavraki, and M. Mason, editors, *Proc. Workshop Algorithmic Found. Robot.*, pages 191–209. A.K. Peters, Wellesley, 1998.
- [Gui02] L.J. Guibas. Sensing, tracking, and reasoning with relations. *IEEE Signal Proc. Mag.*, pages 73–85, 2002.
- [GXZ01] L.J. Guibas, F. Xie, and L. Zhang. Kinetic collision detection: Algorithms and experiments. In *Proc. Internat. Conf. Robot. Autom.*, pages 2903–2910, 2001.
- [GZ98] L.J. Guibas and L. Zhang. Euclidean proximity and power diagrams. In *Proc. 10th d. Conf. Comput. Geom.*, pages 90–91, 1998.
- [HS99] J. Hershberger and S. Suri. Kinetic connectivity of rectangles. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 237–246, 1999.
- [Kah91] S. Kahan. A model for data in motion. In *Proc. 23th Annu. ACM Sympos. Theory Comput.*, pages 267–277, 1991.
- [KSS00] D.G. Kirkpatrick, J. Snoeyink, and B. Speckmann. Kinetic collision detection for simple polygons. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 322–330, 2000.
- [OH02] O. Hall-Holt. *Kinetic visibility*. Ph.D. thesis, Stanford Univ., 2002.
- [PV96] M. Pocchiola and G. Vegter. The visibility complex. *Internat. J. Comput. Geom. Appl.*, 6:279–308, 1996.
- [Sha94] M. Sharir. Almost tight upper bounds for lower envelopes in higher dimensions. *Discrete Comput. Geom.*, 12:327–345, 1994.
- [ST95] E. Schömer and C. Thiel. Efficient collision detection for moving polyhedra. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 51–60, 1995.
- [ST96] E. Schömer and C. Thiel. Subquadratic algorithms for the general collision detection problem. In *Abstracts 12th European Workshop Comput. Geom.*, pages 95–101. Universität Münster, 1996.

51 PATTERN RECOGNITION

Joseph O'Rourke and Godfried T. Toussaint

INTRODUCTION

The two fundamental problems in a pattern recognition system are feature extraction (shape measurement) and classification. The problem of extracting a vector of shape measurements from a digital image can be further decomposed into three subproblems. The first is the image segmentation problem, i.e., the separation of objects of interest from their background. The cluster analysis methods discussed in Section 51.1 are useful here. The second subproblem is that of finding the objects in the segmented image. An example is the location of text lines in a document as illustrated in Section 51.2. The final subproblem is extracting the shape information from the objects detected. Here there are many tools available depending on the properties of the objects that are to be classified. The Hough transform (Section 51.2), polygonal approximation (Section 51.3), shape measurement (Section 51.4), and polygon decomposition (Section 51.6), are some of the favorite tools used here. Important to many of these tasks is finding a nice viewpoint from which extraction is robust and efficient (Section 51.5). Proximity graphs, discussed in Section 51.2, are used extensively for both cluster analysis and shape measurement.

The classification problem involves the design of efficient decision rules with which to classify the feature vector. The most powerful decision rules are the nonparametric rules which make no assumptions about the underlying distributions of the feature vectors. Of these the nearest-neighbor (NN) rule, treated in Section 51.7, is the most well known. This section covers the three main issues concerning NN-rules: how to edit the data set so that little storage space is used, how to search for the nearest neighbor of a vector efficiently, and how to estimate the future performance of a rule both reliably and efficiently.

51.1 CLUSTER ANALYSIS AND CLASSIFICATION

GLOSSARY

Cluster analysis problem: Partitioning a collection of n points in some fixed-dimensional space into $m < n$ groups that are “natural” in some sense. Here m is usually much smaller than n .

Image segmentation problem: Partitioning the pixels in an image into “meaningful” regions, usually such that each region is associated with one physical object.

Dendrogram: A tree representing a hierarchy of categories or clusters.

Hierarchical clustering algorithms: Those that produce a dendrogram whose root is the whole set and whose leaves are the individual points.

Graph-theoretic clustering: Clustering based on deleting edges from a proximity graph.

K-means clustering: Tracking clusters over time by comparing new data with old means.

Data mining: Intelligent and efficient information retrieval from huge (and often unstructured) data repositories.

Classical cluster analysis requires partitioning points into natural clumps. “Natural” may mean that the clustering agrees with human perception, or it may simply optimize some natural mathematical measure of similarity or distance so that points that belong to one cluster are similar to each other and points far away from each other are assigned to different clusters. It is not surprising that such a general and fundamental tool has been applied to widely different subproblems in pattern recognition. One obvious application is to the determination of the number and description of classes in a pattern recognition problem where the classes are not known *a priori*, such as disease classification in particular or taxonomy in general. In this case m is not known beforehand and the cluster analysis reveals it.

A fundamental problem in pattern recognition of images is the segmentation problem: distinguishing the figure from the background. Clustering is one of the most powerful approaches to image segmentation, applicable even to complicated images such as those of outdoor scenes. In this approach each pixel in the $N \times N$ image is treated as a complicated object by associating it with a local neighborhood. For example, we may define the 5×5 neighborhood of pixel p_{ij} , denoted by $N_5[p_{ij}]$, as $\{p_{mn} \mid i - 2 \leq m \leq i + 2, j - 2 \leq n \leq j + 2\}$. We next measure k properties of p_{ij} by making k measurements in $N_5[p_{ij}]$. Such measurements may include various moments of the intensity values (grey levels) found in $N_5[p_{ij}]$, etc. Thus each pixel is mapped into a point in k -dimensional pixel-space. Performing a cluster analysis of all the resulting $N \times N$ points in pixel-space yields the desired partitioning of the pixels into categories. Labeling each category of pixels with a different color then produces the segmentation.

See [Gor96] for a survey of clustering methods in which the objects in one class are not only required to be similar to each other but must satisfy other constraints on the distances between the objects or on the topology of the resulting dendrograms.

HIERARCHICAL CLUSTERING

In taxonomy there is no special number m that we want to discover; rather the goal is the production of a *dendrogram* (tree) that grows all the way from one cluster to n clusters and shows us at once how good a partitioning is obtained for any number of clusters between one and n . Such methods are referred to as *hierarchical* methods. They fall into two groups: *agglomerative* (bottom-up, merging) and *divisive* (top-down, splitting). Furthermore, each of these methods can be used with a variety of distance measures between subsets to determine when to merge or split. Two popular agglomerative clustering algorithms are the *single-link* and the *complete-linkage* (or farthest-neighbor) algorithm. In the former, cluster similarity is measured by the minimum of the distances between all pairs

of elements, one in each subset, whereas in the latter similarity is measured by the maximum pairwise distance. The complete-linkage criterion tends to produce more compact clusters, while the single-link criterion can suffer from chaining [JMF99]. Krznaric and Levcopoulos [KL02] show that a complete-linkage hierarchy can be computed in optimal $O(n \log n)$ time and $O(n)$ space.

GRAPH-THEORETIC CLUSTERING

The most powerful methods of clustering in difficult problems, which give results having the best agreement with human performance, are the graph-theoretic methods [JT92]. The idea is simple: Compute some proximity graph (such as the minimum spanning tree) of the original points. Then delete (in parallel) any edge in the graph that is much longer (according to some criterion) than its neighbors. The resulting forest is the clustering (each tree in this forest is a cluster). Proximity graphs have also been used effectively to design cluster validity tests [PB97].

K-MEANS TYPE CLUSTERING

There are many applications where we know that there are exactly K clusters, for example in character recognition. However, because of external factors such as the variations in people's hand-printing over time, or a change in the parameters of a machine due to wear or weather conditions, the clusters must be "tracked" over time. One of the most popular methods for doing this is the ***K-means algorithm***. The k -means algorithm searches for k cluster centroids in \mathbb{R}^d with the property that the mean squared (Euclidean) distance between each point and its nearest centroid is minimized [Mac67]. A determining characteristic of this approach is that the number of clusters k is fixed. A typical heuristic starts with an initial partition, computes centers, assigns data points to their nearest center, recomputes the centroids, and iterates until convergence is achieved according to some criterion. (A neural network equivalent was developed by Kohonen [Koh95, BB95].) Unfortunately, this attractively simple algorithm's performance depends upon the initial partitioning, and in fact can be forced into a suboptimal solution of arbitrarily high approximation ratio (with respect to the optimal mean squared distance). This led recently to developing algorithms with performance guarantees. Matoušek achieved an $O(n \log^k n)$ ϵ -approximation algorithm under the assumption that k and d are fixed [Mat00]. This was improved to an $(9 + \epsilon)$ -approximation algorithm via a center-swap heuristic with this provable upper bound [KM⁺02]. On the other hand, there is some evidence that the exact k -means algorithm can be implemented to work well in practice for small d [PM99].

A variation on the k -means algorithm permits splitting and merging of clusters. This technique is employed by the ISODATA algorithm (Interactive Self-Organizing Data Analysis Technique) [Jen96].

DISTANCES BETWEEN SETS

A fundamental computational primitive of almost all clustering algorithms is the frequent computation of some distance measure between two sets (subsets) of points. This is especially so in the popular hierarchical methods discussed above. There exists a large variety of distance and more general similarity measures for this purpose. Here we mention a few. Most efficient algorithms for distance between sets apply only in \mathbb{R}^2 but some methods extend to higher dimensions; see [Smi00]. Let P and Q be two convex polygons of n sides each. Two distance measures should be distinguished: the minimum *element* distance, the smallest distance between a vertex or edge of P and a vertex or edge of Q , and the minimum *vertex* distance, the minimum distance between a vertex of P and a vertex of Q . The minimum element distance can be computed in $O(\log n)$ time [Ede85]. On the other hand, computation of the minimum vertex distance between P and Q has a linear lower bound. For the case of two nonintersecting convex polygons several different $O(n)$ time algorithms are available, and the same bound can be achieved for crossing convex polygons [Tou84].

Let R be a set of n red points and B a set of n blue points in the plane. Both the minimum distance and the maximum distance between R and B can be computed in $O(n \log n)$ time. For the latter problem, two algorithms are available. The first [TM82] is simple but does not appear to generalize to higher dimensions. The second [BT83] works by reducing the maximum distance problem between R and B to computing the diameter of 81 convex polygons. These are obtained by computing the convex hulls of the unions of 81 carefully selected subsets of R and B , and then reporting the maximum of these 81 diameters as the maximum distance. These ideas can be extended to obtain efficient algorithms for all dimensions [Rob93]. Therefore, any improvement in high-dimensional diameter algorithms automatically improves maximum-distance algorithms.

DATA MINING

The explosion of the Web has given new impetus to intelligent and efficient information retrieval from huge, often unstructured, data repositories. This activity has become known as ***data mining***. Clustering is often used to segment the data set, to support subsequent “mining.” The k -means algorithm and its variants remain popular, not only in geometric domains (e.g., in geological databases [JMF99], celestial databases [PM99], image databases, and so on), but even for text databases. For example, Fayyad et al. [FRB98] report some success on a Reuter’s database using the 302 most frequently occurring words, i.e., the dimension is $d = 302$. There is some movement in the literature away from point distances for categorical attributes, for which the iterative centroid-based clustering algorithms are often inappropriate. For example, recent work uses “links” [GRS00] or context-based measures [DM00], which places this work close to graph-theoretic clustering. A related new direction is finding “unusual” strings of ACTG characters within the human genome [ABL02].

51.2 EXTRACTING SHAPE FROM DOT PATTERNS

HOUGH TRANSFORMS

The **Hough transform** was originally proposed (and patented) as an algorithm to detect straight lines in digital images [Lea92]. The method may be used to detect any parametrizable pattern, and has been generalized to locate arbitrary shapes in images. The basic idea is to let each above-threshold pixel in the image vote for the points in the parameter space that could generate it. The votes are accumulated in a quantized version of the parameter space, with high vote tallies indicating detection.

Examples

1. *Lines.* Let the lines in the (x, y) image plane be parametrized as $y = mx + b$. Then a pixel at (x_0, y_0) is a witness to a line passing through it, that is, an (m, b) pair satisfying $y_0 = mx_0 + b$. Thus, (x_0, y_0) votes for all those (m, b) pairs: the line in parameter space dual to the pixel.
2. *Circles.* Parametrize the circles by their center and radius, (x_c, y_c, r) . Then a pixel (x_0, y_0) gives evidence for all the parameter triples on the circular cone in the 3-parameter space with apex at $(x_0, y_0, 0)$ and axis parallel to the r -axis.
3. *Object location.* Suppose a known but arbitrary shape S is expected to be found in an image, and its most likely location is sought. For translation-only, the parameter space represents the location of some fixed point of S . Each pixel in the image of the right shading or color votes for all those translations that cover it appropriately.

The above approaches are not necessarily optimal for the tasks listed. For example, it was shown in [CT77] that nonuniform (maximum entropy) quantization with $\rho - \theta$ parametrization for lines is superior to uniform quantization with $m - b$ parametrization.

The demands of high-dimensional vote accumulators have engendered the study of *dynamic quantization* of the parameter space, and **geometric hashing**. This latter technique has features in the image vote for each member of a library of shapes by hashing into a table using the feature coordinates as a key. Each table entry may correspond to several shapes at several displacements, but all receive votes. Geometric hashing has been applied with some success to the *molecular docking* problem [LW88]; see also [MSW02].

More recently, new variants of the Hough transform inspired by results in computational geometry have appeared. In [AK96] two such algorithms are presented and studied with respect to the tradeoff that exists between computational complexity and effectiveness of line detection. They obtain efficient implementations by using the plane-sweep paradigm.

TEXT-LINE ORIENTATION INFERENCE

In an automated document analysis system, given a block (paragraph) of text, the text-line orientation inference problem consists of determining the location and direction of the lines of text on the page. Almost always these lines are either horizontal (e.g., English) or vertical (e.g., Chinese). The fundamental geometric property that allows this problem to be solved is the fact that according to a universal typesetting convention guided by ease of reading, characters are printed closer together within textlines than between textlines.

One of the most successful, robust, skew-tolerant, simple, and elegant techniques for text-line orientation inference was proposed by Ittner [Itt93]. Assume that the given text block B consists of n black connected components (characters). The three key steps in the procedure are: (1) idealize each character by a point, thus obtaining a set S of n points in the plane; (2) construct the Euclidean minimum spanning tree $\text{MST}(S)$ of the n points obtained in (1); and (3) determine the textline orientation by analysis of the distribution of the orientations of the edges in $\text{MST}(S)$. Step (1) is done by computing the center of the bounding box of each character. Cheriton and Tarjan proposed a simple algorithm for computing the MST of a graph in $O(E)$ time where E is the number of edges in the graph [CT76]. Fortunately there are many graphs defined on S (usually belonging to the class of proximity graphs [JT92]) that have the property that they contain the $\text{MST}(S)$ and also have $O(n)$ edges. For these graphs the Cheriton-Tarjan algorithm runs in $O(n)$ time.

RELATIVE NEIGHBORHOOD GRAPHS

Relative neighborhood graphs (RNG's), introduced in [Tou80b], capture proximity between points by connecting nearby points with a graph edge. The many possible notions of "nearby" (in several metrics) lead to a variety of related graphs. It is easiest to view the graphs as connecting points only when certain regions of space are empty. Let V be a set of n points in \mathbb{R}^d .

GLOSSARY

$\delta(p,q)$: the distance between two points p and q .

L_p : The distance metric L_p defined as $\delta_p(x,y) = (\sum_{i=1}^d |x_i - y_i|^p)^{1/p}$. For L_1 this reduces to $\sum_{i=1}^d |x_i - y_i|$, and for L_∞ , to $\max_{1 \leq i \leq d} |x_i - y_i|$.

Ball $B(x,r)$: The open ball $B(x,r) = \{y \mid \delta(x,y) < r\}$.

Nearest-neighbor Graph NNG(V): The graph with vertex set V and an edge (p,q) iff $B(p, \delta(p,q)) \cap V = \emptyset$.

Lune $L(p,q)$: $L(p,q) = B(p, \delta(p,q)) \cap B(q, \delta(p,q))$.

Relative neighborhood graph RNG(V): The graph with vertex set V and an edge (p,q) iff $L(p,q) \cap V = \emptyset$. Thus the edge is present iff

$$\delta(p,q) \leq \max_{v \in V \setminus \{p,q\}} \{\delta(p,v), \delta(q,v)\}.$$

Gabriel graph GG(V): The graph with vertex set V and an edge (p, q) iff

$$B\left(\frac{p+q}{2}, \frac{\delta(p,q)}{2}\right) \cap V = \emptyset.$$

β -lune: $L_\beta(p, q) = B\left(p(1 - \frac{\beta}{2}) + q\frac{\beta}{2}, \frac{\beta}{2}\delta(p, q)\right) \cap B\left(q(1 - \frac{\beta}{2}) + p\frac{\beta}{2}, \frac{\beta}{2}\delta(p, q)\right)$.

β -skeleton: The graph $G_\beta(V)$ with vertex set V and an edge (p, q) iff $L_\beta(p, q) \cap V = \emptyset$. The range $1 \leq \beta \leq 2$ is especially relevant, with $G_2(V) = \text{RNG}(V)$ and $G_1(V) = \text{GG}(V)$.

Sphere-of-influence graph SIG(V): Let C_p be the circle centered on p with radius equal to the distance to a nearest neighbor of p . SIG(V) has node set V and an edge (p, q) iff C_p and C_q intersect in at least two points.

Minimum-Weight Triangulation MWT(V): A triangulation of minimum total edge length.

The *relative neighborhood graph* connects two points if the *lune* they determine is empty of points of V . The *Gabriel graph* is defined similarly, but with the *diameter sphere* required to be empty. The β -skeletons are a continuous generalization of the Gabriel graph. These graph definitions are motivated by various applications: computer vision, texture discrimination, geographic analysis, pattern analysis, cluster analysis, and others.

Proximity graphs form a nested hierarchy, a version of which was first established in [Tou80b]:

THEOREM 51.2.1 Hierarchy

In any L_p metric, for a fixed set V and $1 \leq \beta \leq 2$,

$$\text{NNG} \subseteq \text{MST} \subseteq \text{RNG} \subseteq G_\beta \subseteq \text{GG} \subseteq \text{DT}.$$

MST is the minimum spanning tree, and DT the Delaunay triangulation, of V .

Neighborhood graphs can have at most $O(n^2)$ edges, and $\Theta(n^2)$ complexity is achieved in many instances. For the L_2 metric in \mathbb{R}^2 , however, RNG's (and their relatives) have linear size, which increases their usefulness. See Table 51.2.1.

TABLE 51.2.1 Size of relative neighborhood graphs.

DIM	METRIC	SIZE
2	$L_p, 1 < p < \infty$	$\Theta(n): \in [n - 1, 3n - 6]$
≥ 2	L_1, L_∞	$\Theta(n^2)$
3	L_2	$O(n^{4/3})$
$d > 4$	L_p	$\Theta(n^2)$

The many applications of neighborhood graphs have stimulated considerable effort on developing efficient algorithms for constructing them. The RNG has the most applications and has received the most attention. $O(n^3)$ time complexity is trivial to achieve. Exploiting the fact that the Delaunay triangulation is a superset

leads easily to $O(n^2)$ in the plane. Further development is more difficult. Two milestones in algorithm development were Supowit's $O(n \log n)$ algorithm for L_2 in \mathbb{R}^2 [Sup83], and Agarwal and Matoušek's near- $O(n^{3/2})$ algorithm for general position points in \mathbb{R}^3 [AM92]. See Table 51.2.2.

TABLE 51.2.2 Relative neighborhood graph algorithms.

DIM	METRIC	COMPLEXITY
2	L_2	$O(n \log n)$
	L_1, L_∞	$O(n \log n)$
3	L_2	$O(n^{3/2+\epsilon})$
	L_1, L_∞	$O(n \log^2 n)$
d	L_2	$O(n^{2(1-\frac{1}{d+1})+\epsilon})$
	L_1, L_∞	$O(n \log^{d-1} n)$

A related graph is the *sphere-of-influence* (SIG) graph. It has at most $15n = O(n)$ edges in \mathbb{R}^2 and can be computed in $\Theta(n \log n)$ time. The SIG can serve as a type of graph-theoretical “primal sketch.” It, in some sense, explains the dot-pattern version of the Müller-Lyer illusion.

See the survey [JT92] for further details on neighborhood graphs.

An important connection between β -skeletons and minimum-weight triangulations (MWT) was discovered by Keil [Kei94]: for $\beta = \sqrt{2}$, $G_\beta \subseteq \text{MWT}$. This was subsequently sharpened to $\beta = \frac{1}{6}\sqrt{2\sqrt{3} + 45}$, which is optimal [WY99].

OPEN PROBLEMS

1. What is the maximum number of edges of an RNG in \mathbb{R}^3 ? It is known that it has at most $O(n^{4/3})$ edges, but no supra-linear lower bound is known.
2. That the SIG has at most $15n$ edges in the plane follows from a theorem of Erdős and Panitzsch [Sos99], but the best lower bound is $9n$. It is also known that the SIG has a linear number of edges in any fixed dimension [GPS94], and bounds on the expected number of edges are known [Dwy95], but again tight results are not available.

51.3 POLYGONAL APPROXIMATION

Let $P = (p_1, p_2, \dots, p_n)$ be a polygonal curve or chain in the plane, consisting of n points p_i joined by line segments $p_i p_{i+1}$. In general P may be closed and self-intersecting. Polygonal curves occur frequently in pattern recognition either as representations of the boundaries of figures or as functions of time representing, e.g., speech. In order to reduce the complexity of costly processing operations,

it is often desirable to approximate a curve P with one that is composed of far fewer segments, yet is a close enough replica of P for the intended application. Some methods of reduction attempt smoothing as well. An important instance of the problem is to determine a new curve $Q = (q_1, q_2, \dots, q_m)$ such that (1) m is significantly smaller than n ; (2) the q_j are selected from among the p_i ; and (3) any segment $q_j q_{j+1}$ that replaces the chain $q_j = p_r, \dots, p_s = q_{j+1}$ is such that the distance between $q_j q_{j+1}$ and each p_k , $r \leq k \leq s$, is less than some predetermined error tolerance ω . Different notions of distance, or error criteria, lead to different algorithmic issues. Moreover, for each distance definition, there are two constrained optimization problems that are of interest, Min-# and Min- ϵ .

GLOSSARY

Distance from point p to segment s : Minimum distance from p to any point of s .

Parallel-strip criterion: All the vertices p_i, \dots, p_t lie in a parallel strip of width 2ω whose center line is collinear with $q_j q_{j+1}$ [ET94].

Segment criterion: For each p_k , $r \leq k \leq s$, the distance from p_k to $q_j q_{j+1}$ is less than ω [MO88, CC96].

Min-# problem: Given the error tolerance ω , find a curve $Q = (q_1, \dots, q_m)$ satisfying the constraint such that m is minimum.

Min- ϵ problem: Given m , find a curve $Q = (q_1, \dots, q_m)$ satisfying the constraint such that the error tolerance is minimized.

The main results obtained are listed in Table 51.3.1.

TABLE 51.3.1 Polygonal approximation algorithm time complexities.

ERROR CRITERION	MIN-#	MIN- ϵ
Parallel strip	$O(n^2 \log n)$	$O(n^2 \log^2 n)$
Segment	$O(n^2)$	$O(n^2 \log n)$

The task of polygonal approximation has been given new significance in three-dimensions for its importance in simplifying polyhedral models in computer graphics, e.g., [CVM⁺96, LE97]. This topic is covered in detail in [Chapter 54](#).

OPEN PROBLEMS

Find algorithms for the strip criterion problems that match those for the segment problems. Perhaps quadratic performance is achievable for all four problems.

ITERATIVE ENDPOINT FITTING

There are many algorithms in both the pattern recognition and automated cartography literatures that are intended to yield approximations with few, but not necessarily the minimum number of, segments. This suffices for many applications,

and often can be obtained more efficiently in both time and space complexities. One of the most popular such algorithms is *iterative endpoint fitting*, which employs the parallel-strip criterion. Given a tolerance ω , the algorithm first attempts to use only one segment $Q = (p_1, p_n)$ to approximate P . If the error tolerance is exceeded, then that vertex of P whose distance to the line through p_1p_n is maximum is chosen to divide P into two subchains. The procedure is then recursively applied to these subchains. This procedure can be implemented to run in $O(n \log n)$ time.

51.4 SHAPE MEASUREMENT AND REPRESENTATION

MEDIAL AXIS

GLOSSARY

Medial axis: The set of points of a region P with more than one closest point among the boundary points ∂P of the region. Equivalently, it is the set of centers of maximal balls, i.e., of balls in P that are themselves not enclosed in another ball in P .

Voronoi diagram: The partition of a polygonal region P into cells each consisting of the points closer to a particular open edge or vertex than to any other.

The *medial* or *symmetric axis* was introduced by Blum [Blu67] to capture biological shape, and it has since found many other applications, for example, to geometric modeling (*offset* computations; see [Section 47.2](#)) and to mesh generation [SNT⁺92] ([Section 22.4](#)). It provides a central “skeleton” for an object that has found many uses. It connects to several other mathematical concepts, including the *cut locus* and most importantly, the Voronoi diagram ([Chapter 20](#)).

The medial axis of a convex polygon is a tree with each vertex a leaf. For a nonconvex polygon, the medial axis may have a parabolic arc associated with each reflex vertex ([Figure 47.1.5](#)). The basic properties of the medial axis were detailed by Lee [[Lee82](#)], who showed that the medial axis of a polygon P is just the Voronoi diagram minus the Voronoi edges incident to reflex vertices, and provided an $O(n \log n)$ algorithm for constructing it. After a long search by the community, an $O(n)$ algorithm was obtained [[CSW95](#)]. The simplest implementations are, however, quadratic [[YR91](#)]. See [Table 52.4.1](#).

The medial axis has also found much use in image processing, where its digital computation is via *thinning algorithms*. Pioneered by Rosenfeld, these algorithms are very simple and easily parallelized [[Cyc94](#)].

The definition of medial axis extends to \mathbb{R}^d . Some work has explored \mathbb{R}^3 [[SPB95](#)], but currently the lack of reliable software hampers extensive applications.

POINT PATTERN MATCHING

Exact point pattern matching is an interesting algorithmic question related to string matching, but pattern recognition applications usually require some type of approximate matching. Two types may be distinguished [[AG00](#)]: one-to-one matching, and Hausdorff matching.

GLOSSARY

One-to-one approximate matching: Let two finite sets of points A and B have the same cardinality. One-to-one matching requires finding a transformation (of a given type) of B such that each point of B is mapped to within a distance of ϵ of a matched point of A . The matches are either determined by *labels* on the points, or the points are *unlabeled* and the match is to be discovered.

Decision problem: Given ϵ , is there such a matching?

Optimization problem: Find the minimum ϵ for which an approximate matching exists.

Hausdorff distance: For two finite sets A and B , perhaps of different cardinalities, the largest of the between-sets nearest-neighbor distances.

Hausdorff matching: Find a transformation of B that minimizes the Hausdorff distance from A .

The most combinatorially interesting point matching (unrealistically) demands exact matching. One version of this is the **congruent subset detection problem**: Given a pattern set A of m points, find all subsets of a background set B of n points that are congruent to A . Solving this in the plane relies on the unsolved Erdős problem of bounding the number of unit-distance pairs among n points, whose best upper bound is $O(n^{4/3})$ ([Chapter 10](#)). Important variations are obtained by acting on the pattern by some group, e.g., translations. Results here are surveyed in [Bra02], from which the results shown in Table 51.4.1 are gathered ($\alpha()$ is the near-constant inverse Ackermann function; cf. Chapter 47).

TABLE 51.4.1 Subset detection of m points among n points.

GROUP	DIM	MATCHES	ALGORITHM
Congruence	2	$O(n^{4/3})$	$O(mn^{4/3} \log n)$
Congruence	3	$\Omega(n^{4/3})$	$O(mn^{5/3} \log n 2^{O(\alpha(n)^2)})$
Translation	d	$n - \Theta(n^{1-1/k}), k \leq d$	$O(mn \log n)$
Homothets	d	$O(n^{1+1/k}), k \leq d$	$O(mn^{1+1/d} \log n)$
Similarity	d	$O(n^d)$	$O(mn^d \log n)$
Affine	d	$O(n^{d+1})$	$O(mn^{d+1} \log n)$

A window-restricted version of the problem led Brass to pose the following interesting Conjecture:

- Any set of n points in the plane contains only $O(n)$ empty congruent triangles.

There are sets with $\binom{n}{3}$ empty triangles.

Results on **one-to-one approximate matching** algorithms obtained in [AM⁺88] for a variety of permissible transformations are shown in [Table 51.4.2](#).

Hausdorff matching leads to analysis of envelopes of *Voronoi surfaces*. Typical results are shown in [Table 51.4.3](#). Here we show the complexities when $|A| = |B| = n$, although the algorithms work for sets of different cardinalities.

TABLE 51.4.2 One-to-one point matching in two dimensions.

MOVEMENTS	LABLED	ϵ	COMPLEXITY
Translation	labeled	dec, opt	$O(n)$
Translation	unlabeled	decision	$O(n^6)$
Translation	unlabeled	optimization	$O(n^6 \log n)$
Rotation	labeled	decision	$O(n \log n)$
Rotation	labeled	optimization	$O(n^2)$
Trans+rot+refl	labeled	decision	$O(n^3 \log n)$
Trans+rot+refl	unlabeled	decision	$O(n^8)$

TABLE 51.4.3 Hausdorff matching in the L_2 metric.

MOVEMENTS	DIM	COMPLEXITY
Translation	2	$O(n^3 \log n)$
Translation + rotation	2	$O(n^6 \log n)$
Translation	3	$O(n^{5+\epsilon})$

Another type of matching is *order type matching* (cf. Section 5.2). In [GP83], an $O(n^3)$ algorithm is given for finding all matchings between two planar point configurations in which their order types agree.

SYMMETRY DETECTION

Symmetry is an important feature in the analysis and synthesis of shape and form and has received considerable attention in the pattern recognition and computer graphics literatures. In [WWV85] an $O(n \log n)$ algorithm is presented for computing the rotational symmetries of polygons and polyhedra of n vertices, but the constant in \mathbb{R}^3 is very large. Jiang and Bunke [JB91] give a simple and practical $O(n^2)$ time algorithm for polyhedra. One of the earliest applications of computational geometry to symmetry detection was the algorithm of Akl and Toussaint [AT79] to check for polygon similarity. Since then attention has been given to other aspects of symmetry and for objects other than polygons. Sugihara [Sug84] shows how a modification of the planar graph-isomorphism algorithm of Hopcroft and Tarjan can be used to find all symmetries of a wide class of polyhedra in $O(n \log n)$ time.

A related topic is centers of symmetry. Given a convex polygon P , associate with each point p in P the minimum area of the polygon to the left of any chord through p . The maximum over all points in P is known as **Winternitz's measure of symmetry** and the point p^* that achieves this maximum is called the **center of area**. Diaz and O'Rourke [DO94] show that p^* is unique and propose an algorithm for computing p^* in time $O(n^6 \log^2 n)$. For a survey of the early work on detecting symmetry, see [Ead88].

THE ALPHA HULL

The α -shape \mathcal{S}_α of a set S of n points in \mathbb{R}^3 is a polyhedral surface whose boundary is a particular collection of triangles, edges, and vertices determined by the points of S [EM94]. It is similar in spirit to the β -skeleton of Section 51.2 in that it is a parametrized collection of shapes determined by an empty balls condition, but it emphasizes the external rather than the internal structure of the set. Let T be a subset of S of 1, 2, or 3 points. Then the convex hull of T , $\text{conv}(T)$, is part of the boundary $\partial\mathcal{S}_\alpha$ of the α -shape iff the surface of some ball of radius α includes exactly the points of T while its interior is empty of points of S . Thus a triangle $\text{conv}(T)$ is part of $\partial\mathcal{S}_\alpha$ iff there is an open α -ball that can “erase” all of the triangle but leave its vertices. \mathcal{S}_α is defined for all $0 \leq \alpha \leq \infty$, with $\mathcal{S}_0 = S$ and $\mathcal{S}_\infty = \text{conv}(S)$.

Every edge and triangle of \mathcal{S}_α is present in the Delaunay triangulation DT of S , and every edge and triangle in DT is present in some \mathcal{S}_α . If α is varied continuously over its full range starting from ∞ , the convex hull of S is gradually “eaten away” by smaller and smaller α -ball erasers, eventually exposing the original set of points. In between, the α -shape bounds a subcomplex of DT that represents the shape of S .

The alpha shape has been used for cluster analysis, molecular modeling, and the analysis of medical data, among other applications. High-quality code is available [EF99]; now CGAL includes alpha shapes in its basic library ([Chapter 65](#)).

51.5 NICE VIEWPOINTS AND PROJECTIONS

A robot navigating in 3D space faces a variety of pattern recognition problems that involve classifying objects modeled as polyhedra. A polyhedral object in 3D space is often well represented by a set of points (vertices) and line segments (edges) that act as its features. The feature extraction process involves obtaining *nice* viewpoints of the polyhedron. By a nice viewpoint of an object we mean a projective view in which all (or most) of the features of the object, relevant for some task, are clearly visible. Such a view is often called a nondegenerate view or projection. A recent survey of this topic can be found in [Tou00].

GLOSSARY

Nice viewpoint: A projection of a 3D object (set of points, etc) onto a plane such that it has some desirable special property.

Knot diagram: A regular projection of a polygon in 3-dimensions onto a plane.

Degeneracies: Properties of objects such as three points collinear.

General position: A configuration of an object such that some specified degeneracies are absent.

Orthogonal projection: A projection from a point at infinity.

Perspective projection: A projection from a point not at infinity.

Robust algorithm: One that works correctly even in the presence of degeneracies.

Regular projection: An orthogonal projection of S such that no three points of S project to the same point on H , and no vertex of S projects to the same point on H as any other point of S .

Wirtinger projections: Regular projections in which no two consecutive edges of the 3D polygon have collinear projections.

Robust nondegenerate projection: A projection that remains nondegenerate even if the object is slightly perturbed.

Decision problem: Given an object and a projection of it, does the projection contain a degeneracy?

Computation problem: Given an object, compute a projection that does not contain a specified degeneracy.

Optimization problem: Given an object, compute the most robust nondegenerate projection.

REGULAR PROJECTIONS

The earliest work on nondegenerate orthogonal projections appears to be in the area of knot theory. Let S be a set of n disjoint line segments in \mathbb{R}^3 specified by the cartesian coordinates of their endpoints (vertices of S) and let H be a plane. Let SH be the orthogonal projection of S onto H . An orthogonal projection of S is said to be *regular* if no three points of S project to the same point on H and no vertex of S projects to the same point on H as any other point of S [Liv93]. This definition implies that for disjoint line segments (1) no point of SH corresponds to more than one vertex of S , (2) no point of SH corresponds to a vertex of S and an interior point of an edge of S , and (3) no point of SH corresponds to more than two interior points of edges of S . Therefore the only crossing points (intersections) allowed in a regular projection are those points that belong to the interiors of precisely two edges of S . This condition is crucial for the successful visualization and manipulation of knots [Liv93].

Regular projections of 3D polygons were first studied by the knot theorist K. Reidemeister [Rei32] in 1932 who showed that all 3D polygons (knots) admit a regular projection, and in fact almost all projections of polygons are regular. Reidemeister however was not concerned with computing regular projections. The computational aspects of regular projections of knots were first investigated by Bose et al., [BGRT99] under the real RAM model of computation. Given a polygonal object (geometric graph, wire-frame or skeleton) in \mathbb{R}^3 (such as a simple polygon, knot, skeleton of a Voronoi diagram or solid model mesh), they consider the problem of computing several “nice” orthogonal projections of the object. One such projection, well known in the graph-drawing literature, is a projection with few crossings. They consider the most general polygonal object, i.e., a set of n disjoint line segments, and show that deciding whether it admits a crossing-free projection can be done in $O(n^2 \log n + k)$ time and $O(n^2 + k)$ space, where k is the number of intersections among a set of “forbidden” quadrilaterals on the direction sphere, and $k = O(n^4)$. This implies for example that, given a knot, one can determine if there exists a plane on which its projection is a simple polygon, within the same complexity. Furthermore, if such a projection does not exist, a minimum-crossing projection can be found in $O(n^4)$ time and $O(n^2)$ space. They showed (independently of Reidemeister) that a set of line segments in space (which includes polygonal objects

as special cases) always admits a regular projection, and that such a projection can be obtained in $O(n^3)$ time. A description of the set of all directions which yield regular projections can be computed in $O(n^3 \log n + k \log n)$ time, where k is the number of intersections of a set of quadratic arcs on the direction sphere and $k = O(n^6)$. Finally, when the objects are polygons and trees in space, they consider monotonic projections, i.e., projections such that every path from the root of the tree to every leaf is monotonic in some common direction on the projection plane. For example, given a polygonal chain P , they can determine in $O(n)$ time if P is monotonic on the projection plane, and in $O(n \log n)$ time they can find all the viewing directions with respect to which P is monotonic. In addition, in $O(n^2)$ time, they can determine all directions for which a given tree or a given simple polygon is monotonic.

COMPUTER VISION

In the computer vision field there is both a theoretical [BWR93] interest in nondegenerate projections and a practical one [DWT99]. The theoretical work resembles the work described in the previous section in that it is assumed that the object consists of idealized points and line segments or polygons and polyhedra. A tool used for computing viewpoints from which the maximum number of faces of a solid polyhedron is visible, is the *aspect graph* [PD90] ([Chapter 28](#)).

WIRTINGER PROJECTIONS

That certain types of nondegenerate orthogonal projections of 3D polygons always exist for some directions of projection was rediscovered by Bhattacharya and Rosenfeld [BR94] for a restricted class of regular projections. Those regular projections, in which it is also required that no two consecutive edges of the 3D polygon have collinear projections, are known as *Wirtinger projections*. Bose et al. [BGRT99] study the complexity of computing a single Wirtinger projection as well as constructing a description of all such projections for the more general input consisting of disjoint line segments. These results include therefore results for 3D chains, polygons, trees and geometric graphs in general. The description of all projections allows one to obtain Wirtinger projections that optimize additional properties. For example, one may be interested in obtaining the most robust projection in the sense that it maximizes the deviation of the viewpoint required to violate the Wirtinger property.

VISUALIZATION

In computer graphics one is interested in visualizing objects well, and therefore *nice* views and nondegenerate views are major concerns. For example, Kamada and Kawai [KK88] proposed a method to obtain nice projections by making sure that in the projection, parallel line segments on a plane in 3D project as far away from each other as possible. Intuitively, the viewer should be as orthogonal as possible to every face of the 3D object. Of course this is not possible and therefore they suggest minimizing (over all faces) the maximum angle deviation between a normal to the face and the line of sight from the viewer. They then propose an

algorithm to solve this problem in $O(n^6 \log n)$ time, where n is the number of edges in the polyhedral object in 3D. Gómez et al. [GRT01] reduce this complexity to $O(n^4)$ time. Furthermore, they show that if one is restricted to viewing an object from only a hemisphere, as is the case with a building on top of flat ground, then a further reduction in complexity is possible to $O(n^2)$ time.

REMOVING DEGENERACIES

Algorithms in computational geometry are usually designed for the real RAM (random access machine) assuming that the input is in *general position*. More specifically, the general position assumption implies that the input to an algorithm for solving a specific problem is free of certain degeneracies. Yap [Yap90] has distinguished between intrinsic or *problem-induced* and extrinsic or *algorithm-induced* degeneracies (see also Chapter 41). For example, in computing the convex hull of a set of points in the plane, where “left” turns and “right” turns are fundamental primitives, three collinear points constitute a problem-induced degeneracy. On the other hand, for certain vertical line-sweep algorithms two points with the same x -coordinate constitute an algorithm-induced degeneracy. Computational geometers make these assumptions because doing so makes it not only much easier to design algorithms but often yields algorithms with reduced worst-case complexities. On the other hand, to the implementers of geometric algorithms these assumptions are frustrating. Programmers would like the algorithms to work for any input that they may encounter in practice, regardless of the degeneracies that such an input may contain.

Often a typical computational geometry paper will make a nondegeneracy assumption that can in fact be removed (*without* perturbing the input) by a global rigid transformation of the input (such as a rotation, for example). Once the solution is obtained on the transformed nondegenerate input, the solution can be transformed back trivially (by an inverse rotation) to yield the solution to the original problem. In these situations, by applying suitable *pre-* and *post-* processing steps, one obtains the *exact* solution to the *original* problem using an algorithm that assumes a nondegenerate input, even when that input is in fact degenerate. This approach not only handles algorithm-induced degeneracies via orthogonal projections but some problem-induced degeneracies as well with the aid of perspective projections.

Gómez et al. [GRT01] consider several nondegeneracy assumptions that are typically made in the literature, propose efficient algorithms for performing the suitable rotations that remove these degeneracies, analyze their complexity in the real RAM model of computation and, for some of these problems, give lower bounds on their worst-case complexity. The assumptions considered in [GRT01] are summarized in Tables 51.5.1 and .51.5.2 ($\lambda(\cdot)$ is nearly linear; cf. Chapter 47.4).

PERSPECTIVE PROJECTIONS AND INTRINSIC DEGENERACIES

Intrinsic degeneracies cannot be removed by rotations of the input. If a set of points S in 3D contains three collinear points then so does every orthogonal projection of S . This is where *perspective* projections come to the rescue. However, not all

TABLE 51.5.1 Removing degeneracies: Point sets.

PROBLEM	DECISION	COMPUTATION	OPTIMIZATION
2D	$\Theta(n \log n)$	$O(n \log n)$	$O(n^2 \log n)$ time, $O(n^2)$ space
No two on vertical line			$O(n^2)$ time, space with floor functions
3D	$\Theta(n \log n)$	$O(n \log n)$	$O(n^2 \log n)$ time, $O(n^2)$ space
No two with same x -coordinate			$O(n^4)$ time, space
No two with same x , y or z -coord	$\Theta(n \log n)$	$O(n \log n)$	OPEN
No three on vertical plane	(3SUM-hard) $O(n^2)$ time, space	(3SUM-hard) $O(n^2)$ time, space $O(n^3)$ time, $O(n)$ space	$O(n^6)$ time and space

TABLE 51.5.2 Removing degeneracies: Line segments and faces.

PROBLEM	DECISION	COMPUTATION	OPTIMIZATION
LINE SEGMENTS			
2D	$\Theta(n)$	$\Theta(n)$	$O(n \log n)$ time, $O(n)$ space
No vertical			
3D	$\Theta(n)$	$\Theta(n)$	$O(n \log n)$ time, $O(n)$ space
No vertical			
No two on vertical plane	$O(n \log n)$	$O(n^2)$ time, $O(n)$ space	$O(n^4)$ time, space
			$O(n^2 \lambda_6(n^2) \log n)$ time, $O(n^2)$ space
FACES			
No face of polyhedron vertical	$\Theta(n)$	$\Theta(n)$	$O(n^2)$ time, space
			$O(n \lambda_6(n) \log n)$ time, $O(n)$ space

intrinsic degeneracies can be removed with perspective projections. Intrinsic degeneracies that can be removed via perspective projections are called *quasi-intrinsic degeneracies* [HS97, GHS⁺⁰¹].

Gómez et al. [GHS⁺⁰¹] consider computing nondegenerate *perspective* projections of sets of points and line segments in 3D space. For sets of points they give algorithms for computing perspective projections such that (1) all points have distinct x -coordinates, (2) all points have both distinct x - and y -coordinates, (3) no three points in the projection are collinear, and (4) no four points in the projection are cocircular. For sets of line segments they present an algorithm for computing

a perspective projection with no two segments parallel. All their algorithms have time and space complexities bounded by low degree polynomials.

FINITE-RESOLUTION MODELS OF VIEW DEGENERACY

View degeneracy is a central concern in robotics where a robot must navigate and recognize objects based on views of the scene at hand [DPR92a, DPR92b]. In the idealized world assumed in the previous sections, degenerate views are not much of a problem if a viewpoint is chosen at random, since almost all projections are not degenerate. On the other hand, real world digital cameras have a finite resolution and therefore view degeneracy can no longer be ignored [KF87].

OPEN PROBLEMS

A more practical approach would give some thickness to the objects, i.e., consider the points as little balls and the edges of the polyhedra as thin cylinders, and then to redesign the algorithms accordingly. This may turn out to be rather expensive. In practice it may be much more efficient to perform a half-dozen *random* rotations to obtain a nice projection. After all, for many problems in the idealized infinite precision model, a single random rotation yields a nice projection with probability one. Computing optimal projections on the other hand is another matter. Here approximate algorithms may yield efficient solutions that are near-optimal, but these are open problems.

51.6 POLYGON DECOMPOSITION

COVERS AND PARTITIONS

A typical strategy for recognizing a shape as a particular member of a library of shapes is to decompose the shape into “primitive” parts, and then compare these with the library entries via a similarity function. This has led to considerable effort on decomposing shapes and, in particular, polygons into simpler components.

GLOSSARY

Let P be a polygon.

Cover of P : A collection of sets S_1, \dots, S_k such that $S_1 \cup \dots \cup S_k = P$.

Partition of P : A collection of sets S_1, \dots, S_k with pairwise disjoint interiors such that $S_1 \cup \dots \cup S_k = P$.

Diagonal of P : A segment s between two vertices x and y of P such that $s \subseteq P$ and $s \cap P = \{x, y\}$.

Steiner point of P : A point of P that is not a vertex.

Polygon with holes: A multiply connected region bounded by polygonal chains.

Decompositions may be classified along two primary dimensions: covers or parti-

tions, and with or without Steiner points. A cover permits a polygon in the shape of the symbol ‘+’ to be represented as the union of two rectangles, whereas a minimal partition requires three rectangles, a less natural decomposition. Decompositions without Steiner points use diagonals, and are in general easier to find but less parsimonious. For each of the four types of decomposition, different primitives may be considered. The ones most commonly used are rectangles, convex polygons, star-shaped polygons, spiral polygons, and trapezoids (see [Section 23.1](#) for definitions). Restrictions on the shape of the piece being decomposed are often available; for example, orthogonal polygons for rectangle covers. Lastly, the distinction between simple polygons and polygons with holes is often relevant for algorithms.

Finding minimum covers of polygons is NP-hard in nearly every instance explored. Minimum partitions of polygons are somewhat easier, in that polynomial algorithms exist for convex pieces with Steiner points, star-shaped pieces without Steiner points, and rectangles for orthogonal polygons. See [Section 23.2](#) for specific results.

Shape decomposition has a wide variety of applications, including character recognition (spiral and convex pieces), VLSI design (rectangles), and electron-beam lithography (trapezoids) [AAI86].

OPEN PROBLEM

Finding approximation algorithms with proven bounds with respect to optimality remains largely unexplored for most decomposition problems. There are, however, many heuristic algorithms.

SUM-DIFFERENCE DECOMPOSITIONS

Permitting set subtraction as well as set union leads to natural shape decompositions. This is evident from the field of Constructive Solid Geometry, where shapes are described with CSG trees whose nodes are union or difference operators, and whose leaves are primitive shapes ([Section 47.1](#)). Batchelor developed a similar concept for shape description, the *convex deficiency tree* [Bat80]. For a shape P , the root of this tree is its hull $\text{conv}(P)$, the children of the root the hulls of the convex deficiencies $\text{conv}(P) \setminus P$, and so on [O'R98, p. 98].

Chazelle suggested [Cha79] representing a shape by the difference of convex sets: $A \setminus B$ where A and B are unions of convex polygons. It has been established that finding the minimum number of convex pieces in such a sum-difference decomposition of a multiply connected polygonal region is NP-hard [Con90].

TEXT-BLOCK ISOLATION

The text-block isolation problem consists of extracting blocks of text (paragraphs) from a digitized document. By finding the enclosing rectangles around each connected component (character) and around the entire set of characters we obtain a well structured geometric object, namely, a rectangle with n rectangular “holes.” This problem is ideally suited to a computational geometric treatment. Here we mention an elegant method that analyzes the empty (white) spaces in the document [BJF90]. This approach enumerates all maximal white rectangles implied by

the black rectangles. A white rectangle is called maximal if it cannot be enlarged while remaining outside the black rectangles. Their enumeration algorithm takes $O(n \log n + m)$ time, where m is the number of maximal rectangles generated in the search. In the worst case $m = O(n^2)$. However, using a clever heuristic to exploit some properties of layouts they obtain $O(n)$ expected time.

51.7 NEAREST-NEIGHBOR DECISION RULES

GLOSSARY

Nearest-neighbor decision rule: Classifies a feature vector with the closest sample point in parameter space.

In the typical nonparametric classification problem (see Devroye, Gyorfy and Lugosi [DGL96]) we have available a set of d measurements or observations (also called a feature vector) taken from each member of a data set of n objects (patterns) denoted by $\{X, Y\} = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$, where X_i and Y_i denote, respectively, the feature vector on the i th object and the class label of that object. One of the most attractive decision procedures is the nearest-neighbor rule (1-NN -rule) [FH51]. Let Z be a new pattern (feature vector) to be classified and let X_j be the feature vector in $\{X, Y\} = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ closest to Z . The nearest neighbor decision rule classifies the unknown pattern Z into class Y_j . In the 1960s and 1970s many pattern recognition practitioners resisted using the 1-NN -rule on the grounds of the mistaken assumptions that (1) all the data $\{X, Y\}$ must be stored in order to implement such a rule, (2) to determine Y_j , distances must be computed between the unknown vector Z and all the members of $\{X, Y\}$, and (3) such a rule is difficult to implement in parallel using a neural network. Computational geometry research in the 1980s and 1990s along with faster and cheaper hardware has made the NN -rules a practical reality [Tou02].

MINIMAL-SIZE TRAINING-SET CONSISTENT SUBSETS

A question that has received a lot of attention in the past fifty years concerns the problem of reducing the number of patterns in the training set $\{X, Y\}$ without degrading the performance of the decision rule. In 1968 Hart was the first to propose an algorithm for reducing the size of the stored data for the nearest neighbor decision rule [Har68]. Hart defined a *consistent* subset of the data as one that classified the entire set correctly with the nearest neighbor rule. He then proposed an $O(n^3)$ time algorithm that he called *CNN* (Condensed Nearest Neighbor) for selecting a consistent subset by heuristically searching for data that were near the decision boundary. However, the method does not in general yield a minimal-size consistent subset.

The first researchers to deal with computing a *minimal-size* consistent subset were Ritter et al. [RWLI75]. They proposed a procedure they called a *selective* nearest neighbor rule (*SNN*) to obtain a minimal-size consistent subset of $\{X, Y\}$, call it S , with one additional property that Hart's *CNN* does not have. Any

consistent subset C obtained by CNN has the property that every element of $\{X, Y\}$ is nearer to an element in C of the same class than to any element in C of a different class. On the other hand, the consistent subset S of Ritter et al. [RWLI75] has the additional property that every element of $\{X, Y\}$ is nearer to an element in S of the same class than to any element, in the *complete* set, $\{X, Y\}$ of a different class. This additional property of SNN tends to keep points closer to the decision boundary than does CNN , and allows Ritter et al. [RWLI75] to compute the selected subset S without testing all possible subsets of $\{X, Y\}$. Nevertheless, their algorithm still runs in time exponential in n (Wilfong [Wil91]) in the worst case. However, Wilson and Martinez [WM97] and Wilson [WM00] claim that the average running time of SNN is $O(n^3)$. In 1994 Dasarathy [Das94] proposed a complicated algorithm intended to compute a *minimal-size* consistent subset but did not provide a proof of optimality. However, counter-examples to this claim were found by Kuncheva and Bezdek [KB98], Cerverón and Fuertes [CF98] and Zhang and Sun [ZS02]. Wilfong [Wil91] showed in 1991 that the problem of finding the smallest size training-set consistent subset is NP-complete when there are three or more classes. The problem is still open for two classes. Furthermore, he showed that even for only two classes the problem of finding the smallest size consistent *selective* subset (Ritter et al. [RWLI75]) is also NP-complete.

TRAINING-DATA EDITING RULES

Methods have been developed [TBP85] to edit (delete) “redundant” members of $\{X, Y\}$ in order to obtain a subset of $\{X, Y\}$ that implements exactly the same decision boundary as would be obtained using all of $\{X, Y\}$. Such methods depend on the computation of Voronoi diagrams and of other proximity graphs that are subgraphs of the Delaunay triangulation, such as the Gabriel graph. Furthermore, the fraction of data discarded in such a method is a useful measure of the resulting reliability of the rule. If few vectors are discarded the feature space is relatively empty and more training data are needed. During the past twenty years proximity graphs have proven to be very useful both in theory and in practice for solving many of the problems encountered with NN -rules. A description of many of these graphs along with related computational geometry problems can be found in [Tou02].

NEAREST-NEIGHBOR SEARCHING

Another important issue in the implementation of nearest-neighbor decision rules, whether editing has or has not been performed, concerns the efficient search for the nearest neighbor of an unknown vector in order to classify it. Various methods exist for computing a nearest neighbor without computing distances to all the candidates. The problem is in general quite difficult when the dimension is high, which it is for most pattern recognition tasks. Simple brute-force search yields $O(dn)$ query time. To improve upon this, one builds a data structure for the points that supports more efficient queries, often at the expense of space for the data structure. For a set of n points in \mathbb{R}^d , one could construct a Voronoi diagram for the points of size $O(n^{\lceil d/2 \rceil})$ ([Chapter 22](#)), and respond to a query in $O(\log n)$ time. But the exponential space makes this impractical beyond $d \leq 3$. Range searching ([Chapter 31](#)) supports structures with linear space and achieving slightly sublinear

time. But all constants are exponential in d . This has led to intensive work on approximate nearest-neighbor search, where one seeks a point within $(1 + \epsilon)$ of the distance to the true nearest neighbor. An example of an important early milestone along these lines is an algorithm by Arya et al. [AMN⁺98], which constructs a data structure of size $O(dn)$ that can report approximate nearest neighbors in $O(c \log n)$ time, with $c = O(d(1 + d/\epsilon)^d)$. The algorithm traverses down a balanced box-tree decomposition (BDD) of $O(\log n)$ height and stops when the approximation criterion is satisfied. The query time is logarithmic in n but still the constant is exponential in d . The many advances beyond this and similar algorithms with exponential query time or space requirements are described in [Chapter 39](#).

ESTIMATION OF MISCLASSIFICATION

A very important problem in pattern recognition is the estimation of the performance of a decision rule [McL92]. Many geometric problems occur here also, for which computational geometry offers elegant and efficient solutions. For example, a good method of estimating the performance of the NN-rule is to delete each member of $\{X, Y\} = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ in turn and classify it with the remaining set. Geometrically this problem reduces to computing for a given set of points in d -space the nearest neighbor of each (the *all-nearest neighbors problem*). Vaidya [Vai89] gives an $O(n \log n)$ time algorithm to solve this problem.

51.8 SOURCES AND RELATED MATERIAL

SURVEYS

[Tou80a]: A survey of the overlap between pattern recognition and computational geometry.

[Tou91]: A survey of computer vision problems where computational geometry may apply. This survey references several others; the entire collection is of interest as well.

[JMF99]: A survey of clustering from the pattern recognition point of view.

[Tou00]: A survey on computing nice viewpoints of objects in space.

RELATED CHAPTERS

[Chapter 1: Finite point configurations](#)

[Chapter 10: Geometric graphs](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 26: Polygons](#)

[Chapter 34: Point location](#)

[Chapter 36: Range searching](#)

[Chapter 39: Nearest-neighbor searching in high dimensions](#)

[Chapter 41: Robust geometric computation](#)

REFERENCES

- [AAI86] Ta. Asano, Te. Asano, and H. Imai. Partitioning a polygonal region into trapezoids. *J. Assoc. Comput. Mach.*, 33:290–312, 1986.
- [ABL02] A. Apostolico, M.E. Bock, and S. Lonardi. Monotony of surprise and large-scale quest for unusual words (extended abstract). In E.W. Myers, editor, *Internat. Conf. Research Comput. Molecular Biology*, pages 22–31, 2002. ACM.
- [AG00] H. Alt and L.J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier North-Holland, Amsterdam, 2000.
- [AK96] Te. Asano and N. Katoh. Variants for the Hough transform for line direction. *Comput. Geom. Theory Appl.*, 6:231–252, 1996.
- [AM92] P.K. Agarwal and J. Matoušek. Relative neighborhood graphs in three dimensions. *Comput. Geom. Theory Appl.*, 2:1–14, 1992.
- [AMN⁺98] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. Assoc. Comput. Mach.*, 45:891–923, 1998.
- [AM⁺88] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity and symmetries of geometric objects. *Discrete Comput. Geom.*, 3:237–256, 1988.
- [AT79] S.G. Akl and G.T. Toussaint. Addendum to “An improved algorithm to check for polygon similarity.” *Inform. Process. Lett.*, 8:157–158, 1979.
- [Bat80] B.G. Batchelor. Hierarchical shape description based upon convex hulls of concavities. *J. Cybern.*, 10:205–210, 1980.
- [BB95] L. Bottou and Y. Bengio. Convergence properties of the k -means algorithms. In G. Tesauro and D. Touretzky, editors, *Advances in Neural Information Processing Systems 7*, pages 585–592. MIT Press, Cambridge, 1995.
- [BGRT99] P. Bose, F. Gómez, P. Ramos, and G.T. Toussaint. Drawing nice projections of objects in space. *J. Visual Commun. Image Rep.*, 10:155–172, 1999.
- [BJF90] H.S. Baird, S.E. Jones, and S.J. Fortune. Image segmentation by shape-directed covers. In *Proc. 10th Internat. Conf. Pattern Recognition*, pages 820–825. IEEE Computer Society, 1990.
- [Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.
- [BR94] P. Bhattacharya, and A. Rosenfeld. Polygons in three dimensions. *J. Visual Communication and Image Representation*, 5:139–147, 1994.
- [Bra02] P. Brass. Combinatorial geometry problems in pattern recognition. *Discrete Comput. Geom.*, 28:495–510, 2002.
- [BT83] B.K. Bhattacharya and G.T. Toussaint. Efficient algorithms for computing the maximum distance between two finite planar sets. *J. Algorithms*, 4:121–136, 1983.
- [BWR93] J.B. Burns, R.S. Weiss, and E.M. Riseman. View variation of point-set and line-segment features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:51–68, 1993.
- [CC96] W.S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Internat. J. of Comput. Geom. Appl.*, 6:59–77, 1996.

- [CF98] V. Cerverón and A. Fuertes. Parallel random search and Tabu search for the minimum consistent subset selection problem. In *Lecture Notes Comput. Sci.*, pages 248–259. Springer-Verlag, Berlin, 1998.
- [Cha79] B. Chazelle. *Computational geometry and convexity*. Ph.D. thesis, Dept. Comput. Sci., Yale Univ., New Haven, 1979. Carnegie-Mellon Univ. Report CS-80-150.
- [Con90] H. Conn. *Some polygon decomposition problems*. Ph.D. thesis, Dept. Comput. Sci., J. Hopkins Univ., Baltimore, MD, 1990.
- [CSW95] F. Chin, J. Snoeyink, and C.-A. Wang. Finding the medial axis of a simple polygon in linear time. In *Proc. 6th Annu. Internat. Sympos. Algorithms Comput.*, volume 1004 of *Lecture Notes Comput. Sci.*, pages 382–391. Springer-Verlag, Berlin, 1995.
- [CT76] D. Cheriton and R.E. Tarjan. Finding minimum spanning trees. *SIAM J. Comput.*, 5:724–742, 1976.
- [CT77] Me. Cohen and G.T. Toussaint. On the detection of structures in noisy pictures. *Pattern Recognition*, 9:95–98, 1977.
- [CVM⁺96] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P.K. Agarwal, F.P. Brooks, Jr., and W.V. Wright. Simplification envelopes. In *Proc. ACM Conf. SIGGRAPH 96*, pages 119–128. 1996.
- [Cyc94] J.M. Cychosz. Efficient binary image thinning using neighborhood maps. In P. Heckbert, editor, *Graphics Gems IV*, pages 465–473. Academic Press, Boston, 1994.
- [Das94] B.V. Dasarathy. Minimal consistent set (MCS) identification for optimal nearest neighbor decision system design. *IEEE Trans. Syst. Man Cybern.*, 24:511–517, 1994.
- [DGL96] L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.
- [DM00] G. Das and H. Mannila. Context-based similarity measures for categorical databases. In *Principles of Data Mining and Knowledge Discovery*, pages 201–210, 2000.
- [DO94] M. Díaz and J. O’Rourke. Algorithms for computing the center of area of a convex polygon. *Visual Comput.*, 10:432–442, 1994.
- [DPR92a] S.J. Dickinson, A. Pentland, and A. Rosenfeld. 3d shape recovery using distributed aspect matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14, 1992.
- [DPR92b] S.J. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3d object recognition. *CVGIP: Image Understanding*, 55:130–154, 1992.
- [DWT99] S.J. Dickinson, D. Wilkes, and J.K. Tsotsos. A computational model of view degeneracy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:673–689, 1999.
- [Dwy95] R. Dwyer. The expected size of the sphere-of-influence graph. *Comput. Geom. Theory Appl.*, 5:155–164, 1995.
- [Ead88] P. Eades. Symmetry finding algorithms. In G.T. Toussaint, editor, *Computational Morphology*, pages 41–51. North-Holland, Amsterdam, 1988.
- [Ede85] H. Edelsbrunner. Computing the extreme distances between two convex polygons. *J. Algorithms*, 6:213–224, 1985.
- [EF99] H. Edelsbrunner and P. Fu. <http://www.alphashapes.org/alpha/>. Release 4.1 (1996), 1999.
- [EM94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13:43–72, 1994.
- [ET94] D. Eu and G.T. Toussaint. On approximating polygonal curves in two and three dimensions. *CVGIP: Graph. Models Image Process.*, 56:231–246, 1994.

- [FH51] E. Fix and J. Hodges. Discriminatory analysis. Nonparametric discrimination: Consistency properties. Tech. Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
- [FRB98] U.M. Fayyad, C. Reina, and P.S. Bradley. Initialization of iterative refinement clustering algorithms. In *Knowledge Discovery and Data Mining*, pages 194–198, 1998.
- [GHS⁺01] F. Gómez, F. Hurtado, A.A. Sellàres, and G.T. Toussaint. On degeneracies removable by perspective projection. *Internat. J. Math. Alg.*, 2:227–248, 2001.
- [Gor96] A.D. Gordon A survey of constrained classification. *Comput. Stat. Data Anal.*, 21:17–29, 1996.
- [GP83] J.E. Goodman and R. Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12:484–507, 1983.
- [GPS94] L.J. Guibas, J. Pach, and M. Sharir. Sphere-of-influence graphs in higher dimensions. In K. Böröczky and G. Fejes Tóth, editors, *Intuitive Geometry*, Coll. Math. Soc. J. Bolyai 63, pages 131–137. North-Holland, Amsterdam, 1994.
- [GRS00] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *Info. Sys.*, 25:345–366, 2000.
- [GRT01] F. Gómez, S. Ramaswami, and G.T. Toussaint. On computing general position views of data in three dimensions. *J. Visual Commun. Image Rep.*, 12:387–400, 2001.
- [Har68] P.E. Hart. The condensed nearest neighbor rule. *IEEE Trans. Inform. Theory*, 14:515–516, 1968.
- [HS97] F. Hurtado and A.A. Sellàres. Proyecciones perspectivas regulares: Correspondencia por proyección perspectiva entre configuraciones planas de puntos. In *Actas VII Encuentros de Geometría Computacional*, pages 57–70, 1997.
- [Itt93] D.J. Ittner. Automatic inference of textline orientation. In *Proc. 2nd Annu. Sympos. Document Anal. Info. Retrieval*, pages 123–133, 1993.
- [JB91] X.-Y. Jiang and H. Bunke. Determination of the symmetries of polyhedra and an application to object recognition. In *Proc. Comput. Geom.: Methods, Algorithms, Appl.*, volume 553 of *Lecture Notes Comput. Sci.*, pages 113–121. Springer-Verlag, Berlin, 1991.
- [Jen96] J.R. Jensen. *Introductory Digital Image Processing: A Remote Sensing Perspective*, 2nd edition. Prentice-Hall, Englewood Cliffs, 1996.
- [JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, 1999.
- [JT92] J.W. Jaromczyk and G.T. Toussaint. Relative neighborhood graphs and their relatives. *Proc. IEEE*, 80:1502–1517, 1992.
- [KB98] L.I. Kuncheva and J.C. Bezdek. Nearest prototype classification: clustering, genetic algorithms, or random search. *IEEE Trans. Syst. Man Cybern.*, 28:160–164, 1998.
- [Kei94] J.M. Keil. Computing a subgraph of the minimum weight triangulation. *Comput. Geom. Theory Appl.*, 4:13–26, 1994.
- [KF87] J. Kender and D. Freudenstein. What is a degenerate view? In *Proc. 10th Internat. Joint Conf. Artif. Intell.*, pages 801–804, 1987.
- [KK88] T. Kamada and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. *Comput. Vision Graph. Image Process.*, 41:43–56, 1988.
- [KL02] D. Krznicic and C. Levcopoulos. Optimal algorithms for complete linkage clustering in d dimensions. *Theoret. Comput. Sci.*, 286:139–149, 2002.

- [KM⁺02] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. A local search approximation algorithm for k -means clustering. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 10–18, 2002.
- [Koh95] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.
- [LE97] D.P. Luebke and C. Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proc. ACM Conf. SIGGRAPH 97*, pages 199–208, 1997.
- [Lea92] V.F. Leavers. *Shape Detection in Computer Vision using the Hough Transform*. Springer-Verlag, Berlin, 1992.
- [Lee82] D.T. Lee. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. Mach. Intell.*, PAMI-4:363–369, 1982.
- [Liv93] C. Livingston. *Knot Theory*. Math. Assoc. Amer., Washington, 1993.
- [LW88] Y. Lamdan and H.J. Wolfson. Geometric hashing: a general and efficient model-based recognition scheme. In *2nd Inter. Conf. on Comput. Vision*, pages 238–249, 1988.
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Sympos. Math., Stat. and Prob.*, pages 281–296, 1967.
- [Mat00] J. Matoušek. On approximate geometric k -clustering. *Discrete Comput. Geom.*, 24:61–84, 2000.
- [McL92] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, New York, 1992.
- [MO88] A. Melkman and J. O'Rourke. On polygonal chain approximation. In G.T. Toussaint, editor, *Computational Morphology*, pages 87–95. North-Holland, Amsterdam, 1988.
- [MSW02] R. Nussinov M. Shatsky and H.J. Wolfson. Flexible protein alignment and hinge detection. *Proteins*, 48:242–256, 2002.
- [O'R98] J. O'Rourke. *Computational Geometry in C*, 2nd edition. Cambridge University Press, 1998.
- [PB97] N.R. Pal and J. Biswas. Cluster validation using graph theoretic concepts. *Pattern Recognition*, 30:847–857, 1997.
- [PD90] H. Plantinga and C.R. Dyer. Visibility, occlusion, and the aspect graph. *Internat. J. Comput. Vision*, 5:137–160, 1990.
- [PM99] D. Pelleg and A. Moore. Accelerating exact k -means algorithms with geometric reasoning. In *Knowledge Discovery and Data Mining*, pages 277–281, AAAI Press, New York, 1999.
- [Rei32] K. Reidemeister. *Knotentheorie*. Springer-Verlag, Berlin, 1932. L.F. Boron, C.O. Christenson and B.A. Smith (English translation), *Knot Theory*, BSC Associates, Moscow, Idaho, USA, 1983.
- [Rob93] J.-M. Robert. Maximum distance between two sets of points in E^d . *Pattern Recogn. Lett.*, 14, 1993.
- [RWLI75] G.L. Ritter, H.B. Woodruff, S.R. Lowry, and T.L. Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Trans. Inform. Theory*, 21:665–669, 1975.
- [Smi00] M. Smid. Closest point problems in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 877–935. Elsevier North-Holland, Amsterdam, 2000.
- [SNT⁺92] V. Srinivasan, L.R. Nackman, J.-M. Tang, and S.N. Meshkat. Automatic mesh generation using the symmetric axis transform of polygonal domains. *Proc. IEEE*, 80:1485–1501, 1992.

- [Sos99] M. Soss. On the size of the Euclidean sphere of influence graph. In *11th Canad. Conf. Comput. Geom.*, pages 43–46, 1999.
- [SPB95] E.C. Sherbrooke, N.M. Patrikalakis, and E. Brisson. Computation of the medial axis transform of 3-d polyhedra. In *Proc. 3rd Sympos. Solid Modeling and Appl.*, pages 187–199, 1995.
- [Sug84] K. Sugihara. An $n \log n$ algorithm for determining the congruity of polyhedra. *J. Comput. Syst. Sci.*, 29:36–47, 1984.
- [Sup83] K.J. Supowit. The relative neighborhood graph with an application to minimum spanning trees. *J. Assoc. Comput. Mach.*, 30:428–448, 1983.
- [TBP85] G.T. Toussaint, B.K. Bhattacharya, and R.S. Poulsen. The application of Voronoi diagrams to nonparametric decision rules. In *Computer Science and Statistics: The Interface*, pages 97–108, 1985.
- [TM82] G.T. Toussaint and M.A. McAlear. A simple $O(n \log n)$ algorithm for finding the maximum distance between two finite planar sets. *Pattern Recogn. Lett.*, 1:21–24, 1982.
- [Tou80a] G.T. Toussaint. Pattern recognition and geometrical complexity. In *Proc. 5th IEEE Internat. Conf. Pattern Recogn.*, pages 1324–1347, 1980.
- [Tou80b] G.T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recogn.*, 12:261–268, 1980.
- [Tou84] G.T. Toussaint. An optimal algorithm for computing the minimum vertex distance between two crossing convex polygons. *Computing*, 32:357–364, 1984.
- [Tou91] G.T. Toussaint. Computational geometry and computer vision. In B. Melter, A. Rosenfeld, and P. Bhattacharya, editors, *Vision Geometry*, pages 213–224. Amer. Math. Soc., Providence, 1991.
- [Tou00] G.T. Toussaint. The complexity of computing nice viewpoints of objects in space. In *Proc. Vision Geometry IX, SPIE Internat. Sympos. Optical Sci. Tech.*, pages 1–11, 2000.
- [Tou02] G.T. Toussaint. Proximity graphs for nearest neighbor decision rules: Recent progress. In *Interface-2002, Sympos. Comput. Statist. (Geoscience and Remote Sensing)*, Montreal, 2002.
- [Vai89] P.M. Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete Comput. Geom.*, 4:101–115, 1989.
- [Wil91] G. Wilfong. Nearest neighbor problems. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 224–233, 1991.
- [WM97] D. Randall Wilson and T.R. Martinez. Instance pruning techniques. In D. Fisher, editor, *Machine Learning: Proc. 14th Internat. Conf.*, pages 404–411. Morgan Kaufmann Publishers, San Francisco, 1997.
- [WM00] D. Randall Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38:257–286, 2000.
- [WWV85] J.D. Wolter, T. Woo, and R.A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. *Visual Comput.*, 1:37–48, 1985.
- [WY99] C.-A. Wang and B.-T. Yang. A tight bound for β -skeleton of minimum weight triangulations. In F. Dehne, A. Gupta, J.-R. Sack, and R. Tamassia, editors, *Algorithms Data Struct. 6th Internat. Workshop (WADS'99)*, volume 1663 of *Lecture Notes Comput. Sci.*, pages 265–275. Springer-Verlag, Berlin, 1999.

- [Yap90] C.K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990.
- [YR91] C. Yao and J.G. Rokne. A straightforward algorithm for computing the medial axis of a simple polygon. *Internat. J. Comput. Math.*, 39:51–60, 1991.
- [ZS02] H. Zhang and G. Sun. Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recogn.*, 35:1481–1490, 2002.

52 GRAPH DRAWING

Roberto Tamassia, Giuseppe Liotta

INTRODUCTION

Graph drawing addresses the problem of constructing geometric representations of graphs, and has important applications to key computer technologies such as software engineering, database systems, visual interfaces, and computer-aided design. Research on graph drawing has been conducted within several diverse areas, including discrete mathematics (topological graph theory, geometric graph theory, order theory), algorithmics (graph algorithms, data structures, computational geometry, VLSI), and human-computer interaction (visual languages, graphical user interfaces, software visualization). This chapter overviews aspects of graph drawing that are especially relevant to computational geometry. Basic definitions on drawings and their properties are given in Section 52.1. Bounds on geometric and topological properties of drawings (e.g., area and crossings) are presented in Section 52.2. Section 52.3 deals with the time complexity of fundamental graph drawing problems. An example of a drawing algorithm is given in Section 52.4. General techniques for drawing graphs are surveyed in Section 52.5. Section 52.6 covers selected topics that have recently attracted considerable research interest.

52.1 DRAWINGS AND THEIR PROPERTIES

TYPES OF GRAPHS

First, we define some terminology on graphs pertinent to graph drawing. Throughout this chapter let n and m be the number of graph vertices and edges respectively, and d the maximum vertex degree (i.e., number of incident edges).

GLOSSARY

Degree- k graph: Graph with maximum degree $d \leq k$.

Digraph: Directed graph, i.e., graph with directed edges (drawn as arrows).

Acyclic digraph: Without directed cycles.

Transitive edge: Edge (u, v) of a digraph is transitive if there is a directed path from u to v not containing edge (u, v) .

Reduced digraph: Without transitive edges.

Source: Vertex of a digraph without incoming edges.

Sink: Vertex of a digraph without outgoing edges.

st-digraph: Acyclic digraph with exactly one source and one sink, which are joined by an edge (also called **bipolar digraph**).

Connected graph: Any two vertices are joined by a path.

Biconnected graph: Any two vertices are joined by two vertex-disjoint paths.

Triconnected graph: Any two vertices are joined by three (pairwise) vertex-disjoint paths.

Tree: Connected graph without cycles.

Rooted tree: Directed tree with a distinguished vertex, the **root**, such that each vertex lies on a directed path to the root.

Binary tree: Rooted tree where each vertex has at most two incoming edges.

Layered (di)graph: The vertices are partitioned into sets, called layers. A rooted tree can be viewed as a layered digraph where the layers are sets of vertices at the same distance from the root.

k -layered (di)graph: Layered (di)graph with k layers.

TYPES OF DRAWINGS

In a drawing of a graph, vertices are represented by points (or by geometric figures such as circles or rectangles) and edges are represented by curves such that any two edges intersect at most in a finite number of points. Except for Section 52.6, which covers 3D drawings, we consider drawings in the plane.

GLOSSARY

Polyline drawing: Each edge is a polygonal chain ([Figure 52.1.1\(a\)](#)).

Straight-line drawing: Each edge is a straight-line segment ([Figure 52.1.1\(b\)](#)).

Orthogonal drawing: Each edge is a chain of horizontal and vertical segments ([Figure 52.1.1\(c\)](#)).

Bend: In a polyline drawing, point where two segments belonging to the same edge meet ([Figure 52.1.1\(a\)](#)).

Orthogonal Representation: Representation of orthogonal drawing in terms of bends along each edge and angles around each vertex.

Crossing: Point where two edges intersect ([Figure 52.1.1\(b\)](#)).

Grid drawing: Polyline drawing such that vertices, crossings, and bends have integer coordinates.

Planar drawing: No two edges cross (see [Figure 52.1.1\(d\)](#)).

Planar (di)graph: Admits a planar drawing.

Embedded (di)graph: Planar (di)graph with a prespecified topological embedding (i.e., set of faces), which must be preserved in the drawing.

Upward drawing: Drawing of a digraph where each edge is monotonically non-decreasing in the vertical direction (see [Figure 52.1.1\(d\)](#)).

Upward planar digraph: Admits an upward planar drawing.

Layered drawing: Drawing of a layered graph such that vertices in the same layer lie on the same horizontal line (also called **hierarchical drawing**).

Face: A region of the plane defined by a planar drawing, where the unbounded region is called the **external face**.

Convex drawing: Planar straight-line drawing such that the boundary of each face is a convex polygon.

Visibility drawing: Drawing of a graph based on a geometric visibility relation, e.g., the vertices might be drawn as horizontal segments, and the edges associated with vertically visible segments.

Proximity drawing: Drawing of a graph based on a geometric proximity relation, e.g., a tree is drawn as the Euclidean minimum spanning tree of a set of points.

Dominance drawing: Upward drawing of an acyclic digraph such that there exists a directed path from vertex u to vertex v if and only if $x(u) \leq x(v)$ and $y(u) \leq y(v)$, where $x(\cdot)$ and $y(\cdot)$ denote the coordinates of a vertex.

hv-drawing: Upward orthogonal straight-line drawing of a binary tree such that the drawings of the subtrees of each node are separated by a horizontal or vertical line.

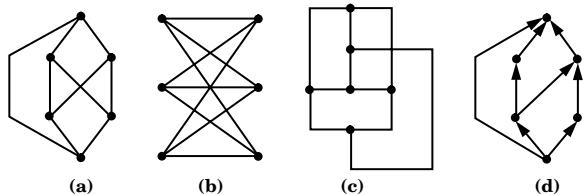


FIGURE 52.1.1

Types of drawings: (a) polyline drawing of $K_{3,3}$; (b) straight-line drawing of $K_{3,3}$; (c) orthogonal drawing of $K_{3,3}$; (d) planar upward drawing of an acyclic digraph.

Straight-line and orthogonal drawings are special cases of polyline drawings. Polyline drawings provide great flexibility since they can approximate drawings with curved edges. However, edges with more than two or three bends may be difficult to “follow” for the eye. Also, a system that supports editing of polyline drawings is more complicated than one limited to straight-line drawings. Hence, depending on the application, polyline or straight-line drawings may be preferred. If vertices are represented by points, orthogonal drawings exist only for graphs of maximum vertex degree 4.

PROPERTIES OF DRAWINGS

GLOSSARY

Crossings χ : Total number of crossings in a drawing.

Area: Area of the convex hull of the drawing.

Total edge length: Sum of the lengths of the edges.

Maximum edge length: Maximum length of an edge.

Total number of bends: Total number of bends on the edges of a polyline drawing.

Maximum number of bends: Maximum number of bends on an edge of a polyline drawing.

Angular resolution ρ : Smallest angle formed by two edges, or segments of edges, incident on the same vertex or bend, in a polyline drawing.

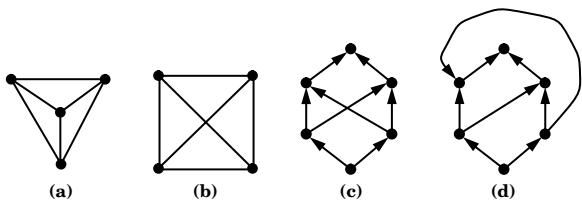
Aspect ratio: Ratio of the longest to the shortest side of the smallest rectangle with horizontal and vertical sides covering the drawing.

There are infinitely many drawings for a graph. In drawing a graph, we would like to take into account a variety of properties. For example, planarity and the display of symmetries are highly desirable in visualization applications. Also, it is customary to display trees and acyclic digraphs with upward drawings. In general, to avoid wasting valuable space on a page or a computer screen, it is important to keep the area of the drawing small. In this scenario, many graph drawing problems can be formalized as multi-objective optimization problems (e.g., construct a drawing with minimum area and minimum number of crossings), so that tradeoffs are inherent in solving them. Typically, it is desirable to maximize the angular resolution and to minimize the other measures.

The following examples illustrate two typical tradeoffs in graph drawing problems. Figure 52.1.2(a–b) shows two drawings of K_4 , the complete graph on four vertices. The drawing of part (a) is planar, while the drawing of part (b) “maximizes symmetries.” It can be shown that no drawing of K_4 is optimal with respect to both criteria, i.e., the maximum number of symmetries cannot be achieved by a planar drawing. Figure 52.1.2(c–d), shows two drawing of the same acyclic digraph G . The drawing of part (c) is upward, while the drawing of part (d) is planar. It can be shown that there is no drawing of G which is both planar and upward.

FIGURE 52.1.2

(a–b) Tradeoff between planarity and symmetry in drawing K_4 . (c–d) Tradeoff between planarity and upwardness in drawing an acyclic digraph G .



52.2 BOUNDS ON DRAWING PROPERTIES

For various classes of graphs and drawing types, many universal/existential upper and lower bounds for specific drawing properties have been discovered. Such bounds typically exhibit tradeoffs between drawing properties. A universal bound applies to all the graphs of a given class. An existential bound applies to infinitely many graphs of the class. Whenever we give bounds on the area or edge length, we assume that the drawing is constrained by some resolution rule that prevents it from being reduced by an arbitrary scaling (e.g., requiring a grid drawing, or stipulating a minimum unit distance between any two vertices).

BOUNDS ON THE AREA

Table 52.2.1 summarizes selected universal upper bounds and existential lower bounds on the area of drawings of graphs. In the table, a is an arbitrary constant $0 \leq a < 1$, and b and c are fixed constants $1 < b < c$, and ϵ for an arbitrary positive

constant. The abbreviations “PSL” and “PSLg” are used for “planar straight-line” “planar straight-line grid,” respectively. In general, the effect of bends on the area

TABLE 52.2.1 Universal upper and existential lower bounds on area.

	CLASS OF GRAPHS	DRAWING TYPE	AREA	
1	Rooted tree	upward PSLg	$\Omega(n)$	$O(n \log n)$
2	Rooted tree	strictly upward PSLg	$\Omega(n \log n)$	$O(n \log n)$
3	deg- $O(n^a)$ rooted tree	upward planar polyline grid	$\Omega(n)$	$O(n)$
4	Binary tree	upward planar orthog grid	$\Omega(n \log \log n)$	$O(n \log \log n)$
5	Binary tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n^{1+\epsilon})$
6	Fibonacci tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n)$
7	AVL tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n)$
8	Balanced tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n)$
9	Binary tree	PSLg	$\Omega(n)$	$O(n)$
10	Tree	PSLg	$\Omega(n)$	$O(n \log n)$
11	deg- $O(n^a)$ tree	planar polyline grid	$\Omega(n)$	$O(n)$
12	deg-4 tree	planar orthog grid	$\Omega(n)$	$O(n)$
13	Planar graph	planar polyline grid	$\Omega(n^2)$	$O(n^2)$
14	Planar graph	PSL	$\Omega(c^{\rho n})$	
15	Planar graph	PSLg	$\Omega(n^2)$	$O(n^2)$
16	Triconn planar graph	PSL convex grid	$\Omega(n^2)$	$O(n^2)$
17	Planar graph	planar orthog grid	$\Omega(n^2)$	$O(n^2)$
18	Planar degree-4 graph	orthog grid	$\Omega(n \log n)$	$O(n \log^2 n)$
19	Upward planar digraph	upward PSLg	$\Omega(b^n)$	$O(c^n)$
20	Reduced planar st-digraph	upward PSLg dominance	$\Omega(n^2)$	$O(n^2)$
21	Upward planar digraph	upward planar grid polyline	$\Omega(n^2)$	$O(n^2)$
22	General graph	polyline grid	$\Omega(n + \chi)$	$O((n + \chi)^2)$

requirement is dual. On the one hand, bends occupy space and hence negatively affect the area. On the other hand, bends may help in routing edges without using additional space. The following comments apply to Table 52.2.1, where specific rows of the table are indicated within parentheses. Linear or almost-linear bounds on the area can be achieved for trees (1–12). See Table 52.2.4 for tradeoffs between area and aspect ratio in drawings of trees. Planar graphs admit planar drawings with quadratic area (13–18). However, the area requirement of planar straight-line drawings may be exponential if high angular resolution is also desired (14). Almost linear area instead can be achieved in nonplanar drawings of planar graphs (18), which have applications to VLSI circuits. Upward planar drawings provide an interesting tradeoff between area and the total number of bends (19–21). Indeed, unless the digraph is reduced (20), the area can become exponential if a straight-line drawing is required (19). A quadratic area bound is achieved only at the expense of a linear number of bends (21).

BOUNDS ON THE ANGULAR RESOLUTION

Table 52.2.2 summarizes selected universal lower bounds and existential upper bounds on the angular resolution of drawings of graphs. Here c is a fixed constant with $c > 1$.

 TABLE 52.2.2 Universal lower and existential upper bounds on angular resolution.

CLASS OF GRAPHS	DRAWING TYPE	ANGULAR RESOLUTION	
General graph	straight-line	$\Omega(\frac{1}{d^2})$	$O(\frac{\log d}{d^2})$
Planar graph	straight-line	$\Omega(\frac{1}{d})$	$O(\frac{1}{d})$
Planar graph	planar straight-line	$\Omega(\frac{1}{cd})$	$O(\sqrt{\frac{\log d}{d^3}})$

BOUNDS ON THE NUMBER OF BENDS

Table 52.2.3 summarizes selected universal upper bounds and existential lower bounds on the total and maximum number of bends in orthogonal drawings. Some bounds are stated for $n \geq 5$ or $n \geq 7$ because the maximum number of bends is at least 2 for K_4 and at least 3 for the skeleton graph of an octahedron, in any planar orthogonal drawing.

 TABLE 52.2.3 Orthogonal drawings: universal upper and existential lower bounds on the number of bends. Notes: $\dagger n \geq 7$; $\ddagger n \geq 5$.

CLASS OF GRAPHS	DRAWING TYPE	TOTAL # BENDS		MAX # BENDS	
		$\geq n$	$\leq 2n + 2$	≥ 2	≤ 2
deg-4 \dagger	orthog	$\geq n$	$\leq 2n + 2$	≥ 2	≤ 2
Planar deg-4 \dagger	orthog planar	$\geq 2n - 2$	$\leq 2n + 2$	≥ 2	≤ 2
Embedded deg-4	orthog planar	$\geq 2n - 2$	$\leq \frac{12}{5}n + 2$	≥ 3	≤ 3
Biconn embedded deg-4	orthog planar	$\geq 2n - 2$	$\leq 2n + 2$	≥ 3	≤ 3
Triconn embedded deg-4	orthog planar	$\geq \frac{4}{3}(n - 1) + 2$	$\leq \frac{3}{2}n + 4$	≥ 2	≤ 2
Embedded deg-3 \ddagger	orthog planar	$\geq \frac{1}{2}n + 1$	$\leq \frac{1}{2}n + 1$	≥ 1	≤ 1

TRADEOFF BETWEEN AREA AND ASPECT RATIO

The ability to construct area-efficient drawings is essential in practical visualization applications, where screen space is at a premium. However, achieving small area is not enough, e.g., a drawing with high aspect ratio may not be conveniently placed on a workstation screen, even if it has modest area. Hence, it is important to keep the aspect ratio small. Ideally, one would like to obtain small area for any given aspect ratio in a wide range. This would provide graphical user interfaces with the flexibility of fitting drawings into arbitrarily shaped windows. A variety of tradeoffs for the area and aspect ratio arise even when drawing graphs with a simple structure, such as trees. Table 52.2.4 summarizes selected universal bounds that can be simultaneously achieved on the area and the aspect ratio of various types of drawings of trees. In the table, a is an arbitrary constant with $0 \leq a < 1$ and the abbreviation “PSLog” is used for “planar straight-line orthogonal grid,” that is, a PLG with edges either horizontal or vertical segments. Only for a few

cases there exist algorithms that guarantee efficient area performance and that can accept any user-specified aspect ratio in a given range. For such cases the aspect ratio in Table 52.2.4 is given as an interval.

TABLE 52.2.4 Trees: Universal upper bounds simultaneously achievable for area and aspect ratio.

CLASS OF GRAPHS	DRAWING TYPE	AREA	ASPECT RATIO
Rooted tree	upward PSL layered grid	$O(n^2)$	$O(1)$
Rooted tree	upward PSLg	$O(n \log \log n)$	$O(n \log \log n / \log^2 n)$
Rooted deg- $O(1)$ tree	upward planar polyline grid	$O(n)$	$O(n^a)$
Binary tree	upward planar orthog grid	$O(n \log \log n)$	$O(n \log \log n / \log^2 n)$
Binary tree	PSLg	$O(n)$	$[O(1), O(n^a)]$
Binary tree	PSLog	$O(n \log \log n)$	$[O(1), O(n \log \log n / \log^2 n)]$
Binary tree	upward PSLog	$O(n \log n)$	$O(1)$
deg-4 tree	orthog grid	$O(n)$	$O(1)$
deg-4 tree	orthog grid, leaves on hull	$O(n \log n)$	$O(1)$

While upward planar straight-line drawings are the most natural way of visualizing rooted trees, the existing drawing techniques are unsatisfactory with respect to either the area requirement or the aspect ratio. The situation is similar for orthogonal drawings. Regarding polyline drawings, linear area can be achieved with a prescribed aspect ratio. However, experiments show that this is done at the expense of a somehow aesthetically unappealing drawing. For nonupward drawings of trees, linear area and optimal aspect ratio are possible for planar orthogonal drawings, and a small (logarithmic) amount of extra area is needed if the leaves are constrained to be on the convex hull of the drawing (e.g., pins on the boundary of a VLSI circuit). However, the nonupward drawing methods do not seem to yield aesthetically pleasing drawings, and are suited more for VLSI layout than for visualization applications.

TRADEOFF BETWEEN AREA AND ANGULAR RESOLUTION

Table 52.2.5 summarizes selected universal bounds that can be simultaneously achieved on the area and the angular resolution of drawings of graphs. Here b and c are fixed constants, $b > 1$ and $c > 1$. Universal lower bounds on the angular resolution exist that depend only on the degree of the graph. Also, substantially better bounds can be achieved by drawing a planar graph with bends or in a non-planar way.

TABLE 52.2.5 Universal upper bounds for area and lower bounds for angular resolution, simultaneously achievable.

CLASS OF GRAPHS	DRAWING TYPE	AREA	ANGULAR RESOLUTION
Planar graph	straight-line	$O(d^6 n)$	$\Omega(\frac{1}{d^2})$
Planar graph	straight-line	$O(d^3 n)$	$\Omega(\frac{1}{d})$
Planar graph	planar straight-line grid	$O(n^2)$	$\Omega(\frac{1}{n^2})$
Planar graph	planar straight-line	$O(b^n)$	$\Omega(\frac{1}{c^d})$
Planar graph	planar polyline grid	$O(n^2)$	$\Omega(\frac{1}{d})$

OPEN PROBLEMS

1. Determine the area requirement of (upward) planar straight-line drawings of trees. There is currently an $O(\log n)$ gap between the known upper and lower bounds ([Table 52.2.1](#)).
2. Determine the area requirement of strictly upward planar order preserving straight-line drawings of binary trees ([Table 52.2.1](#)).
3. Determine the area requirement of orthogonal (or, more generally, polyline) nonplanar drawings of planar graphs. There is currently an $O(\log n)$ gap between the known upper and lower bounds ([Table 52.2.1](#)).
4. Close the wide gap between the $\Omega(\frac{1}{d^2})$ universal lower bound and the $O(\frac{\log d}{d^2})$ existential upper bound on the angular resolution of straight-line drawings of general graphs ([Table 52.2.2](#)).
5. Close the gap between the $\Omega(\frac{1}{c^d})$ universal lower bound and the $O(\sqrt{\frac{\log d}{d^3}})$ existential upper bound on the angular resolution of planar straight-line drawings of planar graphs ([Table 52.2.2](#)).
6. Determine the best possible aspect ratio and area that can be simultaneously achieved for (upward) planar straight-line and orthogonal drawings of trees ([Table 52.2.4](#)).

52.3 COMPLEXITY OF GRAPH DRAWING PROBLEMS

[Tables 52.3.1–52.3.3](#) summarize selected results on the time complexity of some fundamental graph drawing problems.

It is interesting that apparently similar problems exhibit very different time complexities. For example, while planarity testing can be done in linear time, upward planarity testing is NP-hard. Note that, as illustrated in [Figure 52.1.2\(c–d\)](#), planarity and acyclicity are necessary but not sufficient conditions for upward planarity. While many efficient algorithms exist for constructing drawings of trees and planar graphs with good universal area bounds, exact area minimization for most types of drawings is NP-hard, even for trees.

TABLE 52.3.1 Time complexity of some fundamental graph drawing problems:
general graphs and digraphs.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
General graph 2-layered graph	minimize crossings minimize crossings in layered drawing with preassigned order on one layer	NP-hard NP-hard	
General graph	maximum planar subgraph	NP-hard	
General graph	planarity testing and computing a planar embedding	$\Omega(n)$	$O(n)$
General graph	maximal planar subgraph	$\Omega(n+m)$	$O(n+m)$
General digraph	upward planarity testing	NP-hard	
Embedded digraph	upward planarity testing	$\Omega(n)$	$O(n^2)$
Single-source digraph	upward planarity testing	$\Omega(n)$	$O(n)$
General graph	draw as the intersection graph of a set of unit diameter disks in the plane	NP-hard	

OPEN PROBLEMS

1. Reduce the time complexity of upward planarity testing for embedded digraphs (currently $O(n^2)$), or prove a superlinear lower bound (Table 52.3.1).
2. Reduce the time complexity of bend minimization for planar orthogonal drawings of embedded graphs (currently $O(n^{7/4} \log n)$), or prove a superlinear lower bound ([Table 52.3.2](#)).
3. Reduce the time complexity of bend minimization for planar orthogonal drawings of degree-3 graphs (Table 52.3.2).

52.4 EXAMPLE OF A GRAPH DRAWING ALGORITHM

In this section we outline the algorithm in [Tam87] for computing, for an embedded degree-4 graph G , a planar orthogonal grid drawing with minimum number of bends and using $O(n^2)$ area (see Table 52.3.2). This algorithm is the core of a practical drawing algorithm for general graphs (see Section 52.5 and [Figure 52.4.1\(d\)](#)). The algorithm consists of two main phases:

1. Computation of an orthogonal representation for G , where only the bends and the angles of the orthogonal drawing are defined.
2. Assignment of integer lengths to the segments of the orthogonal representation.

Phase 1 uses a transformation into a network flow problem (Figure 52.4.1(a–c)), where each unit of flow is associated with a right angle in the orthogonal drawing. Hence, angles are viewed as a commodity that is produced by the vertices, transported across faces by the edges through their bends, and eventually consumed by

TABLE 52.3.2 Time complexity of some fundamental graph drawing problems: Planar graphs and digraphs.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
Planar graph	planar straight-line drawing with prescribed edge lengths	NP-hard	
Planar graph	planar straight-line drawing with maximum angular resolution	NP-hard	
Embedded graph	test the existence of a planar straight-line drawing with prescribed angles between pairs of consecutive edges incident on a vertex	NP-hard	
Maximal planar graph	test the existence of a planar straight-line drawing with prescribed angles between pairs of consecutive edges incident on a vertex	$\Omega(n)$	$O(n)$
Planar graph	planar straight-line grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$\Omega(n)$	$O(n)$
Planar graph	planar polyline drawing with $O(n^2)$ area, $O(n)$ bends, and $O(1/d)$ angular resolutions	$\Omega(n)$	$O(n)$
Triconn planar graph	planar straight-line convex grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$\Omega(n)$	$O(n)$
Triconn planar graph	planar straight-line strictly convex drawing	$\Omega(n)$	$O(n)$
Reduced planar st -digraph	upward planar grid straight-line dominance drawing with minimum area	$\Omega(n)$	$O(n)$
Upward planar digraph	upward planar polyline grid drawing with $O(n^2)$ area and $O(n)$ bends	$\Omega(n)$	$O(n)$
Planar deg-4 graph	planar orthogonal grid drawing with minimum number of bends	NP-hard	
Planar deg-3 graph	planar orthogonal grid drawing with minimum number of bends and $O(n^2)$ area	$\Omega(n)$	$O(n^5 \log n)$
Embedded deg-4 graph	planar orthogonal grid drawing with minimum number of bends and $O(n^2)$ area	$\Omega(n)$	$O(n^{7/4} \log n)$
Planar deg-4 graph	planar orthogonal grid drawing with $O(n^2)$ area and $O(n)$ bends	$\Omega(n)$	$O(n)$
Planar orthog rep	planar orthogonal grid drawing with minimum area	NP-hard	

the faces. From the embedded graph G we construct a flow network N as follows. The nodes of network N are the vertices and faces of G . Let $\deg(f)$ denote the number of edges of the circuit bounding face f . Each vertex v supplies $\sigma(v) = 4$ units of flow, and each face f consumes $\tau(f)$ units of flow, where

$$\tau(f) = \begin{cases} 2\deg(f) - 4 & \text{if } f \text{ is an internal face} \\ 2\deg(f) + 4 & \text{if } f \text{ is the external face} \end{cases} .$$

By Euler's formula, $\sum_v \sigma(v) = \sum_f \tau(f)$, i.e., the total supply is equal to the total consumption.

 TABLE 52.3.3 Time complexity of some fundamental graph drawing problems: trees.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
Tree	draw as the Euclidean minimum spanning tree of a set of points in the plane	NP-hard	
degree-4 tree	minimize area in planar orthogonal grid drawing	NP-hard	
degree-4 tree	minimize total/maximum edge length in planar orthogonal grid drawing	NP-hard	
Rooted tree	minimize area in a planar straight-line upward layered grid drawing that displays symmetries and isomorphisms of subtrees	NP-hard	
Rooted tree	minimize area in a planar straight-line upward layered drawing that displays symmetries and isomorphisms of subtrees	$\Omega(n)$	$O(n^k)$, $k \geq 1$
Binary tree	minimize area in hv-drawing	$\Omega(n)$	$O(n\sqrt{n \log n})$
Rooted tree	planar straight-line upward layered grid drawing with $O(n^2)$ area	$\Omega(n)$	$O(n)$
Rooted tree	planar polyline upward grid drawing with $O(n)$ area	$\Omega(n)$	$O(n)$

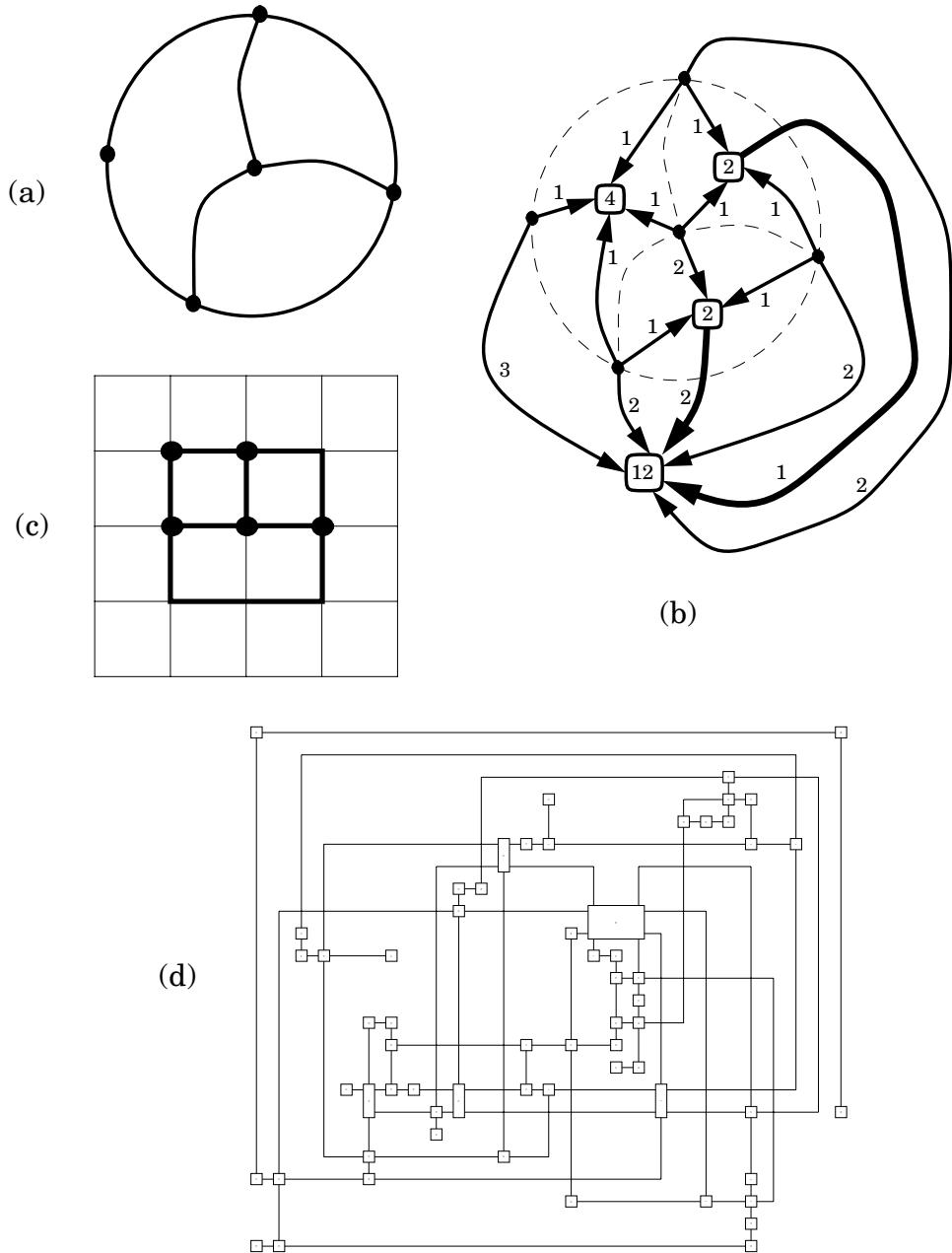
Network N has two types of arcs:

- arcs of the type (v, f) , where f is a face incident on vertex v ; the flow in (v, f) represents the angle at vertex v in face f , and has lower bound 1, upper bound 4, and cost 0;
- arcs of the type (f, g) , where face f shares an edge e with face g ; the flow in (f, g) represents the number of bends along edge e with the right angle inside face f , and has lower bound 0, upper bound $+\infty$, and cost 1.

The conservation of flow at the vertices expresses the fact that the sum of the angles around a vertex is equal to 2π . The conservation of flow at the faces expresses the fact that the sum of the angles at the vertices and bends of an internal face is equal to $\pi(p - 2)$, where p is the number of such angles. For the external face, the above sum is equal to $\pi(p + 2)$. It can be shown that every feasible flow ϕ in network N corresponds to an admissible orthogonal representation for graph G , whose number of bends is equal to the cost of flow ϕ . Hence, an orthogonal representation for G with the minimum number of bends can be computed from a minimum-cost flow in G . This flow can be constructed in $O(n^2 \log n)$ time with standard flow-augmentation methods. Phase 2 uses a simple compaction strategy derived from VLSI layout, where the lengths of the horizontal and vertical segments are computed independently after a preliminary refinement of the orthogonal representation that decomposes each face into rectangles. The resulting drawing is shown in [Figure 52.4.1\(d\)](#).

FIGURE 52.4.1

(a) Embedded graph G . (b) Minimum cost flow in network N : the flow is shown next to each arc; arcs with zero flow are omitted; arcs with unit cost are drawn with thick lines; a face f is represented by a box labeled with $\tau(f)$. (c) Planar orthogonal grid drawing of G with minimum number of bends. (d) Orthogonal grid drawing of a nonplanar graph produced by a drawing method for general graphs based on the algorithm of Section 52.4.



52.5 TECHNIQUES FOR DRAWING GRAPHS

In this section we outline some of the most successful techniques that have been devised for drawing general graphs.

PLANARIZATION

The planarization approach is motivated by the availability of many efficient and well-analyzed drawing algorithms for planar graphs (see [Table 52.3.2](#)). If the graph is nonplanar, it is transformed into a planar graph by means of a preliminary planarization step that replaces each crossing with a fictitious vertex. Finding the minimum number of crossings or a maximum planar subgraph are NP-hard problems. Hence, existing planarization algorithms use heuristics. The best available heuristic for the maximum planar subgraph problem is described in [JM96]. This method has a solid theoretical foundation in polyhedral combinatorics, and achieves good results in practice. A successful drawing algorithm based on the planarization approach and a bend-minimization method [Tam87] is described in [TDB88] ([Figure 52.4.1\(d\)](#) was generated by this algorithm). It has been widely used in software visualization systems.

LAYERING

The layering approach for constructing polyline drawings of directed graphs transforms the digraph into a layered digraph and then constructs a layered drawing. A typical algorithm based on the layering approach consists of the following main steps:

1. Assign each vertex to a layer, with the goal of maximizing the number of edges oriented upward.
2. Insert fictitious vertices along the edges that cross layers, so that each edge in the resulting digraph connects vertices in consecutive layers. (The fictitious vertices will be displayed as bends in the final drawing.)
3. Permute the vertices on each layer with the goal of minimizing crossings.
4. Adjust the positions of the vertices in each layer with the goal of distributing the vertices uniformly and minimizing the number of bends.

Most of the subproblems involved in the various steps are NP-hard, hence heuristics must be used. The layering approach was pioneered by Sugiyama et al. [STT81]. The most notable developments of this technique are due to Gansner et al. [GNV88, GKNV93]. For a survey on heuristics for the layering approach see also the paper by Jünger and Mutzel [JM97].

PHYSICAL SIMULATION

This approach uses a physical model where the vertices and edges of the graph are viewed as objects subject to various forces. Starting from an initial random configuration, the physical system evolves into a final configuration of minimum

energy, which yields the drawing. Rather than solving a system of differential equations, the evolution of the system is usually simulated using numerical methods (e.g., at each step, the forces are computed and corresponding incremental displacements of the vertices are performed). Drawing algorithms based on the physical simulation approach are often able to detect and display symmetries in the graph. However, their running time is typically high. The physical simulation approach was pioneered in [Ead84, KS80]. Sophisticated developments and applications include [DH96, EH00, FR91, GGK01, LMP01]. Related topics include declarative methods for graph drawing and approaches to graph drawing based on graph grammars; see, e.g. [CG95, LE95, Bra95].

52.6 RECENT RESEARCH TRENDS

In this section, we present an overview of selected areas of graph drawing that have recently attracted increasing attention.

THREE-DIMENSIONAL STRAIGHT-LINE DRAWINGS

The increasing demand of visualization algorithms to draw and browse very large networks makes it natural to investigate how much benefit can be obtained from the third dimension to represent the overall structure of a huge graph in a small portion of a virtual 3D environment. While the problem of computing small-sized crossing-free straight-line drawings in the plane has a long tradition, its 3D counterpart has become the subject of much attention only in recent years.

Chrobak, Goodrich, and Tamassia [CGT96] gave an algorithm for constructing 3D convex drawings of triconnected planar graphs with $O(n)$ volume and non-integer coordinates. Cohen, Eades, Lin, and Ruskey [CELR97] showed that every graph admits a straight-line crossing-free 3D drawing on an integer grid of $O(n^3)$ volume, and proved that this is asymptotically optimal. Calamoneri and Sterbini [CS97] showed that all 2-, 3-, and 4-colorable graphs can be drawn in a 3D grid of $O(n^2)$ volume with $O(n)$ aspect ratio and proved a lower bound of $\Omega(n^{1.5})$ on the volume of such graphs. For r -colorable graphs, Pach, Thiele and Tóth [PTT97] showed a bound of $\theta(n^2)$ on the volume. Garg, Tamassia, and Vocca [GTV96] showed that all 4-colorable graphs (and hence all planar graphs) can be drawn in $O(n^{1.5})$ volume and with $O(1)$ aspect ratio by using a grid model without restriction to integer vertex-coordinates.

Felsner, Liotta, and Wismath [FGW01] showed that all outerplanar graphs can be drawn in a restricted integer 3D grid of linear volume consisting of three parallel lines at distance 1 from each other. Dujmović, Morin, and Wood [DMW02] present $O(n \log^2 n)$ volume drawings of graphs with bounded tree-width and $O(n)$ volume for graphs with bounded path-width. Wood [Woo02] shows that also graphs with bounded queue number have 3D straight-line grid drawings of $O(n)$ volume. A result by Dujmović and Wood [DW03a] shows that linear volume can also be achieved for graphs with bounded tree-width; they show 3D straight-line grid drawings of volume $c \times n$ for these graphs, where c is a constant whose value exponentially depends on the tree-width. Di Giacomo, Liotta, and Wismath [DLW02a, DLW02b] show $4 \times n$ and $32 \times n$ volume for two subclasses of series-parallel graphs. Very

recently Dujmović and Wood [DWo03b] proved that several families of graphs, including planar graphs, admit a 3D straight-line grid drawing of $O(n^{1.5})$ volume.

PROXIMITY DRAWINGS

Recently much attention has been devoted to the study of the combinatorial properties of different types of proximity graphs. In a proximity graph two points are connected by an edge if and only if they are deemed *close* by some proximity measure. It is the measure that determines the type of graphs that result. Examples of proximity graphs include well-known geometric graphs such as minimum spanning trees, Gabriel graphs, minimum weight triangulations, rectangle of influence graphs, visibility graphs, and Delaunay diagrams. See [Section 51.2](#).

Proximity graphs can be regarded as straight-line drawings that satisfy some additional geometric constraints. Thus the problem of analyzing the combinatorial properties of a given type of proximity graph naturally raises the question of the characterization of those graphs admitting the given type of drawing. This, in turn, leads to the investigation of the design of efficient algorithms for computing such a drawing when one exists. These questions are far from being resolved in general, and only partial answers have appeared in the literature so far (see, e.g., [Dil90, LL96, LL02, LLMW98, LM03, LS93, WCY00]). One example is provided by a *minimum-weight drawing* of a planar triangulated graph G : a straight-line drawing Γ of G with the additional property that Γ is a minimum-weight triangulation of the points representing the vertices. If a graph admits a minimum weight drawing, it is called *minimum-weight drawable*; otherwise it is called *minimum-weight forbidden*. Little is known about the problem of constructing a minimum-weight drawing of a planar triangulation. Moreover, it is still not known whether computing a minimum-weight triangulation of a set of points in the plane is an NP-hard problem (see Garey and Johnson [GJ79]).

In [LL96] Lenhart and Liotta show that all maximal outerplanar triangulations are minimum-weight drawable, and gave a linear time (real RAM) algorithm for constructing such a drawing. This naturally leads to investigation of the internal structure of minimum-weight drawable triangulations. In [LL02] Lenhart and Liotta examine the *endoskeleton*—or *skeleton*, for short—of a triangulation: that is, the subgraph induced by the internal vertices of the triangulation. They construct skeletons that cannot appear in any minimum weight drawable triangulation; skeletons that do appear in minimum weight drawable triangulations; and skeletons that guarantee minimum weight drawability. Wang, Chin, and Yang [WCY00] also focus on the minimum weight drawability of triangulations with acyclic skeletons and show examples of triangulations of this type that do not admit a minimum weight drawing.

GRAPH DRAWING CHECKERS

The intrinsic structural complexity of the implementation of geometric algorithms makes the problem of formally proving the correctness of code infeasible in most cases. This has motivated research on *checkers*. A checker is an algorithm that receives as input a geometric structure and a predicate stating a property that should hold for the structure. The task of the checker is to verify whether the

structure satisfies or not the given property. Here, the expectation is that it is often easier to evaluate the quality of the output than the correctness of the software that produces it. Different papers (see, e.g., [DLPT98, MSNS⁺99]) have agreed on the basic features that a “good” checker should have:

Correctness: The checker should be correct beyond any reasonable doubt. (Otherwise, one is faced with checking the checker.)

Simplicity: The implementation should be straightforward.

Efficiency: The checker should not be less efficient than the algorithm producing the geometric structure.

Robustness: The checker should be able to handle degenerate input configurations and should not be affected by errors in flow control due to round-off approximations.

Checking is especially relevant in the graph drawing context. Indeed, graph drawing algorithms are among the most sophisticated of the entire computational geometry field, and their goal is to construct complex geometric structures with specific properties. Also, because of their immediate impact on application areas, graph drawing algorithms are usually implemented soon after invention. Further, such implementations are often available on the Web without any certification of their correctness. Of course, the checking problem becomes crucial when the drawing algorithm deals with very large data sets, when a simple complete visual inspection of the drawing is difficult. Devising graph drawing checkers involves answering only apparently innocent questions such as: “Is this drawing planar?” or “Is this drawing upward?” or “Are the faces convex?” The problem of checking the planarity of a subdivision has been independently studied by Mehlhorn et al. [MSNS⁺99] and by Devillers et al. [DLPT98]. In these papers linear-time algorithms are given to check the planarity of a subdivision composed by convex faces. Di Battista and Liotta [DL98] check the upward planarity of straight-line oriented drawings that may also have nonconvex faces.

INCREMENTAL GRAPH DRAWING

In several applications, such as software engineering and database design, users interact extensively with a displayed graph, continuously adding or deleting vertices and edges. Under such a scenario, a graph drawing system should update the drawing each time the displayed graph is modified by the user. Unfortunately, traditional drawing algorithms may not be suitable in these situations. Since they typically construct a drawing from scratch, they may fail to update the drawing quickly after the user modifies the displayed graph. Also, the new drawing constructed after the modification may be significantly different from the previous one, even if only a small change has been made in the displayed graph. In this case, the user’s *mental map* [ELMS95], that is, the mental image the user has of the graph, is not preserved, and a considerable cognitive effort is required to correlate the new drawing and the previous one. Bridgeman and Tamassia [BT00a] formulate and validate several difference metrics that can be used to measure how much a drawing algorithm changes the user’s mental map in an interactive environment. Tradeoffs between running time, optimization of the drawing properties, and preservation

of the mental map are typical issues to be addressed in incremental graph drawing. Several papers dealing with these issues have recently appeared in journal and conference proceedings. A limited list includes the work by Cohen, Di Battista, Tamassia, and Tollis [CDTT95] on data structures for dynamic graph drawing; the Bayesian framework of Brandes and Wagner [BW97]; the definition and experimental investigation of four scenarios for interactive orthogonal graph drawing by Papakostas and Tollis [PT98] and Papakostas, Six, and Tollis [PST97]; the papers on interactive orthogonal graph drawing by Biedl and Kaufmann [BK97], and by Brandes and Wagner [BW98]; the fully dynamic algorithms for orthogonal drawings in 3D-space by Closson, Gartshore, Johansen, and Wismath [CGJW00]; and the work by North and Woodhull [NW02] on on-line hierarchical graph drawing.

EXPERIMENTATION

Many graph drawing algorithms have been implemented and used in practical applications. Most papers show sample outputs, and some also provide limited experimental results on small test suites. However, in order to evaluate the practical performance of a graph drawing algorithm in visualization applications, it is essential to perform extensive experimentations with input graphs derived from the application domain and over a large set of aesthetic requirements that are desirable for the user to have in the drawing. Among papers that test the human perception of the aesthetic properties of graph drawing we mention the work by Purchase, Allder, and Carrington [PAC02] and by Bridgeman and Tamassia in an incremental setting [BT00b]. The first broad-view experimental study on graph drawing algorithms is due to Himsolt [Him95] presents a comparative study of twelve graph drawings algorithms based on various approaches. Di Battista et al. [DGL⁺97] report on an extensive experimental study comparing four orthogonal drawing algorithms based on the planarization approach. The test data are 11,582 graphs, ranging from 10 to 100 vertices, which are generated from a core set of 112 graphs used in “real-life” software engineering and database applications. A similar experimental setting is then used to analyze the performance of four graph drawing algorithms for directed acyclic graphs in [DGL⁺00]. Heuristics for computing orthogonal drawings with good area performance are experimentally validated in the works by Klau, Klein, and Mutzel [KKM01] and by Di Battista et al. [BDD⁺00]. Experimentation of graph drawing techniques for computing graphical representations of database schemas are conducted by Di Battista, Didimo, Patrignani, and Pizzonia [DDPP02]. An extensive experimental comparison of five algorithms based on force-directed and randomized methods is described in the work by Brandenburg, Himsolt, and Roher [BHR96]. Jünger and Mutzel [JM97] experimentally compare the performance of eight heuristics for straight-line drawings of 2-layer graphs. An extensive survey on experimental studies on graph drawing can be found in [VBL⁺00].

FIXED PARAMETER TRACTABILITY

Recently, the theory of parametrized complexity [DF97] has been applied with success to some computationally difficult graph drawing problems. A problem Π specified in terms of one or more parameters is *fixed-parameter tractable*, or in the

FPT class, if there is an algorithm that solves Π in $O(f(k) \cdot n^c)$ time, where n is the input size, k is the parameter size, c is a constant, and f is an arbitrary function dependent only on parameter k . For example it is NP-complete to decide, given a graph G and a positive integer k , whether G can be drawn on the plane with at most k edge crossings (edges are drawn as simple curves). However it has been shown by Grohe [Gro01] that this problem is fixed-parameter tractable since there exists a quadratic time algorithm that solves it for any fixed value of k . Other relevant NP-hard graph drawing problems have been proved to be in the FPT class [DFH⁺01b, DFH⁺01a, DW03]; some of these results are summarized in Table 52.6.1. In the table, h and k denote integer positive constants. It must be noted, however, that the constants hidden in the time complexities shown in Table 52.6.1 may depend heavily on the values of the parameter. For example, the crossing minimization problem for general graph has time complexity $O(f(k) \cdot n^2)$ where $f(k)$ is a doubly exponential function [Gro01]. Thus, it is equally important to find time-complexity bounds that can be of practical use for fixed-parameter tractable problems.

TABLE 52.6.1 Some NP-hard graph drawing problems that are fixed-parameter tractable.

GRAPH CLASS	NP-HARD PROBLEM	TIME COMPLEXITY
2-layered graph	2-layers planarization: remove at most k edges so biplanar	$O(f(k) + G)$
2-layered graph	2-layers crossing minimization: compute straight-line drawing on two layers with at most k crossings	$O(f(k) \cdot n^2)$
general graph	h -layers planarization: remove at most k edges so h -level planar	$O(f(h, k) \cdot n)$
general graph	h -layers crossing minimization: compute a straight-line drawing on h layers with at most k crossings	$O(f(h, k) \cdot n^2)$
general graph	crossing minimization: compute a straight-line drawing with at most k crossings	$O(f(k) \cdot n^2)$

52.7 SOURCES AND RELATED MATERIAL

Three books devoted to graph drawing have been published [DETT99, KE01, Sug02]. The proceedings of the annual Symposium on *Graph Drawing* are published by Springer-Verlag in the *Lecture Notes in Computer Science* series (volumes 2265, 1984, 1731, 1547, 1353, 1190, 1027, 894). Surveys on various aspects of graph drawing appear in [DLL95, DPS02, GT95, HMM00, JM97, Riv93, San99, SSV95, Tam90a, Tam90b, Tam99, VBL⁺00]. Special issues devoted to graph drawing have appeared in *Algorithmica* (vol. 16, no. 1, 1996), *Computational Geometry: Theory and Applications* (vol. 9, no. 1–2, 1998), the *Journal of Visual Languages and*

Computing (vol. 6, 1995), and the *Journal of Graph Algorithms and Applications* (vol. 3, no. 4, 1999; vol. 4, no. 3, 2000; vol. 6, no. 1, 2002; vol. 6, no. 3, 2002).

Sites with pointers to graph drawing resources and tools include the Web page maintained by Tamassia (<http://www.cs.brown.edu/people/rt/gd.html>) and the Web page maintained by Brandes (<http://graphdrawing.org/>).

RELATED CHAPTERS

[Chapter 10: Geometric graph theory](#)

[Chapter 25: Triangulations and mesh generation](#)

[Chapter 41: Robust geometric computation](#)

[Chapter 51: Pattern recognition](#)

REFERENCES

- [BDD⁺00] S.S. Bridgeman, G. Di Battista, W. Didimo, G. Liotta, R. Tamassia, and L. Vismara. Turn-regularity and optimal area drawings of orthogonal representations. *Comput. Geom. Theory Appl.*, 16:53–93, 2000.
- [BHR96] F.J. Brandenburg, M. Himsolt, and C. Roher. An experimental comparison of force-directed and randomized graph drawing algorithms. In *Proc. Graph Drawing 95*, volume 1027 of *Lecture Notes Comput. Sci.*, pages 76–87. Springer-Verlag, Berlin, 1996.
- [BK97] T.C. Biedl and M. Kaufmann. Area-efficient static and incremental graph drawings. In *Proc. European Sympos. Algorithms*, volume 1284 of *Lecture Notes Comput. Sci.*, pages 37–52. Springer-Verlag, Berlin, 1997.
- [Bra95] F.J. Brandenburg. Designing graph drawings by layout graph grammars. In R. Tamassia and I.G. Tollis, editors, *Proc. Graph Drawing 94*, volume 894 of *Lecture Notes Comput. Sci.*, pages 416–427. Springer-Verlag, Berlin, 1995.
- [BT00a] S.S. Bridgeman and R. Tamassia. Difference metrics for interactive orthogonal graph drawing algorithms. *J. Graph Algorithms Appl.*, 4:47–74, 2000.
- [BT00b] S.S. Bridgeman and R. Tamassia. A user study in similarity measures for graph drawing. In J. Marks, editor, *Proc. Graph Drawing 00*, volume 1984 of *Lecture Notes Comput. Sci.*, pages 19–30. Springer-Verlag, Berlin, 2000.
- [BW97] U. Brandes and D. Wagner. A Bayesian paradigm for dynamic graph layout. In G. Di Battista, editor, *Proc. Graph Drawing 97*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 236–247. Springer-Verlag, Berlin, 1997.
- [BW98] U. Brandes and D. Wagner. Dynamic grid embedding with few bends and changes. In *Proc. Annu. Internat. Sympos. Algorithms Comput.*, volume 1533 of *Lecture Notes Comput. Sci.*, pages 89–98. Springer-Verlag, Berlin, 1998.
- [CDTT95] R.F. Cohen, G. Di Battista, R. Tamassia, and I.G. Tollis. Dynamic graph drawings: Trees, series-parallel digraphs, and planar *st*-digraphs. *SIAM J. Comput.*, 24:970–1001, 1995.
- [CELR97] R.F. Cohen, P. Eades, T. Lin, and F. Ruskey. Three-dimensional graph drawing. *Algorithmica*, 17:199–208, 1997.
- [CG95] I.F. Cruz and A. Garg. Drawing graphs by example efficiently: Trees and planar acyclic digraphs. In R. Tamassia and I.G. Tollis, editors, *Proc. Graph Drawing 94*,

- volume 894 of *Lecture Notes Comput. Sci.*, pages 404–415. Springer-Verlag, Berlin, 1995.
- [CGJW00] M. Closson, S. Gartshoer, J. Johansen, and S.K. Wismath. Fully dynamic 3-dimensional orthogonal graph drawing. *J. Graph Algorithms Appl.*, 5:1–34, 2000.
 - [CGT96] M. Chrobak, M.T. Goodrich, and R. Tamassia. Convex drawings of graphs in two and three dimensions. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 319–328, 1996.
 - [CS97] T. Calamoneri and A. Sterbini. Drawing 2-, 3-, and 4-colorable graphs in $o(n^2)$ volume. In S. North, editor, *Proc. Graph Drawing 96*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 53–62. Springer-Verlag, Berlin, 1997.
 - [DDPP02] G. Di Battista, W. Didimo, M. Patrignani, and M. Pizzonia. Drawing database schemas. *Software—Practice and Experience*, 32:1065–1098, 2002.
 - [DETT99] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing*. Prentice-Hall, Upper Saddle River, 1999.
 - [DF97] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, 1997.
 - [DFH⁺01a] V. Dujmović, M.R. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Radge, F. Rosamond, M. Suderman, S.H. Whitesides, and D.R. Wood. A fixed parameter approach to two-layer crossing minimization. In P. Mutzel, M. Jünger, and S. Leipert, editors, *Proc. Graph Drawing 01*, volume 2265 of *Lecture Notes Comput. Sci.*, pages 1–15. Springer-Verlag, Berlin, 2001.
 - [DFH⁺01b] V. Dujmović, M.R. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Radge, F. Rosamond, M. Suderman, S.H. Whitesides, and D.R. Wood. On the parameterized complexity of layered graph drawing. In *Proc. European Symp. Algorithms*, volume 2161 of *Lecture Notes Comput. Sci.*, pages 488–499. Springer-Verlag, Berlin, 2001.
 - [DGL⁺97] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara. An experimental comparison of four graph drawing algorithms. *Comput. Geom. Theory Appl.*, 7(5–6):303–325, 1997.
 - [DGL⁺00] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara. Drawing directed acyclic graphs: An experimental study. *Internat. J. Comput. Geom. Appl.*, 10:623–648, 2000.
 - [DH96] R. Davidson and D. Harel. Drawing graphics nicely using simulated annealing. *ACM Trans. Graph.*, 15:301–331, 1996.
 - [Dil90] M.B. Dillencourt. Realizability of Delaunay triangulations. *Inform. Process. Lett.*, 33:283–287, 1990.
 - [DL98] G. Di Battista and G. Liotta. Upward planarity checking: “faces are more than polygons.” In S.H. Whitesides, editor, *Proc. Graph Drawing 98*, volume 1547 of *Lecture Notes Comput. Sci.*, pages 72–86. Springer-Verlag, Berlin, 1998.
 - [DLL95] G. Di Battista, W. Lenhart, and G. Liotta. Proximity drawability: A survey. In R. Tamassia and I.G. Tollis, editors, *Graph Drawing (Proc. GD ’94)*, volume 894 of *Lecture Notes Comput. Sci.*, pages 328–339. Springer-Verlag, Berlin, 1995.
 - [DLPT98] O. Devillers, G. Liotta, F.P. Preparata, and R. Tamassia. Checking the convexity of polytopes and the planarity of subdivisions. *Comput. Geom. Theory Appl.*, 11:187–208, 1998.
 - [DLW02a] E. Di Giacomo, G. Liotta, and S.K. Wismath. Drawing series-parallel graphs on a box. In *14th Canad. Conf. Comput. Geom.*, 2002.

- [DLW02b] E. Di Giacomo, G. Liotta, and S.K. Wismath. The k-lines drawability problem for series-parallel graphs. Tech. Rep. TR-CS-02-02, Dept. Computer Science, Univ. Lethbridge, 2002.
- [DMW02] V. Dujmović, P. Morin, and D.R. Wood. Pathwidth and three-dimensional straight line grid drawings of graphs. In M.T. Goodrich, editor, *Graph Drawing (Proc. GD'02)*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 42–53. Springer-Verlag, Berlin, 2002.
- [DPS02] J. Diaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Comput. Surv.*, 34:313–356, 2002.
- [DWo03a] V. Dujmović and D.R. Wood. Tree-partitions of k -trees with applications in graph layout. In *Proc. 29th Workshop Graph Th. Concepts Comput. Sci.*, volume 2880 of *Lecture Notes Comput. Sci.*, pages 205–217. Springer-Verlag, Berlin, 2003.
- [DWo03b] V. Dujmović and D.R. Wood. New results in graph layout. Tech. Report TR-2003-04, School of Computer Science, Carleton Univ., Canada, 2003.
- [DW03] V. Dujmović and S.H. Whitesides. An efficient fixed parameter tractable algorithm for 1-sided crossing minimization. In M.T. Goodrich and S. Kobourov, editors, *Proc. Graph Drawing 02*, volume 2528 of *Lecture Notes Comput. Sci.*, pages 42–53, Springer-Verlag, Berlin, 2003.
- [Ead84] P. Eades. A heuristic for graph drawing. *Congr. Numer.*, 42:149–160, 1984.
- [EH00] P. Eades and M.L. Huang. Navigating clustered graphs using force-directed methods. *J. Graph Algorithms Appl.*, 4:157–181, 2000.
- [ELMS95] P. Eades, W. Lai, K. Misue, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Languages Comput.*, 6:183–210, 1995.
- [FGW01] S. Felsner, G. Liotta, and S.K. Wismath. Straight line drawings on restricted integer grids in two and three dimensions. In P. Mutzel, M. Jünger, and S. Leipert, editors, *Graph Drawing (Proc. GD '01)*, volume 2265 of *Lecture Notes Comput. Sci.*, pages 328–342. Springer-Verlag, Berlin, 2001.
- [FR91] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software—Practice and Experience*, 21:1129–1164, 1991.
- [GGK01] P. Gajer, M.T. Goodrich, and S.G. Kobourov. A multi-dimensional approach to force-directed layout of large graphs. In J. Marks, editor, *Proc. Graph Drawing 00*, volume 1984 of *Lecture Notes Comput. Sci.*, pages 211–221. Springer-Verlag, Berlin, 2001.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, New York, 1979.
- [GKNV93] E.R. Gansner, E. Koutsofios, S.C. North, and K.P. Vo. A technique for drawing directed graphs. *IEEE Trans. Software Eng.*, 19:214–230, 1993.
- [GNV88] E.R. Gansner, S.C. North, and K.P. Vo. DAG – A program that draws directed graphs. *Software—Practice and Experience*, 18:1047–1062, 1988.
- [Gro01] M. Grohe. Computing crossing numbers in quadratic time. In *Sympos. the Theory of Computing (Proc. STOC 2001)*, pages 231–236, 2001.
- [GT95] A. Garg and R. Tamassia. Upward planarity testing. *Order*, 12:109–133, 1995.
- [GTV96] A. Garg, R. Tamassia, and P. Vocca. Drawing with colors. In *Proc. 4th Annu. European Sympos. Algorithms*, volume 1136 of *Lecture Notes Comput. Sci.*, pages 12–26. Springer-Verlag, Berlin, 1996.

- [Him95] M. Himsolt. Comparing and evaluating layout algorithms within GraphEd. *J. Visual Languages Comput.*, 6:255–273, 1995 (special issue on graph visualization, I.F. Cruz and P. Eades, editors).
- [HMM00] I. Herman, G. Melancon, and M.S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. Visualization Comput. Graph.*, 6:24–43, 2000.
- [JM96] M. Jünger and P. Mutzel. Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica*, 16:33–59, 1996.
- [JM97] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: performance of exact and heuristics algorithms. *J. Graph Algorithms Appl.*, 1:1–25, 1997.
- [KE01] M. Kaufmann and D. Wagner (editors). *Drawing Graphs Methods and Models*. Springer-Verlag, Berlin, 2001.
- [KKM01] G. Klau, K. Klein, and P. Mutzel. An experimental comparison of orthogonal compaction algorithms. In J. Marks, editor, *Proc. Graph Drawing 00*, volume 1984 of *Lecture Notes Comput. Sci.*, pages 37–51, 2001.
- [KS80] J.B. Kruskal and J.B. Seery. Designing network diagrams. In *Proc. 1st General Conf. Social Graphics*, pages 22–50. U.S. Department of the Census, 1980.
- [LE95] T. Lin and P. Eades. Integration of declarative and algorithmic approaches for layout creation. In R. Tamassia and I.G. Tollis, editors, *Proc. Graph Drawing 94*, volume 894 of *Lecture Notes Comput. Sci.*, pages 376–387. Springer-Verlag, Berlin, 1995.
- [LL96] W. Lenhart and G. Liotta. Drawing outerplanar minimum weight triangulations. *Inform. Process. Lett.*, 6:253–260, 1996.
- [LL02] W. Lenhart and G. Liotta. The drawability problem for minimum weight triangulations. *Theoret. Comput. Sci.*, 270:261–286, 2002.
- [LLMW98] G. Liotta, A. Lubiw, H. Meijer, and S.H. Whitesides. The rectangle of influence drawability problem. *Comput. Geom. Theory Appl.*, 10:1–22, 1998.
- [LM03] G. Liotta and H. Meijer. Voronoi drawings of trees. *Comput. Geom. Theory Appl.*, 24:147–178, 2003.
- [LMP01] N. Leash, J. Marks, and M. Patrignani. Interactive partitioning. In *Proc. Graph Drawing 00*, volume 1984 of *Lecture Notes Comput. Sci.*, pages 31–36. Springer-Verlag, Berlin, 2001.
- [LS93] A. Lubiw and N. Sleumer. Maximal outerplanar graphs are relative neighborhood graphs. In *Canad. Conf. Comput. Geom. (Proc. CCCG '93)*, pages 198–203, 1993.
- [MSNS⁺99] K. Mehlhorn, T. Schilz, S. Näher, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. *Comput. Geom. Theory Appl.*, 12:85–113, 1999.
- [NW02] S.C. North and G. Woodhull. Online hierarchical graph drawing. In P. Mutzel, M. Jünger, and S. Leipert, editors, *Proc. Graph Drawing 01*, volume 2265 of *Lecture Notes Comput. Sci.*, pages 232–246. Springer-Verlag, Berlin, 2002.
- [PAC02] H.C. Purchase, J.A. Allder, and D. Carrington. Aesthetics in UML diagrams: User preferences. *J. Graph Algorithms Appl.*, 6:255–279, 2002.
- [PST97] A. Papakostas, J. Six, and I.G. Tollis. Experimental and theoretical results in interactive orthogonal graph drawing. In *Proc. Graph Drawing 96*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 371–386. Springer-Verlag, Berlin, 1997.
- [PT98] A. Papakostas and I.G. Tollis. Interactive orthogonal graph drawing. *IEEE Trans. Comput.*, C-47:83–110, 1998.

- [PTT97] J. Pach, T. Thiele, and G. Tóth. Three-dimensional grid drawings of graphs. In G. Di Battista, editor, *Proc. Graph Drawing 97*, volume 1353 of *Lecture Notes Comput. Sci.*, pages 47–51. Springer-Verlag, Berlin, 1997.
- [Riv93] I. Rival. Reading, drawing, and order. In I.G. Rosenberg and G. Sabidussi, editors, *Algebras and Orders*, pages 359–404. Kluwer Academic, Dordrecht, 1993.
- [San99] G. Sander. Graph layout for applications in compiler construction. *Theoret. Comput. Sci.*, 217:175–214, 1999.
- [SSV95] F. Shahrokhi, L.A. Székely, and I. Vrtó. Crossing numbers of graphs, lower bound techniques and algorithms: a survey. In R. Tamassia and I.G. Tollis, editors, *Proc. Graph Drawing 94*, volume 894 of *Lecture Notes Comput. Sci.*, pages 131–142. Springer-Verlag, Berlin, 1995.
- [STT81] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical systems. *IEEE Trans. Syst. Man Cybern.*, SMC-11:109–125, 1981.
- [Sug02] K. Sugiyama. *Graph Drawing and Applications for Software and Knowledge Engineers*. World Scientific, Singapore, 2002.
- [Tam87] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16:421–444, 1987.
- [Tam90a] R. Tamassia. Drawing algorithms for planar st-graphs. *Australasian J. Combinatorics*, 2:217–235, 1990.
- [Tam90b] R. Tamassia. Planar orthogonal drawings of graphs. In *Proc. IEEE Internat. Sympos. Circuits Systems*, pages 319–322, 1990.
- [Tam99] R. Tamassia. Advances in the theory and practice of graph drawing. *Theoret. Comput. Sci.*, 17:235–254, 1999.
- [TDB88] R. Tamassia, G. Di Battista, and C. Batini. Automatic graph drawing and readability of diagrams. *IEEE Trans. Syst. Man Cybern.*, SMC-18:61–79, 1988.
- [VBL⁺00] L. Vismara, G. Di Battista, G. Liotta, R. Tamassia, and F. Vargiu. Experimental studies on graph drawing algorithms. *Software—Practice and Experience*, 30:1235–1284, 2000.
- [WCY00] C.-A. Wang, F. Chin, and B.-T. Yang. Triangulations without minimum weight drawing. In *Algorithms and Complexity (Proc. CIAC '00)*, volume 1767 of *Lecture Notes Comput. Sci.*, pages 163–173. Springer-Verlag, Berlin, 2000.
- [Woo02] D.R. Wood. Queue layouts, tree-width, and three-dimensional graph drawing. In *22nd Found. Software Tech. Theoret. Comput. Sci.*, volume 2556 of *Lecture Notes Comput. Sci.*, pages 348–359. Springer-Verlag, Berlin, 2002.

53 SPLINES AND GEOMETRIC MODELING

Chandrajit L. Bajaj

INTRODUCTION

Piecewise polynomials of fixed degree and continuously differentiable up to some order are known as *splines* or *finite elements*. Splines are used in applications ranging from computer-aided design, computer graphics, data visualization, geometric modeling, and image processing to the solution of partial differential equations via finite element analysis. The spline-fitting problem of constructing a mesh of finite elements that interpolate or approximate multivariate data is by far the primary research problem in geometric modeling. *Parametric splines* are vectors of a set of multivariate polynomial (or rational) functions while *implicit splines* are zero contours of collections of multivariate polynomials. This chapter dwells mainly on spline surface fitting methods in real Euclidean space. We first discuss tensor product surfaces (Section 53.1), perhaps the most popular. The next sections cover generalized spline surfaces (Section 53.2), free-form surfaces (Section 53.3), and subdivision surfaces (Section 53.4). This classification is not strict, and some overlap exists. Interactive editing of surfaces is discussed in the final section (Section 53.5).

The various spline methods may be distinguished by several criteria:

- Implicit or parametric representations.
- Algebraic and geometric degree of the spline basis.
- Number of surface patches required.
- Computation (time) and memory (space) required.
- Stability of fitting algorithms.
- Local or nonlocal interpolation.
- Splitting or nonsplitting of input mesh.
- Convexity or nonconvexity of the input and solution.
- Fairness of the solution (first- and second-order variation).

These distinctions will guide the discussions throughout the chapter.

53.1 TENSOR PRODUCT SURFACES

Tensor product B-splines have emerged as the polynomial basis of choice for working with parametric surfaces. The theory of tensor product patches requires that data have a rectangular geometry and that the parametrizations of opposite boundary curves be similar. It is based on the concept of bilinear interpolation. The most general results obtained to date are summarized in [Table 53.1.1](#), and will be discussed below.

GLOSSARY

Affine invariance: A property of a curve or surface generation scheme, implying invariance with respect to whether computation of a point on a curve or surface occurs before or after an affine map is applied to the input data.

A-spline: Collection of bivariate Bernstein-Bézier polynomials, each over a triangle and with prescribed geometric continuity, such that the zero contour of each polynomial defines a smooth and single-sheeted real algebraic curve segment. (“A” stands for “algebraic.”)

A-patch: Smooth and “functional” zero contour of a Bernstein-Bézier polynomial over a tetrahedron.

Barycentric combination: A weighted average where the sum of the weights equals one.

Barycentric coordinates: A point in \mathbb{R}^2 may be written as a unique barycentric combination of three points. The coefficients in this combination are its barycentric coordinates. Similarly, a point in \mathbb{R}^3 may be written as a unique barycentric combination of four points. See Figure 28.2.1.

Basis function: Functions form linear spaces, which have bases. The elements of these bases are the basis functions.

Bernstein-Bézier form: Let $p_1, p_2, p_3, p_4 \in \mathbb{R}^3$ be affinely independent. Then the tetrahedron with these points as vertices is $V = [p_1 p_2 p_3 p_4]$. Any polynomial $f(p)$ of degree n can be expressed in the Bernstein-Bézier (BB) form over V as

$$f(p) = \sum_{|\lambda|=n} b_\lambda B_\lambda^n(\alpha), \quad \lambda \in \mathcal{Z}_+^4, \quad (53.1.1)$$

where

$$B_\lambda^n(\alpha) = \frac{n!}{\lambda_1! \lambda_2! \lambda_3! \lambda_4!} \alpha_1^{\lambda_1} \alpha_2^{\lambda_2} \alpha_3^{\lambda_3} \alpha_4^{\lambda_4}$$

are Bernstein polynomials, $|\lambda| = \sum_{i=1}^4 \lambda_i$ with $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^T$, the barycentric coordinates of p are $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$, $b_\lambda = b_{\lambda_1 \lambda_2 \lambda_3 \lambda_4}$ are the *control points*, and \mathcal{Z}_+^4 is the set of all four-dimensional vectors with nonnegative integer components.

Bernstein polynomials: The basis functions for Bézier curves and surfaces.

Bézier curve: A curve whose points are determined by the parameter u in the equation $\sum_{i=0}^n B_i^n(u) P_i$, where the $B_i^n(u)$ are basis functions, and the P_i control points.

Bilinear interpolation: A tensor product of two orthogonal linear interpolants and the “simplest” surface defined by values at four points on a rectangle.

Blending functions: The basis functions used by interpolation schemes such as Gordon surfaces.

B-spline surface: Traditionally, a tensor product of curves defined using piecewise basis polynomials (B-spline basis). Any B-spline can be written in piecewise Bézier form. (“B” stands for “basis.”)

C^k continuity: Smoothness defined in terms of matching of up to k th order derivatives along patch boundaries.

Control point: The coefficients in the expansion of a Bézier curve in terms of Bernstein polynomials.

Convex hull: The smallest convex set that contains a given set.

Convex set: A set such that the straight line segment connecting any two points of the set is completely contained within the set.

G^k continuity: Geometric continuity with smoothness defined in terms of matching of up to k th order derivatives allowing for reparametrization. For example, G^1 smoothness is defined in terms of matching tangent planes along patch boundaries.

Knots: A spline curve is defined over a partition of an interval of the real line. The points that define the partition are called knots.

Mesh: A decomposition of a geometric domain into finite elements; see [Section 22.4](#).

Ruled (lofted) surface: A surface that interpolates two given curves using linear interpolation.

Tensor product surfaces: A surface represented with basis functions that are constructed as products of univariate basis functions. A tensor product Bézier surface is given by the equation $\sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) P_{ij}$, where the $B_i^n(u)$ and $B_j^m(v)$ are the univariate Bernstein polynomial basis functions, and the P_{ij} are control points.

Transfinite interpolation: Interpolating entire curves as opposed to values at discrete points.

Variation diminishing: A curve or surface scheme has this property if its output “wiggles less” than the control points from which it is constructed.

PARAMETRIC BÉZIER AND B-SPLINES

Tensor product Bézier surfaces are obtained by repeated applications of bilinear interpolation. Properties of tensor product Bézier patches include affine invariance, the “convex hull property,” and the variation diminishing property. The boundary curves of a patch are polynomial curves that have their Bézier polygon given by the boundary polygons of the control net of the patch. Hence the four corners of the control net lie on the patch.

Piecewise bicubic Bézier patches may be used to fit a C^1 surface through a rectangular grid of points. After the rectangular network of curves has been created, there are four coefficients left to determine the corner twists of each patch. These four corner twists cannot be specified independently and must satisfy a “compatibility constraint.” Common twist estimation methods include zero twists, Adini’s twist, Bessel twist, and Brunet’s twist [Far98]. To obtain C^1 continuity between two patches the directions and lengths of the polyhedron edges must be matched across the common polyhedron boundary defining the common boundary curve. Piecewise bicubic Hermite patches are similar to the piecewise bicubic Bézier patches, but take points, partials, and mixed partials as input. The mixed partials affect only the interior shape of the patch, and are also called *twist vectors*.

It is not possible to model a general closed surface or a surface with handles as a single nondegenerate B-spline. To represent free-form surfaces a significant amount of recent work has been done in the areas of geometric continuity, nontensor product

 TABLE 53.1.1 Tensor product surfaces.

TYPE	INPUT	PROPERTIES
Piecewise Bézier and Hermite	rectangular grid of points, corner twists	C^1 , initial global data survey data to determine the tangent and cross-derivative vectors at patch corners
Bicubic B-spline	rectangular grid of points	C^1
Coons patches	4 boundary curves	C^1
Gordon surfaces	rectangular network of curves	C^1 , Gregory square
Biquadratic B-spline	limit of Doo-Sabin subdivision of rectangular faces	C^1
Bicubic B-spline	limit of Catmull-Clark subdivision of rectangular faces	C^1
Biquadratic splines	control points on mesh with arbitrary topology	G^1 , system of linear equations for smoothness conditions around singular vertices
Biquartic splines	cubic curve mesh	C^1 , interpolate second-order data at mesh points
Bisextic B-spline	rectangular network of cubic curves	C^1
Triquadratic/tricubic A-patches	rectilinear 3D grid points	C^1 , local calculation of first-order cross derivatives
Triple products of B-splines	rectangular boxes	

patches, and generalizing B-splines [CF83, Pet90a, Pet90b, GW91, DM83, GH87]. Common schemes include splitting, convex combinations of blending functions, subdivision, and local interpolation by construction [for95, HF84, MLL⁺92, Pet93, Pet02].

IMPLICIT BÉZIER AND B-SPLINES

Patrikalakis and Kriegis [PK89] demonstrate how implicit algebraic surfaces can be manipulated in rectangular boxes as functions in a tensor product B-spline basis. This work, however, leaves open the problem of selecting weights or specifying knot sequences for C^1 meshes of tensor product implicit algebraic surface patches that fit given spatial data. Moore and Warren [MW91] extend the “marching cubes” scheme to compute a C^1 piecewise tensor product triquadratic approximation to scattered data using a Powell-Sabin-like split over subcubes. In [BBCS99] an incremental and adaptive approach is used to construct C^1 spline functions defined over an octree subdivision that approximate a dense set of multiple volumetric scattered scalar values. Further details are provided in subsequent sub-sections on A-patches and implicit free-form surfaces.

COONS PATCHES AND GORDON SURFACES

Coons patches interpolate four boundary curves. They are constructed by composing two ruled, or lofted, surfaces and one bilinear surface, and hence are called *bilinearly blended surfaces*. A Coons patch has four blending functions $f_i(u)$, $g_i(v)$, $i = 1, 2$. There are only two restrictions on the f_i and g_i : each pair must

sum to one, and we must have $f_1(0) = g_1(0) = 1$ and $f_2(1) = g_2(1) = 0$ in order to interpolate.

A network of curves may be filled in with a C^1 surface using bicubically blended Coons patches. For this the four twists at the data points and the four cross boundary derivatives must be computed. Compatibility problems may arise in computing the twists. If $\mathbf{x}(u, v)$ is twice differentiable, we have $\mathbf{x}_{uv} = \mathbf{x}_{vu}$, but this simplification does not apply here. One approach is to adjust the given data so that the incompatibilities disappear. Or if the data cannot be changed one can use a method known as *Gregory's square* that replaces the constant twist terms by variable twists that are computed from the cross boundary derivatives. The resulting surface does not have continuous twists at the corners and is rational parametric, which may not be acceptable geometry for certain geometric modeling systems.

Gordon surfaces are a generalization of Coons patches used to construct a surface that interpolates a rectangular network of curves. The idea is to take a univariate interpolation scheme, apply it to all curves, add the resulting surfaces, and subtract the tensor product interpolant that is defined by the univariate scheme. Polynomial interpolation or spline interpolation schemes may be used. Methods for Coons patches and Gordon surfaces can be formulated in terms of Boolean sums and projectors. This has also been generalized to create triangular Coons patches.

53.2 GENERALIZED SPLINE SURFACES

B-PATCHES

The B-patches developed by Seidel [Sei89, DMS92] are based on the study of symmetric recursive evaluation algorithms, and are defined by generalizing the deBoor algorithm for the evaluation of a B-spline segment from curves to surfaces. A polynomial surface that has a symmetric recursive evaluation algorithm is called a **B-patch**. B-patches generalize Bézier patches over triangles, and are characterized by control points and a three-parameter family of knots. Every bivariate polynomial $F : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ of degree n has a unique representation

$$F(U) = \sum_{|\vec{i}|=n} N_{\vec{i}}^n(U) P_{\vec{i}}, \quad P_{\vec{i}} \in \mathbb{R}^d$$

as a B-patch, with parameters $\mathcal{K} = R_0, \dots, R_{n-1}, S_0, \dots, S_{n-1}, T_0, \dots, T_{n-1}$ in \mathbb{R}^2 , if the parameters (R_i, S_j, T_k) are affinely independent for $0 \leq |\vec{i}| \leq n - 1$. The real-valued polynomials $N_{\vec{i}}^n(U)$ are called the **normalized B-weights** of degree n over \mathcal{K} .

MULTISIDED PATCHES

Multisided patches can be generated in basically two ways. Either the polygonal domain which is to be mapped into \mathbb{R}^3 is subdivided in the parametric plane, or one uniform equation is used as a combination of equations. In the former case,

triangular or rectangular elements are put together or recursive subdivision is applied. In the latter case, either the known control point methods are generalized, or a weighted sum of interpolants is used. With constrained domain mapping, a domain point for an n -sided patch is represented by n dependent parameters. If the remainder of the parameters can be computed when any two parameters are independently chosen, it is called a *symmetric system of parameters*. The main results from multisided patch schemes obtained to date are summarized in Table 53.2.1.

TABLE 53.2.1 Multisided schemes.

TYPE	LIMITATIONS	PROPERTIES	DOMAIN POINTS
Sabin	$n=3,5$	C^1	constrained domain mapping, symmetric system of parameters
Gregory/Charrot	$n=3,5$	C^1	barycentric coordinates
Hosaka/Kimura	$n \leq 6$	C^1	constrained domain mapping, symmetric system of parameters
Varady		VC^1	$2n$ variables constrained along polygon sides
Base points	$n = 4, 5, 6$	rational Bézier surfaces	base points in the parametric domain map to rational curves in \mathbb{R}^3
S-patches	multisided	G^1 rational bi-quadratic and bicubic B-splines	embed n -sided domain polygon into simplex of dimension $n - 1$
Multisided A-patches	“functional” bd curves	C^1, C^2 implicit Bezier surfaces	Hermite interpolation of boundary curves

TRIANGULAR RATIONAL PATCHES WITH BASE POINTS

Another approach to creating multisided patches is to introduce base points into rational parametric functions. Base points are parameter values for which the homogeneous coordinates (x, y, z, w) are mapped to $(0, 0, 0, 0)$ by the rational parametrization. Gregory’s patch [Gre83] is defined using a special collection of rational basis functions that evaluate to 0/0 at vertices of the parametric domain, and thus introduce base points in the resulting parametrization. Warren [War92] uses base points to create parametrizations of four-, five-, and six-sided surface patches using rational Bézier surfaces defined over triangular domains. Setting a triangle of weights to zero at one corner of the domain triangle produces a four-sided patch that is the image of the domain triangle.

S-PATCHES

Loop and DeRose [LD89, LD90] present generalizations of biquadratic and bicubic B-spline surfaces that are capable of representing surfaces of arbitrary topology by placing restrictions on the connectivity of the control mesh, relaxing C^1 continuity to G^1 (geometric) continuity, and allowing n -sided finite elements. This generalized view considers the spline surface to be a collection of possibly rational polynomial maps from independent n -sided polygonal domains, whose union possesses continuity of some number of geometric invariants, such as tangent planes. This more

general view allows patches to be sewn together to describe free-form surfaces in more complex ways.

An n -sided S-patch S is constructed by embedding its n -sided domain polygon P into a simplex Δ whose dimension is one less than the number of sides of the polygon. The edges of the polygon map to edges of the simplex. A Bézier simplex \mathbf{B} is then constructed using Δ as a domain. The patch representation S is obtained by restricting the Bézier simplex to the embedded domain polygon.

A-PATCHES

The A-patch technique provides simple ways to guarantee that a constructed implicit surface is single-sheeted and free of undesirable singularities. The technique uses the zero contouring surfaces of trivariate Bernstein-Bézier polynomials to construct a piecewise smooth surface. We call such iso-surfaces *A-patches*. Algorithms to fill an n -sided hole, using either a single multisided A-patch or a network of A-patches, are given in [BE95]. The blends may be C^0 , C^1 , or C^2 exact fits (interpolation), as well as C^1 or C^2 least squares fits (interpolation and approximation).

For degree-bounded patches, a triangular network of A-patches for the hole may be generated in two ways. First, the n -sided hole is projected onto a plane and the result of a planar triangulation is projected back onto the hole. Second, an initial multisided A-patch is created for the hole and then a coarse triangulation for the patch is generated using a rational spline approximation [BX94].

MULTIVARIATE BOX SPLINES AND SIMPLEX SPLINES

Multivariate splines are a generalization of univariate B-splines to a multivariate setting. Multivariate splines have applications in data fitting, computer-aided design, the finite element method, and image analysis. Work on splines has traditionally been for a given planar triangulation using a polynomial function basis. Box splines are multivariate generalizations of B-splines with uniform knots. Many of the basis functions used in finite element calculations on uniform triangles occur as special instances of box splines. In general a box spline is a locally supported piecewise polynomial. One can define translates of box splines that form a negative partition of unity.

In the bivariate case, box splines correspond to surfaces defined over a regular tessellation of the plane. If the tessellation is composed of triangles, it is possible to represent the surface as a collection of Bernstein-Bézier patches. The two most commonly used special tessellations arise from a rectangular grid by drawing in lines in north-easterly diagonals in each subrectangle or by drawing in both diagonals for each subrectangle. For these special triangulations there is an elegant way to construct locally supported splines.

Multivariate splines defined as projections of simplices are called *simplex splines*. Auerbach [AMNS91] constructs approximations with simplex splines over irregular triangles. Bivariate quadratic simplicial B-splines defined by their corresponding sets of knots derived from a (suboptimal) constrained Delaunay triangulation of the domain are employed to obtain a C^1 surface. This approach is well suited for scattered data.

Fong and Seidel [FS86, FS92] construct multivariate B-splines for quadratics and cubics by matching B-patches with simplex splines. The surface scheme is an

approximation scheme based on blending functions and control points and allows the modeling of C^{k-1} continuous piecewise polynomial surfaces of degree k over arbitrary triangulations of the parameter plane. The resulting surfaces are defined as linear combinations of the blending functions, and are parametric piecewise polynomials over a triangulation of the parameter plane whose shape is determined by their control points.

53.3 FREE-FORM SURFACES

The representation of free-form surfaces is one of the major issues in geometric modeling. These surfaces are generally defined in a piecewise manner by smoothly joining several, mostly four-sided, patches. Common approaches to constructing surfaces over irregular meshes are local construction, blending polynomial pieces, and splitting.

GLOSSARY

Blending polynomial pieces: Constructing k pieces for a k -sided mesh facet such that each piece matches a part of the facet data, and a convex combination of the pieces matches the whole.

Vertex enclosure constraint: Not every mesh of polynomial curves with a well-defined tangent plane at the mesh points can be interpolated by a smooth regularly parameterized surface with one polynomial piece per facet. This constraint on the mesh is a necessary and sufficient condition to guarantee the existence of such an interpolant [Pet91]. Rational patches, singular parametrizations, and the splitting of patches are techniques to enforce the vertex enclosure constraint.

MAIN RESULTS

Blending approaches prescribe a mesh of boundary curves and their normal derivatives. For this approach, however, the existence of a well-defined tangent plane at the data points is not sufficient to guarantee the existence of a C^1 mesh interpolant, because the mixed derivatives p_{uv} and p_{vu} are given independently at any point p . Splitting approaches, on the other hand, expect to be given at least tangent vectors at the data points, and sometimes the complete boundary. Mann et al. [MLL⁺92] conclude that local polynomial interpolants generally produce unsatisfactory shapes.

With splitting schemes, every triangle in the triangulation of the data points (also called a *macro-triangle*) is split into several *mini-triangles*. Split-triangle interpolants do not require derivative information of higher order than the continuity of the desired interpolant. The simplest of the split-triangle interpolants is the C^1 Clough-Tocher interpolant. Each vertex is joined to the centroid, and the macro-triangle is split into three mini-triangles. The first-order data that this interpolant requires are position and gradient value at the macro-triangle vertices, plus some cross-boundary derivative at the midpoint of each edge. There are twelve data per

macro-triangle, and cubic polynomials are used over each mini-triangle. The C^1 Powell-Sabin interpolants produce C^1 piecewise quadratic interpolants to C^1 data at the vertices of a triangulated data set. Each macro-triangle is split into six or twelve mini-triangles.

PARAMETRIC PATCH SCHEMES

These patches are given in vector valued parametric form, generally mapping a rectangular or triangular parametric domain into \mathbb{R}^3 . Parametric free-form surface patch schemes are summarized in Table 53.3.1.

TABLE 53.3.1 Free-form parametric schemes.

DEGREE	SCHEME	INPUT	PROPERTIES
Piecewise biquartic	local interpolation	cubic curve mesh	C^1 , interpolate second-order data at mesh points
Piecewise biquadratic	G-edges	control points on a mesh with arbitrary topology	G^1 , system of linear eqns for smoothness conditions around singular vertices
Sextic triangular pieces	approximation, no local splitting	triangular control mesh	G^1
Quadratic/cubic triangular pieces	splitting, subdivision	irregular mesh of points	C^1 , refine mesh by Doo-Sabin to isolate regions of irregular points

IMPLICIT PATCH SCHEMES

While it is possible to model a general closed surface of arbitrary genus as a single implicit surface patch, the geometry of such a global surface is difficult to specify, interactively control, and polygonize. The main difficulties stem from the fact that implicit representations are iso-contours which generally have multiple real sheets, self-intersections, and several other undesirable singularities. Looking on the bright side, implicit polynomial splines of the same geometric degree have more degrees of freedom compared with parametric splines, and hence potentially are more flexible for approximating a complicated surface with fewer pieces and for achieving a higher order of smoothness. The potential of implicits remains largely latent: virtually all commercial and many research modeling systems are based on the parametric representation. An exception is SHASTRA, which allows modeling with both implicit and parametric splines [Baj93]. Implicit free-form surface schemes are summarized in Table 53.3.2.

A-SPLINES

An *A-spline* is a piecewise G^k -continuous chain of real algebraic curve segments, such that each curve segment is a smooth and single-sheeted zero contour of a bivariate Bernstein-Bézier polynomial (called a regular curve segment). A-splines are

TABLE 53.3.2 Free-form implicit schemes.

DEGREE	SCHEME	INPUT	PROPERTIES
5, 7	local interpolation, no splitting	curve mesh from spatial triangulation	C^1 interpolate or approximate
	simplicial hull construction	spatial triangulation	
	simplicial hull construction, Clough-Tocher split	spatial triangulation	C^1
	simplicial hull construction, Clough-Tocher split of coplanar faces	spatial triangulation	C^1 A-patches, 3 or 4 sides
	simplicial hull construction, Clough-Tocher split of coplanar faces	spatial triangulation	C^2 A-patches

a suitable polynomial form for working with piecewise implicit polynomial curves. A characterization of A-splines defined over triangles or quadrilaterals is available [BX99, XB00], as is a detailing of their applications in curve design and fitting [BX01].

CURVILINEAR MESH SCHEMES

Bajaj and Ihm [Baj92, BI92a] construct implicit surfaces to solve the scattered data-fitting problem. The resulting surfaces approximate or contain with C^1 continuity any collection of points and algebraic space curves with derivative information. Their Hermite interpolation algorithm solves a homogeneous linear system of equations to compute the coefficients of the polynomial defining the algebraic surface. This idea has been extended to C^k (rescaling continuity) interpolate or least squares approximate implicit or parametric curves in space [BIW93]. This problem is formulated as a constrained quadratic minimization problem, where the algebraic distance is minimized instead of the geometric distance.

In a *curvilinear-mesh-based* scheme, Bajaj and Ihm [BI92b] construct low-degree implicit polynomial spline surfaces by interpolating a mesh of curves in space using the techniques of [Baj92, BI92a, BIW93]. They consider an arbitrary spatial triangulation \mathcal{T} consisting of vertices in \mathbb{R}^3 (or more generally, a simplicial polyhedron \mathcal{P} when the triangulation is closed), with possibly normal vectors at the vertex points. Their algorithm constructs a C^1 mesh of real implicit algebraic surface patches over \mathcal{T} or \mathcal{P} . The scheme is local (each patch has independent free parameters) and there is no local splitting. The algorithm first converts the given triangulation or polyhedron into a curvilinear wire frame, with at most cubic parametric curves which C^1 interpolate all the vertices. The curvilinear wire frame is then fleshed to produce a single implicit surface patch of degree at most 7 for each triangular face \mathcal{T} of \mathcal{P} . If the triangulation is convex then the degree is at most 5. Similar techniques exist for parametrics [Pet91, Far86, Sar87]; however, the geometric degrees of the solution surfaces tend to be prohibitively high.

SIMPLEX- AND BOX-BASED SCHEMES

In a *simplex-based* approach, one first constructs a tetrahedral mesh (called the simplicial hull) conforming to a surface triangulation \mathcal{T} of a polyhedron \mathcal{P} . The implicit piecewise polynomial surface consists of the zero set of a Bernstein-Bézier polynomial, defined within each tetrahedron (simplex) of the simplicial hull. A

simplex-based approach enforces continuity between adjacent patches by enforcing that vertex/edge/face-adjacent trivariate polynomials are continuous with one another.

Similar to the trivariate interpolation case, Powell-Sabin or Clough-Tocher splits are used to introduce degree-bounded vertices to prevent the continuity system from propagating globally. Such splitting, however, could result in a large number of patches. However, as only the zero set of the polynomial is of interest, one does not need a complete mesh covering the entire space.

Sederberg [Sed85] showed how various smooth implicit algebraic surfaces, represented in trivariate Bernstein basis form, can be manipulated as functions in Bézier control tetrahedra with finite weights. He showed that if the coefficients of the Bernstein-Bézier form of the trivariate polynomial on the lines that parallel one edge, say L , of the tetrahedron all increase (or decrease) monotonically in the same direction, then any line parallel to L will intersect the zero contour algebraic surface patch at most once.

Guo [Guo91] used cubics to create free-form geometric models and enforced monotonicity conditions on a cubic polynomial along the direction from one vertex to a point of the face opposite the vertex. A Clough-Tocher split is used to subdivide each tetrahedron of the simplicial hull. Dahmen and Thamm-Scharr [DTS93] utilize a single cubic patch per tetrahedron, except for tetrahedra on coplanar faces.

Lodha [Lod92] constructed low degree surfaces with both parametric and implicit representations and investigated their properties. A method is described for creating quadratic triangular Bézier surface patches that lie on implicit quadric surfaces. Another method is described for creating biquadratic tensor product Bézier surface patches that lie on implicit cubic surfaces. The resulting patches satisfy all the standard properties of parametric Bézier surfaces, including interpolation of the corners of the control polyhedron and the convex hull property.

Bajaj and Ihm, Guo, and Dahmen [BI92b, Guo91, Guo93, Dah89] provide heuristics based on monotonicity and least square approximation to circumvent the multiple-sheeted and singularity problems of implicit patches.

Bajaj, Chen, and Xu [BCX95] construct 3- and 4-sided A-patches that are implicit surfaces in Bernstein-Bézier (BB) form and that are smooth and single-sheeted. They give sufficiency conditions for the BB form of a trivariate polynomial within a tetrahedron, such that the zero contour of the polynomial is a single-sheeted nonsingular surface within the tetrahedron, and its cubic-mesh complex for the polyhedron \mathcal{P} is guaranteed to be both nonsingular and single-sheeted. They distinguish between convex and nonconvex facets and edges of the triangulation. A double-sided tetrahedron is built for nonconvex facets and edges, and single-sided tetrahedra are built for convex facets and edges. A generalization of Sederberg's condition is given for a three-sided j -patch where any line segment passing through the j th vertex of the tetrahedron and its opposite face intersects the patch only once. Instead of having coefficients be monotonically increasing or decreasing there is a single sign change condition. There are also free parameters for both local and global shape control.

Reconstructing surfaces and scalar fields defined over the surface from scattered data using implicit Bézier splines is described in [BBX95, BX97, CX01]. See also [Chapter 30](#).

53.4 MULTIRESOLUTION SPLINE SURFACES

SUBDIVISION SURFACES

Subdivision techniques can be used to produce generally pleasing surfaces from arbitrary control meshes. The faces of the mesh need not be planar, nor need the vertices lie on a topologically regular mesh. Subdivision consists of splitting and averaging. Each edge or face is split, and each new vertex introduced by the splitting is positioned at a fixed affine combination of its neighbor's weights. Subdivision schemes are summarized in Table 53.4.1.

TABLE 53.4.1 Subdivision schemes.

TYPE	PROPERTIES
Doo-Sabin; Catmull-Clark	C^1 , interpolate centroids of all faces at each step
Nasri	interpolate points/normals on irregular networks
Loop	C^1 , split each triangle of a triangular mesh into 4 triangles
Hoppe et al.	extends Loop's method to incorporate shape edges in limit surfaces; initial vertices belong to vertex, edge, or face of limit surface
Storry and Ball	C^1 n -sided B-spline patch to fit in bicubic surface, one dof
Yn, Levin, and Gregory	interpolatory butterfly subdivision, modify set of deterministic rules for subdivision
Bajaj, Chen, and Xu	approximation, one step subdivision to build simplicial hull, C^1 cubic and C^2 quintic A-patches
Reif	regularity conditions

MAIN ALGORITHMS

Subdivision algorithms start with a polyhedral configuration of points, edges, and faces. The control mesh will in general consist of large regular regions and isolated singular regions. Subdivision enlarges the regular regions of the control net and shrinks the singular regions. Each application of the subdivision algorithm constructs a refined polyhedron, consisting of more points and smaller faces, tending in the limit to a smooth surface. In general the new control points are computed as linear combinations of old control points. The associated matrix is called the **subdivision matrix**. Except for some special cases, the limiting surface does not have an explicit analytic representation. If each face of the polyhedron is a rectangle, the Doo-Sabin subdivision rules generate biquadratic tensor product B-splines, and the Catmull-Clark subdivision rules generate bicubic tensor product B-splines. Also, the subdivision technique of Loop generates three-direction box splines.

Reif [Rei92] presents a unified approach to subdivision algorithms for meshes with arbitrary topology and gives a sufficient condition for the regularity of the surface. The existence of a smooth regular parametrization for the generated surface

near the point is determined from the leading eigenvalues of the subdivision matrix and an associated characteristic map. Details and further discussion of recent subdivision schemes are available from [WW02].

APPROXIMATING SCHEMES

Bajaj, Chen, and Xu [BCX94] construct an “inner” simplicial hull after one step of subdivision of the input polyhedron \mathcal{P} . As in traditional subdivision schemes, \mathcal{P} is used as a control mesh for free-form modeling, while an inner surface triangulation \mathcal{T} of the hull can be considered as the second level mesh. Both a C^1 mesh with cubic A-patches and a C^2 mesh with quintic patches can be constructed to approximate the polyhedron \mathcal{P} [XBE01].

INTERPOLATING SCHEMES

There are two key approaches to constructing interpolating subdivision surfaces. One approach is to first compute a new configuration of vertices, edges, and faces with the same topology such that the vertices of the new configuration converge to the given vertices in the limit. The subdivision technique is then applied to this new configuration. The other approach is to modify the deterministic subdivision rules so that the limiting surface interpolates the vertices.

HIERARCHICAL SPLINES

Hierarchical splines are a multiresolution approach to the representation and manipulation of free-form surfaces. A hierarchical B-spline is constructed from a base surface (level 0) and a series of overlays are derived from the immediate parent in the hierarchy. Forsey and Bartels [FB88] present a refinement scheme that uses a hierarchy of rectangular B-spline overlays to produce C^2 surfaces. Overlays can be added manually to add detail to the surface, and local or global changes to the surface can be made by manipulating control points at different levels.

Forsey and Wang [FW93] create hierarchical bicubic B-spline approximations to scanned cylindrical data. The resulting hierarchical spline surface is interactively modifiable using editing capabilities of the hierarchical surface representation, allowing either local or global changes to surface shape while retaining the details of the scanned data. Oscillations occur, however, when the data have high-amplitude or high-frequency regions. Forsey and Bartels use a hierarchical wavelet-based representation for fitting tensor product parametric spline surfaces to gridded data in [FB95]. The multiresolution representation is extend to include arbitrary meshes in [EDD⁺95]. The method is based on approximating an arbitrary mesh by a special type of mesh and using a continuous parametrization of the arbitrary mesh over a simple domain mesh.

Further discussion of wavelet based multiresolution schemes and some of their applications is available from [SDS96].

53.5 PHYSICALLY BASED APPROACHES TO SURFACE MODELING

ENERGY-BASED SPLINES

A group of researchers [TF88, PB88, PTBK87, WFB87, BHN99] have presented discrete models which are based extensively on the theory of elasticity and plasticity, using energy fields to define and enforce constraints. Haumann [Hau87] used the same approach but used a triangulated model and a simpler physical model based on points, springs, and hinges. Thingvold and Cohen [TC90] defined a model of elastic and plastic B-spline surfaces which supports both animation and design operations. The basis for the physical model is a generalized point-mass/spring/hinge model that has been adapted into a simultaneous refinement of the geometric/physical model. Always having a sculptured surface representation as well as the physical hinge/spring/mesh model allows the user to intertwine physical-based operations, such as force application, with geometrical modeling. Refinement operations for spring and hinge B-spline models are compatible with the physics and mathematics of B-spline models. The models of elasticity and plasticity are written in terms of springs and hinges, and can be implemented with standard integration techniques to model realistic motions of elastic and plastic surfaces. These motions are controlled by the physical properties assigned and by kinematic constraints on various portions of the surface. Terzopoulos and Qin [TQ94] develop a dynamic generalization of the nonuniform rational B-spline (NURBS) model. They present a physics-based model that incorporates mass distributions, internal deformation energies, and other physical quantities into the NURBS geometric substrate. These dynamic NURBS can be used in applications such as rounding of solids, optimal surface fitting to unstructured data, surface design from cross-sections, and free-form deformations.

DIFFERENTIAL EQUATIONS AND SURFACE SPLINES

Early research on using partial differential equations (PDEs) to handle surface modeling problems trace back to Bloor et al.'s work at the end of the 1980s ([BW89a, BW89b, BW90]). The basic idea of these papers is the use of biharmonic equations on a rectangular domain to solve blending and hole filling problems. One of the advantages of using the biharmonic equation is that it is linear, and therefore easier to solve. However, the solution of the equation depends on the surface parametrization.

The evolution technique, based on the heat equation $\partial_t x - \Delta x = 0$, has been extensively used in the area of image processing (see [PM87, PR99, Wei98]), where Δ is a 2D Laplace operator. This was extended later to smoothing or fairing noisy surfaces (see [CDR00, DMSB99, DMSB00]). For a surface M , the counterpart of the Laplacian Δ is the Laplace-Beltrami operator Δ_M (see [dC92]). One then obtains the geometric diffusion equation $\partial_t x - \Delta_M x = 0$ for a surface point $x(t)$ on the surface $M(t)$. Taubin [Tau95] discusses the discretized operator of the Laplacian and related approaches in the context of generalized frequencies on meshes. Kobbelt

[Kob96] considers discrete approximations of the Laplacian in the construction of fair interpolatory subdivision schemes. This work was extended in [KCVS98] to arbitrary connectivity for purposes of multi-resolution interactive editing. Desbrun et al. [DMSB99] used an implicit discretization of geometric diffusion to obtain a strongly stable numerical smoothing scheme. The same strategy of discretization is also adopted and analyzed by Deckelnick and Dziuk [DD02] with the conclusion that this scheme is unconditionally stable. Clarenz et al. [CDR00] introduced anisotropic geometric diffusion to enhance features while smoothing. Ohtake et al. [OBB00] combined an inner fairness mechanism in their fairing process to increase the mesh regularity. Bajaj and Xu [BX03] smooth both surfaces and functions on surfaces, in a C^2 smooth function space defined by the limit of triangular subdivision surfaces (quartic Box splines).

Similar to surface diffusion using the Laplacian, a more general class of PDE based methods called ***flow surface techniques*** have been developed which simulate different kinds of flows on surfaces (see [WJE00] for references) using the equation $\partial_t x - v(x, t) = 0$, where $v(x, t)$ represents the instantaneous stationary velocity field.

Level set methods were also used in surface fairing and surface reconstruction; see [BCO00, BSCO00, CS99, MBWB02, OF00, WB98, ZOF01, ZOMK00]. In these methods, surfaces are formulated as iso-surfaces (level surfaces) of 3D functions, which are usually defined from the signed distance over Cartesian grids of a volume. An evolution PDE on the volume governs the behavior of the level surface. These level-set methods have several attractive features including, ease of implementation, arbitrary topology [BW01] and a growing body of theoretical results. Often, fine surface structures are not captured by level sets, although it is possible to use adaptive [PR99] and triangulated grids as well as Hermite data [JLSW02, KBSS01]. To reduce the computationally complexity, Bertalmio et al. [BCO00, BSCO00] solve the PDE in a narrow band for deforming vectorial functions on surfaces (with a fixed surface represented by the level surface).

Recently, surface diffusion flow has been used to solve the surface blending problem and free-form surface design problem. In [SK00], fair meshes with G^1 conditions are created in the special case where the meshes are assumed to have subdivision connectivity. In this work, local surface parametrization is still used to estimate the surface curvatures. A later paper [SK01] uses the same equation for smoothing meshes while satisfying G^1 boundary conditions. Outer fairness (the smoothness in the classical sense) and inner fairness (the regularity of the vertex distribution) criteria are used in their fairing process.

Another category of surface fairing research is based on utilizing optimization techniques. In this category, one constructs an optimization problem that minimizes certain objective functions [Gre94, HG00, MS92, Sap94, WW92], such as thin plate energy, membrane energy [KCVS98], total curvature [KHPS97, WW94], or sum of distances [Mal92]. Using local interpolation or fitting, or replacing differential operators with divided difference operators, the optimization problems are discretized to arrive at finite dimensional linear or nonlinear systems. Approximate solutions are then obtained by solving the constructed systems. In general, such an approach is quite computationally intensive.

53.6 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[Alf89]: Scattered data fitting and multivariate splines.

[Baj92, Baj97]: Summary of data fitting with implicit algebraic splines.

[BBB87]: Application of B-splines.

[Chu92, Dau92]: An introduction to wavelets.

[deB78]: An introduction to B-splines.

[dHR93]: An introduction to Box splines.

[DM83]: Scattered data fitting and multivariate splines.

[Far86, Far98]: Summary of the history of triangular Bernstein-Bézier patches.

[GL93]: An introduction to Knot manipulation techniques in splines.

[HL93]: An introduction to computer aided geometric design.

[Hol82]: Scattered data fitting and multivariate splines.

[Sch81, Sch94]: Scattered data fitting and multivariate splines.

[SDS96]: Application of wavelet representations.

[WW02]: Subdivision techniques.

RELATED CHAPTERS

[Chapter 25: Triangulations](#)

[Chapter 33: Computational real algebraic geometry](#)

[Chapter 49: Computer graphics](#)

[Chapter 56: Solid modeling](#)

REFERENCES

- [Alf89] P. Alfeld. Scattered data interpolation in three or more variables. In T. Lyche and L.L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 1–34, Academic Press, San Diego, 1989.
- [AMNS91] S. Auerbach, R.H.J. Gmelig Meyling, M. Neamtu, and H. Schaeben. Approximation and geometric modeling with simplex B-splines associated with irregular triangles. *Comput. Aided Geom. Design*, 8:67–87, 1991.
- [Baj92] C.L. Bajaj. Surface fitting with implicit algebraic surface patches. In H. Hagen, editor, *Topics in Surface Modeling*, pages 23–52. SIAM Publications, 1992.
- [Baj93] C.L. Bajaj. The emergence of algebraic curves and surfaces in geometric design.

- In R. Martin, editor, *Directions in Geometric Computing*, pages 1–29. Information Geometers, Winchester, 1993.
- [Baj97] C.L. Bajaj. Implicit surface patches. In J. Bloomenthal, editor, *Introduction to Implicit Surfaces*, pages 98–125, Morgan Kaufman, San Francisco, 1997.
- [BBB87] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, San Francisco, 1987.
- [BBCS99] F. Bernardini, C.L. Bajaj, J. Chen, and D. Schikore. Automatic reconstruction of 3D CAD models from digital scans. *Comput. Geom. Theory Appl.*, pages 327–369, 1999.
- [BBX95] C.L. Bajaj, F. Bernardini, and G. Xu. Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans. *Proc. ACM Conf. SIGGRAPH 95*, pages 109–118, 1995.
- [BCO00] M. Bertalmio, L.T. Cheng, and S.J. Osher. Variational problems and partial differential equations on implicit surfaces. CAM Report 00-23, UCLA, Math. Dept., 2000.
- [BCX94] C.L. Bajaj, J. Chen, and G. Xu. Smooth low degree approximations of polyhedra. *Comput. Sci. Tech. Rep.*, CSD-TR-94-002, Purdue Univ., 1994.
- [BCX95] C.L. Bajaj, J. Chen, and G. Xu. Modeling with cubic A-patches. *ACM Trans. Graph.*, 14:103–133, 1995.
- [BE95] C.L. Bajaj and S. Evans. Smooth multi-sided blends with A-patches. Presented at *Fourth SIAM Conf. Geometric Design*, 1995.
- [BHN99] C.L. Bajaj, R. Holt, and A. Netravali. Energy formulations for A-splines. *Comput. Aided Geom. Design*, 16:39–59, 1999.
- [BI92a] C.L. Bajaj and I. Ihm. Algebraic surface design with hermite interpolation. *ACM Trans. Graph.*, 11:61–91, 1992.
- [BI92b] C.L. Bajaj and I. Ihm. C^1 Smoothing of polyhedra with implicit algebraic splines. *Proc. ACM Conf. SIGGRAPH 92*, pages 79–88, 1992.
- [BIW93] C.L. Bajaj, I. Ihm, and J. Warren. Higher-order interpolation and least-squares approximation using implicit algebraic surfaces. *ACM Trans. Graph.*, 12:327–347, 1993.
- [BSCHO00] M. Bertalmio, G. Sapiro, L.T. Cheng, and S.J. Osher. A framework for solving surface partial differential equations for computer graphics applications. CAM Report 00-43, UCLA, Math. Dept., 2000.
- [BW89a] M.I.G. Bloor and M.J. Wilson. Generating blend surfaces using partial differential equations. *Comput. Aided Design*, 21:165–171, 1989.
- [BW89b] M.I.G. Bloor and M.J. Wilson. Generating N-sided patches with partial differential equations. In *Advances in Comput. Graph.*, pages 129–145. Springer-Verlag, Berlin, 1989.
- [BW90] M.I.G. Bloor and M.J. Wilson. Using partial differential equations to generate free-form surfaces. *Comput. Aided Design*, 22:221–234, 1990.
- [BW01] D. Breen and R. Whitaker. A level-set approach for the metamorphosis of solid models. *IEEE Trans. Visualization Comput. Graphics*, 7:173–192, 2001.
- [BX94] C.L. Bajaj and G. Xu. Rational spline approximations of real algebraic curves and surfaces. In H.P. Dikshit and C. Micchelli, editors, *Advances in Computational Mathematics*, pages 73–85. World Scientific, Singapore, 1994.

- [BX97] C.L. Bajaj and G. Xu. Modeling and visualization of C^1 and C^2 scattered function data on curved surfaces. *Comput. Aided Geom. Design*, 1997.
- [BX99] C.L. Bajaj and G. Xu. A-splines: Local interpolation and approximation using G^k -continuous piecewise real algebraic curves. *Comput. Aided Geom. Design*, 16:557–578, 1999.
- [BX01] C.L. Bajaj and G. Xu. Regular algebraic curve segments (III)—applications in interactive design and data fitting. *Comput. Aided Geom. Design*, 18:149–173, 2001.
- [BX03] C.L. Bajaj and G. Xu. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Trans. Graph.*, 22:4–32, 2003.
- [CDR00] U. Clarenz, U. Diewald, and M. Rumpf. Anisotropic geometric diffusion in surface processing. In *Proc. IEEE Visualization*, pages 397–505, Salt Lake City, Utah, 2000.
- [CF83] H. Chiyokura and K. Fumihiko. Design of solids with free form surfaces. *Comput. Graph.*, 17:289–298, 1983.
- [Chu92] C. Chiu. *An Introduction to Wavelets*. Academic Press, Boston, 1992.
- [CS99] D.L. Chopp and J.A. Sethian. Motion by intrinsic laplacian of curvature. *Interfaces and Free Boundaries*, 1:1–18, 1999.
- [CX01] C.L. Bajaj and G. Xu. Smooth shell construction with mixed prism fat surfaces. In *Geometric Modeling*, pages 19–36. Springer-Verlag, Computing Supplementum 14, 2001.
- [Dah89] W. Dahmen. Smooth piecewise quadratic surfaces. In T. Lyche and L.L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 181–193. Academic Press, Boston, 1989.
- [Dau92] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.
- [dC92] M. do Carmo. *Riemannian Geometry*. Birkhäuser, Boston, 1992.
- [DD02] K. Deckelnick and G. Dziuk. A fully discrete numerical scheme for weighted mean curvature flow. *Numerische Mathematik*, 91:423–452, 2002.
- [deB78] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.
- [dHR93] C. de Boor, K. Hollig, and S. Riemenschneider. *Box Splines*. Springer-Verlag, New York, 1993.
- [DM83] W. Dahmen and C. Micchelli. Recent progress in multivariate splines. In L. Shumaker, C. Chui, and J. Word, editors, *Approximation Theory IV*, pages 27–121. Academic Press, 1983.
- [DMS92] W. Dahmen, C. Micchelli, and H.-P. Seidel. Blossoming begets B-spline bases built better by B-patches. *Mathematics of Computation*, 59:97–115, 1992.
- [DMSB99] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. *Proc. ACM Conf. SIGGRAPH 99*, pages 317–324, 1999.
- [DMSB00] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr. Discrete differential-geometry operators in nD , <http://www.multires.caltech.edu/pubs/>, 2000.
- [DTS93] W. Dahmen and T-M. Thamm-Schaar. Cubicoids: Modeling and visualization. *Comput. Aided Geom. Design*, 10:89–108, 1993.
- [EDD⁺95] M. Eck, T.D. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proc. ACM Conf. SIGGRAPH 95*, pages 173–180, 1995.
- [Far86] G. Farin. Triangular Bernstein-Bézier Patches. *Comput. Aided Geom. Design*, 3:83–127, 1986.

- [Far98] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, Boston, 1998.
- [FB88] D. Forsey and R. Bartels. Hierarchical B-spline refinement. *Proc. ACM Conf. SIGGRAPH 88*, pages 205–212, 1988.
- [FB95] D. Forsey and R. Bartels. Surface fitting with hierarchical splines. *ACM Trans. Graph.*, 14:134–161, 1995.
- [for95] D. Forsey. Surface fitting with hierarchical splines. *ACM Trans. Graph.*, 14:134–161, 1995.
- [FS86] P. Fong and H.-P. Seidel. Control points for multivariate B-spline surfaces over arbitrary triangulations. *Computer Graphics Forum*, 10:309–317, 1986.
- [FS92] P. Fong and H.-P. Seidel. An implementation of multivariate B-spline surfaces over arbitrary triangulations. In *Proc. Graphics Interface*, pages 1–10, Vancouver, B.C., 1992.
- [FW93] D. Forsey and L. Wang. Multi-resolution surface approximation for animation. In *Proc. Graphics Interface*, pages 192–199, Toronto, May 1993.
- [GH87] J.A. Gregory and J. Hahn. Geometric continuity and convex combination patches. *Comput. Aided Geom. Design*, 4:79–89, 1987.
- [GL93] R.N. Goldman and T. Lyche. *Knot Insertion and Deletion Algorithms for B-spline Curves and Surfaces*. SIAM, 1993.
- [Gre83] J.A. Gregory. C^1 rectangular and non-rectangular surface patches. In R.E. Barnhill and W. Boehm, editors, *Comput. Aided Geom. Design*, pages 25–33. North-Holland, Amsterdam, 1983.
- [Gre94] G. Greiner. Variational design and fairing of spline surface. *Comput. Graph. Forum*, 13:143–154, 1994.
- [Guo91] B. Guo. Surface generation using implicit cubics. In N.M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pages 485–530. Springer-Verlag, Tokyo, 1991.
- [Guo93] B. Guo. Non-splitting macro patches for implicit cubic spline surfaces. *Comput. Graph. Forum*, 12:434–445, 1993.
- [GW91] T. Garrity and J. Warren. Geometric continuity. *Comput. Aided Geom. Design*, 8:51–65, 1991.
- [Hau87] D. Haumann. Modeling the physical behavior of flexible objects. ACM SIGGRAPH Course Notes #17, 1987.
- [HF84] M. Hosaka and K. Fumihiko. Non-four-sided patch expressions with control points. *Comput. Aided Geom. Design*, 1:75–86, 1984.
- [HG00] A. Hubeli and M. Gross. Fairing of non-manifolds for visualization. In *Proc. IEEE Visualization*, pages 407–414, Salt Lake City, 2000.
- [HL93] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A.K. Peters, Wellesley, 1993. Translated by L.L. Schumaker.
- [Hol82] K. Hollig. Multivariate splines. *SIAM J. Numer. Anal.*, 19:1013–1031, 1982.
- [JLSW02] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. *Proc. ACM Conf. SIGGRAPH 02*, pages 339–346, 2002.
- [KBSS01] L. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature sensitive surface extraction from volume data. *Proc. ACM Conf. SIGGRAPH 01*, pages 51–66, 2001.

- [KCVS98] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. *Proc. ACM Conf. SIGGRAPH 98*, pages 105–114, 1998.
- [KHPS97] L. Kobbelt, T. Hesse, H. Prautzsch, and K. Schweizerhof. Iterative mesh generation for FE-computation on free form surfaces. *Engng. Comput.*, 14:806–820, 1997.
- [Kob96] L. Kobbelt. Discrete fairing. In T. Goodman and Ralph Martin, editors, *The Mathematics of Surfaces VII*, pages 101–129. Information Geometers, 1996.
- [LD89] C. Loop and T.D. DeRose. A multisided generalization of Bézier surfaces. *ACM Trans. Graph.*, 8:205–234, 1989.
- [LD90] C. Loop and T.D. DeRose. Generalized B-spline surfaces of arbitrary topology. *Comput. Graph.*, 24:347–356, 1990.
- [Lod92] S. Lodha. *Surface approximation by low degree patches with multiple representations*. Ph.D. thesis, Purdue Univ., West Lafayette, 1992.
- [Mal92] J.L. Mallet. Discrete smooth interpolation in geometric modelling. *Comput. Aided Design*, 24:178–191, 1992.
- [MBWB02] K. Museth, D. Breen, R. Whitaker, and A.H. Barr. Level set surface editing operators. *Proc. ACM Conf. SIGGRAPH 88*, pages 330–338, 2002.
- [MLL⁺92] S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T.D. DeRose, and K. Sloan. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curve and Surface Modeling*, pages 145–172. SIAM, Philadelphia, 1992.
- [MS92] H. Moreton and C.H. Séquin Functional optimization for fair surface design. *ACM Comput. Graph.*, pages 409–420, 1992.
- [MW91] D. Moore and J. Warren. Approximation of dense scattered data using algebraic surfaces. In *Proc. 24th Hawaii Internat. Conf. System Sci.*, pages 681–690, Kauai, Hawaii, 1991.
- [OBB00] Y. Ohtake, A.G. Belyaev, and I.A. Bogaevski. Polyhedral surface smoothing with simultaneous mesh regularization. In *Proc. Geom. Modeling Processing*, pages 229–237, 2000.
- [OF00] S.J. Osher and R.P. Fedkiw. Level set methods. CAM Report 00-07, UCLA, Math. Dept., 2000.
- [PB88] J. Platt and A.H. Barr. Constraint methods for flexible models. *Proc. ACM Conf. SIGGRAPH 88*, pages 279–288, 1988.
- [Pet90a] J. Peters. Local cubic and bicubic C^1 surface interpolation with linearly varying boundary normal. *Comput. Aided Geom. Design*, 7:499–516, 1990.
- [Pet90b] J. Peters. Smooth mesh interpolation with cubic patches. *Comput. Aided Design*, 22:109–120, 1990.
- [Pet91] J. Peters. Smooth interpolation of a mesh of curves. *Constructive Approx.*, 7:221–246, 1991.
- [Pet93] J. Peters. Smooth free-form surfaces over irregular meshes generalizing quadratic splines. *Comput. Aided Geom. Design*, 10:347–361, 1993.
- [Pet02] J. Peters. C^2 free-form surfaces of degree (3,5). *Comput. Aided Geom. Design*, 19(2), 2002.
- [PK89] N.M. Patrikalakis and G.A. Kriegis. Representation of piecewise continuous algebraic surfaces in terms of b-splines. *Visual Comput.*, 5:360–374, 1989.

- [PM87] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *IEEE Comput. Soc. Workshop Comput. Vision*, 1987.
- [PR99] T. Preufer and M. Rumpf. An adaptive finite element method for large scale image processing. In *Scale-Space Theories in Computer Vision*, pages 232–234, 1999.
- [PTBK87] J. Platt, D. Terzopoulos, A.H. Barr, K. Fleischer. Elastically deformable models. *Comput. Graph.*, 21:205–214, 1987.
- [Rei92] U. Reif. A unified approach to subdivision algorithms. Tech. Rep. 92-16, Mathematisches Institut A, Universität Stuttgart, 1992.
- [Sap94] N. Sapidis. *Designing Fair Curves and Surfaces*. SIAM, Philadelphia, 1994.
- [Sar87] R.F. Sarraga. G^1 interpolation of generally unrestricted cubic Bézier curves. *Comput. Aided Geom. Design*, 4:23–39, 1987.
- [Sch81] L.L. Schumaker. *Spline Functions: Basic Theory*. Wiley, New York, 1981.
- [Sch94] L.L. Schumaker. Applications of multivariate splines. In *Math. Comput., 1943–1993: A Half-century of Computations Mathematics, Proc. Symposia in Applied Mathematics*, Volume 48. Amer. Math. Soc., Providence, 1994.
- [SDS96] E.J. Stollnitz, T.D. DeRose, and D. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, San Francisco, 1996.
- [Sed85] T.W. Sederberg. Piecewise algebraic patches. *Comput. Aided Geom. Design*, 2:53–59, 1985.
- [Sei89] H.-P. Seidel. A new multiaffine approach to B-splines. *Comput. Aided Geom. Design*, 6:23–32, 1989.
- [SK00] R. Schneider and L. Kobbelt. Generating fair meshes with G^1 boundary conditions. In *Geometric Modeling Processing*, pages 251–261, 2000.
- [SK01] R. Schneider and L. Kobbelt. Geometric fairing of triangular meshes for free-form surface design, *Comput. Aided Geom. Design*, 18:359–379, 2001.
- [Tau95] G. Taubin. A signal processing approach to fair surface design. In *Proc. ACM Conf. SIGGRAPH 95*, pages 351–358, 1995.
- [TC90] J. Thingvold and E. Cohen. Physical modeling with B-spline surfaces for interactive design and animation. *Comput. Graph.*, 24:129–137, 1990.
- [TF88] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *Proc. ACM Conf. SIGGRAPH 88*, pages 269–278, 1988.
- [TQ94] D. Terzopoulos and H. Qin. Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Trans. Graph.*, 13:103–136, 1994.
- [War92] J. Warren. Creating multisided rational bezier surfaces using base points. *ACM Trans. Graph.*, 11:127–139, 1992.
- [WB98] R. Whitaker and D. Breen. Level set models for the deformation of solid objects. In *Proc. 3rd Internat. Eurographics Workshop Implicit Surfaces*, pages 19–35, June 1998.
- [Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing*. B.G. Teubner, Stuttgart, 1998.
- [WFB87] A. Witkin, K. Fleischer, and A.H. Barr. Energy constraints on parameterized models. In *Proc. ACM Conf. SIGGRAPH 87*, pages 225–232, 1987.
- [WJE00] R. Westermann, C. Johnson, and T. Ertl. A level-set method for flow visualization. In *Proc. IEEE Visualization*, pages 147–154, Salt Lake City, 2000.

- [WW92] W. Welch and A. Witkin. Variational surface modeling. *Comput. Graph.*, 26:157–166, 1992.
- [WW94] W. Welch and A. Witkin. Free-form shape design using triangulated surfaces. In *Proc. ACM Conf. SIGGRAPH 94*, pages 247–256, 1994.
- [WW02] J. Warren and H. Weimer. *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann, San Francisco, 2002.
- [XB00] G. Xu and C.L. Bajaj. Regular algebraic curve segments (I)—Definitions and characteristics. *Comput. Aided Geom. Design*, 17:485–501, 2000.
- [XBE01] G. Xu, C.L. Bajaj, and S. Evans. C^1 modeling with hybrid multiple-sided A-patches. *Special issue on Surface and Volume Reconstructions in the Internat. Journal Found. Comput. Sci.*, 13:261–284, 2001.
- [ZOF01] H.K. Zhao, S.J. Osher, and R.P. Fedkiw. Fast surface reconstruction using the level set method. CAM Report 01-01, UCLA, Math. Dept., 2001.
- [ZOMK00] H.K. Zhao, S.J. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Comput. Vision Graph. Image Understanding*, 80:295–319, 2000.

54 SURFACE SIMPLIFICATION AND 3D GEOMETRY COMPRESSION

Jarek Rossignac

INTRODUCTION

Central to 3D modeling, graphics, and animation, *triangle meshes* are used in Computer Aided Design, Visualization, Graphics, and video games to represent polyhedra, control meshes of subdivision surfaces, or tessellations of parametric surfaces or level sets. A triangle mesh that accurately approximates the surface of a complex 3D shape may contain millions of triangles. This chapter discusses techniques for reducing the delays in transmitting it over the Internet. The *connectivity*, which typically dominates the storage cost of uncompressed meshes, may be compressed down to about one bit per triangle by compactly encoding the parameters of a triangle-graph construction process and by transmitting the vertices in the order in which they are used by this process. Vertex coordinates, i.e., the *geometry*, may often be compressed to less than 5 bits each through quantization, prediction, and entropy coding. Thus, *compression* reduces storage of triangle meshes to about a byte per triangle. When necessary, file size may be further reduced through *simplification*, which collapses edges or merges clusters of neighboring vertices to decrease the total triangle count. The application may select the appropriate level-of-detail; trading fidelity for transmission speed. In applications where preserving the exact geometry and connectivity of the mesh is not essential, the triangulated surface may be *re-sampled* to produce a mesh with a more regular connectivity and with vertices that are constrained to, each, lie on a specific curve, and thus may be fully specified by a single parameter. Re-sampling may improve compression significantly, without introducing noticeable distortions. Furthermore, when the accuracy of a simplified or re-sampled model received by a client is insufficient, compressed upgrades may be downloaded as needed to *refine* the model in a *progressive* fashion.

Due to space limitations, we focus primarily on triangle meshes that are homeomorphic to triangulation of a sphere. Strategies for extending the compression, simplification, and refinement techniques to more general meshes, which include polygonal meshes, manifold meshes with handles and boundaries, or nonmanifold models; to tetrahedral, higher dimensional, or animated meshes; and to models with texture or property maps, are discussed elsewhere.

GLOSSARY

Mesh: A set of triangles homeomorphic to the triangulation of a sphere.

Geometry (of a mesh): The positions of the vertices (possibly described by 3 coordinates each).

Incidence: The definition of the triangles of the mesh, each as 3 vertex Ids.

Connectivity: Incidence, combined with adjacency and order relations, which may be derived from the incidence.

Connectivity compression: Encoding the incidence, while ignoring the geometry.

Geometry compression: Encoding of the vertex locations, while assuming that the relevant connectivity information will be available to the decoder.

Simplification: The process of merging the vertices of a mesh to generate a new mesh that approximates the original one, but comprises fewer triangles.

Level-of-Detail (LOD): One of a series of increasingly simplified models that approximate an original shape by trading fidelity for a decrease in triangle count.

Re-sampling: The approximation of an original mesh by one with a new connectivity and a new set of vertices, selected to minimize error and maximize compression.

Progressive transmission: A protocol in which the client receives the lowest LOD, possibly followed by upgrades, if and when needed.

Single-rate compression: A compressed representation that does not support progressive transmission.

54.1 SIMPLE TRIANGLE MESHES

In this section, we introduce the terminology, properties, data structures, and operators used in this chapter. We assume that the mesh is simple, i.e., homeomorphic to the triangulation of a sphere.

GLOSSARY

Vertex and triangle count: A mesh with v vertices and t triangles satisfies $t = 2v - 4$.

Triangle: A node of the connectivity graph representing a closed point-set that is the convex hull of three noncollinear vertices.

Surface (of a mesh): The union of the point-sets of its triangles.

Edge: A link in the connectivity graph representing a relatively open line segment joining two vertices of a mesh, but not containing them.

Face: The relative interior of a triangle, i.e., the triangle minus the union of the point-sets of its edges and vertices.

Corner: A corner c associates a vertex $c.v$ with one of its incident triangles $c.t$.

Cascading corner operators: The notation $c.n.v$ stands for $(c.n).v$.

Next and previous corners: The two corners in $c.t$ other than c are denoted $c.n$ and $c.p$.

Incidence table: An array V of $3t$ entries, where $V[c]$ is denoted $c.v$ and contains a vertex Id. Entries for $c.p$, c , and $c.n$ are consecutive in V . Therefore, $c.t$ is the integer division $c.t \text{ div } 3$; $c.n$ is $c-2$, when $c \bmod 3$ is 2, and $c+1$ otherwise; and $c.p$ is $c.n.n$.

Opposite corner: Two corners c and b are opposite when $b.n.v=c.p.v$ and $b.p.v=c.n.v$. The opposite corner of c , denoted $c.o$, is a corner Id stored as the entry $O[c]$ in the O table.

Orientation: The orientation of a triangle $c.t$ is defined by the choice of $c.n$ amongst the other two corners of $c.t$. The mesh is globally oriented so that $c.n.v=c.o.p.v$ for all corners c .

Corner Table: The two arrays, V and O .

Left and right neighbors of a corner: For convenience, we define $c.l$ to be $c.n.o$ and $c.r$ to be $c.o.p$.

Vertex-Spanning Tree (VST): A subset of cycle-free edges connecting all of the vertices.

Cut-Edges: The edges contained in a given VST.

Cut: The union of the cut-edges with all of the vertices.

Hinge-Edges: Edges that are not cut edges.

Web: Union of all the faces and of all the hinge-edges.

Triangle-Spanning-Tree (TST): Rooted graph with hinge-edge links and triangle nodes.

54.1.1 TERMINOLOGY AND PROPERTIES

Consider a mesh with v vertices, e edges, and t triangles.

GEOMETRY AND INCIDENCE

A mesh is usually defined in terms of a set of *vertices* and its triangle-vertex *incidence*. The vertex description comprises *geometry* (three coordinates per vertex) and optionally *photometry* (surface normals, vertex colors, or texture coordinates), not discussed in this chapter. *Incidence* defines each triangle by three integer references identifying its vertices. Simple and regular data structures, such as the *Corner Table* described below, and formats, such as the *indexed face set* of VRML [VRML97] may be used for representing geometry and incidence.

An *uncompressed representation* uses $12v$ bytes for geometry and $12t$ bytes for incidence. Given that $t \approx 2v$, the total cost is $144t$ bits, of which two thirds are used for incidence.

CUT, WEB, AND SPANNING TREES

The vertices of a mesh that is homeomorphic to a sphere may always be placed on the plane so that the edges and vertices are mutually disjoint. The union of these edges and vertices partition the rest of the plane into *cells* that are each bounded by 3 edges and 3 vertices. One of the cells is unbounded. This mapping forms a *planar graph* of the mesh.

A *Vertex-Spanning Tree* (VST) of a triangle mesh is a subset of its edges, selected so that their union with all of the vertices forms a tree (connected cycle-free graph). We assume henceforth that a particular VST has been chosen for the mesh. The edges that it includes are called the *cut-edges*. The union of the cut-edges

with all the vertices is called a *cut*. Because the VST is a tree, there are $v - 1$ cut-edges.

We use the term *web* to denote the difference between the surface and the point-set of its cut. Edges that are not cut-edges are called *hinge-edges*. The web is composed of all of the faces and of all of the hinge-edges. Removing the loopless cut from the surface will leave a web that is a simply connected (relatively open), triangulated 2D point-set in \mathbb{R}^3 . The web may be represented by an acyclic graph, whose nodes correspond to faces and whose links correspond to hinge edges. Thus there are $t - 1$ hinge edges.

Note that by selecting a leaf of this graph as root and orienting the links, we can always turn it into a binary tree, which we call the *Triangle-Spanning-Tree* (TST), and which is a spanning tree of the dual of the planar graph.

EULER EQUATION

Because an edge is either hinge or cut, the total number e of edges is $v - 1 + t - 1$. Each triangle uses 3 edges and each edge is used by 2 triangles. Thus the number e of edges is also equal to $3t/2$. Combining these two equations yields $t = 2v - 4$, a special case of Euler's formula $f - e + v = 2$.

54.1.2 CORNER TABLE REPRESENTATION

We present a simple data structure for meshes. We call it the Corner Table [RSS03] and use it to explain the details of connectivity compression.

DATA STRUCTURE AND OPERATORS

The corner table is composed of three arrays: G, V, and O.

The geometry is stored in the coordinate table, G, where $G[v]$, denoted v.g., contains the triplet of the coordinates of vertex number v . The order in which the vertices are listed in G is arbitrary, but defines the Id (integer) associated with each vertex.

Triangle-vertex incidence defines each triangle by the three Ids of its vertices, which are stored as consecutive entries in the *V-table*. Note that each one of the $3t$ entries in V represents a *corner* (association of a triangle with one of its vertices). Let c be such a corner. Let $c.t$ denote its triangle and $c.v$ its vertex. Recall that $c.v$ and $c.t$ are integers in $[0, v - 1]$ and $[0, t - 1]$, respectively. Let $c.p$ and $c.n$ refer to the *previous* and *next* corner in the cyclic order of vertices around $c.t$.

Although G and V suffice to completely specify the triangles and thus the surface they represent, they do not offer direct access to a neighboring triangle or

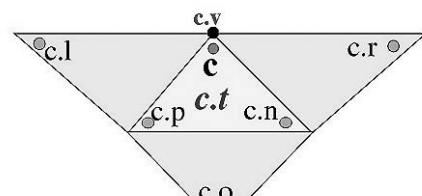


FIGURE 54.1.1

Corner operators for traversing and processing a corner table representation of a triangle mesh.

vertex. We chose to use the *opposite* corner Id, c.o, cached in the *O table* to accelerate mesh traversal from one triangle to its neighbors. For convenience, we introduce the operators c.l and c.r, which return the *left* and *right* neighbors of c (Figure 54.1.1).

Given a corner c, the choice of which of the other two vertices of c.t is c.n defines one of two possible *orientations* for c.t. We assume that all triangles have been consistently *oriented*, so that c.n.v=c.o.p.v for all corners c.

MESH TRAVERSAL ON A CORNER TABLE

Assume that the Boolean c.t.m is set to TRUE when the triangle c.t has been visited. The procedure: visit(c) { if !c.t.m then { c.t.m = TRUE; visit(c.r); visit(c.l) } } will visit all the triangles in a depth-first order of a TST.

THE COMPUTATION OF THE O-TABLE FROM V

Given the V-table, the entries in O may be computed by a double loop over corners, identifying the opposite corners. A faster approach sorts the triplets $\{\min(c.n.v, c.p.v), \max(c.n.v, c.p.v), c\}$ into bins. All entries in a bin have the same first record: $\min(c.n.v, c.p.v)$, an integer in $[0, v - 1]$. There are rarely more than 20 entries in a bin. Then, we sort the entries in each bin by the second record: $\max(c.n.v, c.p.v)$. Now, pairs with identical first two records are consecutive and correspond to opposite corners, identified by the third record in each triplet. Thus, if a sorted bin contains consecutive entries (a,b,c) and (a,b,d), we set c.o:=d and d.o:=c.

54.1.3 REDUCTIONS OF THE TRANSMISSION COST

Because it can be easily recreated, the O-table needs not be transmitted. Furthermore, the $31 - \log_2 v$ leading zeros of each entry in the V table need not be transmitted. Thus, a compact, but uncompressed representation of a triangle mesh requires $48t$ bits for the coordinates and $3t \log_2 t - 3t$ bits for the V-table.

54.2 GEOMETRY COMPRESSION

The compression of vertex coordinates usually combines three steps: quantization, prediction, and statistical coding. The combined three stages usually yield a 7-to-1 compression of geometry.

GLOSSARY

Normalization: Linear transformation of coordinates so that their range spans $[0, 2^B - 1]$.

Quantization: Rounding of each normalized vertex coordinate to the nearest integer.

Prediction: The prediction of the quantized location of a new vertex from its neighbors.

Parallelogram prediction: Using $c.n.v.g + c.p.v.g - c.v.g$ as a prediction for $c.o.v.g$.

Residues: The differences between the actual quantized coordinates and their prediction.

Statistical coding: A bit-efficient encoding of the residues, exploiting the bound on their magnitude and the statistics of the nonuniform distribution of their frequencies.

54.2.1 QUANTIZATION

The common use of *FLOATS* for vertex coordinates has three major drawbacks. First, the range of values that can be represented may significantly exceed the actual range covered by the vertex coordinates, and thus bit-combinations are “wasted on” impossible coordinates. Second, the distance between two consecutive coordinate values, i.e., the **round-off error**, is not uniformly distributed, but depends on the distance to the origin, thus providing more accuracy for some portion of the model than for others. Third, the **resolution** of the representation may significantly exceed what is required by the application.

Quantization addresses these three drawbacks. It truncates the vertex coordinates to a fixed accuracy, thus, by itself, reducing storage size while preserving an acceptable geometric accuracy. It starts with a **normalization** process, which computes a tight, axis-aligned bounding box. Then the longest dimension of the box is divided into 2^B segments or equal size, s . The other dimensions are also divided into cells of size s , possibly enlarging the box to have uniform cells. Thus, the normalization process divides the (enlarged) bounding box into cubic cells of size s . The vertices that fall inside a given cells are snapped to the cell center. Thus, the corresponding error is bounded by half of the diagonal of the cell. The number of bits required to encode each coordinate is less than B . Choosing $B = 12$ ensures a sufficient geometric fidelity for most applications and most models. Thus, this lossy quantization step, by itself, reduces the storage cost of geometry from $96v$ bits to $36v$ bits.

54.2.2 PREDICTION

The next and most crucial geometry compression step involves **vertex prediction**. Both the encoder and the decoder use the same predictor. Thus, only the **residues** between the predicted and the correct coordinates need to be transmitted. The **coherence** between neighboring vertices in meshes of finely sampled smooth surfaces limits the magnitude of the residues. For example, if the magnitude of the largest residue is less than 63 (quantized units), then the total cost for geometry drops to $21v$ bits (a sign bit and a 6-bit magnitude per coordinate). We describe below several predictors.

Because most edges are short with respect to the size of the model, adjacent vertices are in general close to each other and the differences between their coordinates are small. Thus a new vertex may be predicted by a previously transmitted **neighbor** [Dee95]. Instead of using a single neighbor, when vertices are transmitted in VST top-down order, a linear combination of the four ancestors in the VST

may be used [TR98]. The four coefficients of this combination are computed to minimize the magnitude of the residues over the entire mesh and transmitted as part of the compressed stream.

The most popular predictor for *single-rate compression* is based on the *parallelogram* construction [TG89]. Assume that the vertices of c.t have been decoded. We predict c.o.v.g using c.n.v.g + c.p.v.g - c.v.g. The parallelogram prediction may sometimes be improved by predicting the angle between c.t and c.o.t from the angles of previously encountered triangles.

54.2.3 STATISTICAL CODING

Some of the residues may be large. Thus, good prediction by itself may not lead to compression. For example, if the coordinates have been quantized to B -bit integers, some of the coordinates of the corrective vector, $c.o.v.g - c.n.v.g - c.p.v.g + c.v.g$ may require $B + 2$ bits of storage. Thus, parallelogram prediction may expand storage rather than compress it. However, the distribution of the residues is usually biased toward zero, which makes them suitable for statistical compression [Sal00].

For instance, assume that all residues are integers in $[-63, +63]$. Furthermore suppose that in 99% of the cases, the most significant bit of the magnitude of the residue is 0. The entropy of this bit is $-0.99\log_2(0.99) - 0.01\log_2(0.01)$, which amounts to 0.05 bit per coordinate. Arithmetic coding compression may be used to reduce the total storage size of these most significant bits close to its theoretical *entropy*. Furthermore, if in 95% of the cases the second most-significant bit of the residue magnitude is 0, its cost per coordinate may be reduced to 0.15 bits. If the third and fourth bits have respective probabilities of 90% and 80% of being zero, their respective costs are 0.50 and 0.72 bits per coordinates. Even if we assume no statistical compression of the sign and of the two least significant bits, the total cost per coordinate will be 4.42 bits per coordinate, or equivalently $13.3v$ bits.

54.3 CONNECTIVITY COMPRESSION

As discussed above, typically geometry may be encoded with $< 14t$ bits, i.e., $7t$ bits, provided that connectivity information is available during geometry decompression to locate previously decoded neighbors of each vertex along the surface. This section presents techniques for compressing the connectivity information from $3t(2v - 4)\log_2 v$ bits to bt bits, where b is guaranteed never to exceed 1.80 and in practice is usually close to 1.0. As a result, many meshes may be encoded with a total of less than $8t$ bits ($7t$ for geometry, $1t$ for connectivity) with a small error due to quantization.

Two observations could lead to misjudgement of the importance of incidence compression. (1) Some applications [ABK98] produce unorganized clouds of 3D point samples for which the incidence is derived algorithmically, and thus needs not be transmitted. (2) Recently developed graphic techniques for producing images of 3D surfaces directly from clouds of unstructured 3D samples [LW85] eliminate altogether the need for computing and transmitting the incidence information. Thus, one may conclude that it is not only unnecessary to transmit the incidence, but also unadvisable, considering that uncompressed, it is more expensive than geometry.

However, capturing and transmitting the incidence information has several important benefits. (1) Its reconstruction is a computationally expensive and delicate process; thus it is usually faster to transmit and decompress the incidence than to recompute it. (2) To correctly render a cloud of unstructured samples as a continuous surface, the samples must be uniformly distributed over the surface and the density of their distribution must ensure that the distance between neighboring samples along the surface is smaller than the size of the smallest feature. Incidence information permits significant reduction in the density of samples in low-curvature portions of the surface. Thus, the samples in nearly flat regions need not be transmitted, since their approximation will be reproduced automatically by the graphics rasterization at rendering time. (3) The most effective ***geometry compression*** techniques are based on the prediction of the location of the next vertex from the locations of its previously decompressed neighbors. Transmitting information describing who the previously decoded neighbors of each vertex are and how they are organized around a new vertex is equivalent to transmitting the incidence graph. (4) The incidence may be compressed to about a bit per triangle and thus the overhead of transmitting it is negligible when compared to the savings in geometry storage to which it leads.

GLOSSARY

Border edge: An edge of the recovered portion of the triangle mesh during decompression that has, so far, only one incident triangle. The natural orientation of a border edge is consistent with the orientation of the incident triangle.

Hole: A maximally connected component of the relative interior of the not-yet-recovered portion of the mesh.

Loop: A chain of border edges forming a closed manifold boundary of a hole.

Gate: The border edge where the next new triangle will be attached during decompression.

Active loop: The loop containing the gate.

Tip corner: The corner c of the new triangle, where $c.n.v$ and $c.p.v$ bound the gate.

Tip vertex: The vertex, $c.v$, associated with the tip corner, c .

Right edge: The edge joining $c.v$ and $c.n.v$, where c is the tip corner.

Left edge: The edge joining $c.p.v$ and $c.v$, where c is the tip corner.

Offset: The number of vertices in the active loop from the gate to the tip vertex of an S-triangle.

clers string: Connectivity encoding as a sequence of labels in C,L,E,R,S.

Valence (or degree): The number of triangles incident upon a given vertex.

54.3.1 EDGEBREAKER

Instead of retracing the chronological evolution of research results in the field of single-rate incidence compression, we first describe in detail Edgebreaker [Ros99],

one of the simplest and most effective single-rate compression approaches. The source code for Edgebreaker is publicly available.¹

Then, we briefly review several variants and other approaches, using Edgebreaker's terminology to characterize their differences and respective advantages.

The Edgebreaker compression visits the triangles in a spiraling (depth-first) TST order and generates the *clers string* of labels, one label per triangle, which indicate to the decompression how the connectivity of the mesh can be rebuilt by attaching new triangles to previously reconstructed ones. We first describe the Edgebreaker decompression process.

EDGEBREAKER DECOMPRESSION

During decompression, the reconstructed portion of the mesh is a surface with one or more holes, bounded by *loops* of oriented *border edges*. Each decompression step attaches a new triangle to a border edge, called the *gate*, in the active loop (Figure 54.3.1).

The labels in the *clers* string define where the tip of the new triangle is. Edgebreaker uses only five labels: C, L, E, R, and S. Label C indicates that the new triangle will have as tip a new vertex. We say that this triangle is of type C. Note that the three vertices of the triangle incident upon the gate have been previously decoded and may be used in a parallelogram prediction of the new vertex. The numbering of the vertices and hence their order in the G-table of the reconstructed mesh reflects the order in which the vertices are instantiated as tips of C-triangles.

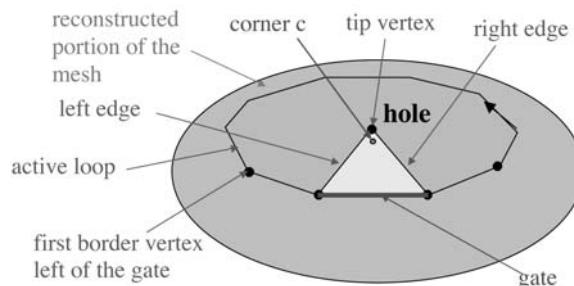
L indicates that the tip vertex is the first border vertex on the left of the gate in the current loop (Figure 54.3.1). R indicates that the tip is the first border vertex on the right of the gate in the current loop. E indicates that the new triangle will close a hole bounded by the current loop, which therefore must have only three vertices. S indicates that the tip of the new triangle is elsewhere in the current loop. An S-triangle splits the current loop in two loops, as well as splitting the hole bounded by that loop into two holes. The one bounded by the right edge of the new triangle is the *right hole*. The other one is the *left hole* (Figure 54.3.2).

After the new triangle is attached, the gate is moved to the right edge of the new triangle for cases C and L. It is moved to the left edge for case R. When an

¹www.gvu.gatech.edu/~jarek/edgebreaker/eb

FIGURE 54.3.1

The terminology used to describe the Edgebreaker decompression.



S-triangle is attached, the gate is first moved to the right edge and the right hole is filled through a recursive call to decompression. Then the gate is moved to the left edge and the process resumes as if the S-triangle had been an R-triangle.

The reconstruction of the connectivity graph from the *clers* string is trivial, except for the S-triangles. Indeed, a C-triangle is attached to the gate and a new vertex is used for its tip. The tips of the L, R, and E triangles are known. The Id of the tip of each S-triangle could be encoded using $\log_2(k)$ bits, where k is the number of previously decoded vertices. A more economical approach is to encode an *offset* o indicating the number of vertices that separate the gate from the tip in the current loop (Figure 54.3.2). Because the current loop may include a large fraction of the vertices, one may still need up to $\log_2(k)$ bits to encode the offset. Although one may prove that the total cost of encoding the offsets is linear in the number of triangles [Gum99], the encoding of the offsets constitutes a significant fraction of the total size of the compressed connectivity. Hence, several authors strived to minimize the number of offsets [AD01b].

The author has observed [Ros99] that the *offsets need not be transmitted* at all, because they can be recomputed by the decompression algorithm from the *clers* string. The observation is based on the fact that the attachment of a triangle of each type changes the number of edges in the current loop by specific amounts (Figure 54.3.2). C increments the edge-count. R and L decrement it. E removes a loop of three edges and thus decreases the edge-count by 3. S splits the current loop in two parts, but if we count the edges in both parts, it increments the total edge count. Each S label starts a recursive call that fills in the hole bounded by the right loop and terminates with the corresponding E label. Thus ***S*** and ***E*** labels work as pairs of *parentheses*. Combining all these observations, we can compute the offset by identifying the substring of the *clers* string between an S and the corresponding E, and by summing the edge-count changes for each label in that substring. To avoid the multiple traversals of the *clers* string, all offsets may be precomputed by reading the *clers* string once and using a stack for each S.

The elegant ***Spirale Reversi*** approach [IS99] for decompressing *clers* strings that have been created by the Edgebreaker compression avoids this preprocessing by reading the *clers* string backwards and building the triangle mesh in reverse order.

A third approach, ***Wrap&Zip*** [RS99], also avoids the preprocessing and directly builds a Corner Table as it reads the *clers* string. It does not require maintaining a linked list of border vertices. For each symbol, as a new triangle is attached to the gate, Wrap&Zip fills in the known entries to the V and O-tables. Specifically, it fills in c.o for the tip corner c of the new triangle and for its opposite, c.o. It assigns vertex Ids for the tips of C-triangles as they are created, by simply incrementing a vertex Id counter. It defers assigning the Ids of other vertices until a Zip process matches them with vertices that already have an Id. Thus, it produces a *web*, as defined earlier. The border edges of the web must be matched into pairs. The correct matching could be specified by encoding the structure of the cut [Tur84] [TR98]. However, as discovered in [RS99], the information may be trivially extracted from the *clers* string by orienting the border edges of the web as shown in Figure 54.3.3. Note that these border-edge orientations are consistent with an upward orientation of the cut-edges toward the root of the VST.

The zipping part of Wrap&Zip matches pairs of adjacent border edges that are oriented away from their shared vertex. Only L and E triangles open new zipping opportunities. Zipping the borders of an E triangle may start a chain of zipping

operations (Figure 54.3.4). The cost of the zipping process is linear, since there are as many zipping operations as edges in the VST and the number of failed zipping tests equals the number of E or L-triangles.

GUARANTEED ENCODING OF THE CLERS STRING

Except for the first two triangles, there is a one-to-one mapping between each C-triangle and each vertex. Consequently, the number of C-triangles is $v - 2$, and the number of non-C-triangles is $t - (v - 2) = v - 2$. Thus exactly half of the triangles are of type C, and Edgebreaker guarantees a compression of not more than 2.0 bits per triangle [Ros99] using a trivial code, where the first bit is used to distinguish C-triangles from non-C-triangles.

Given that the subsequences CE and CL are impossible, a slightly more complex code [KR99] may be used to guarantee that the compressed file will not exceed 1.83t bits.

Further constraints exist on the *clers* string. For example, CCRE is impossible,

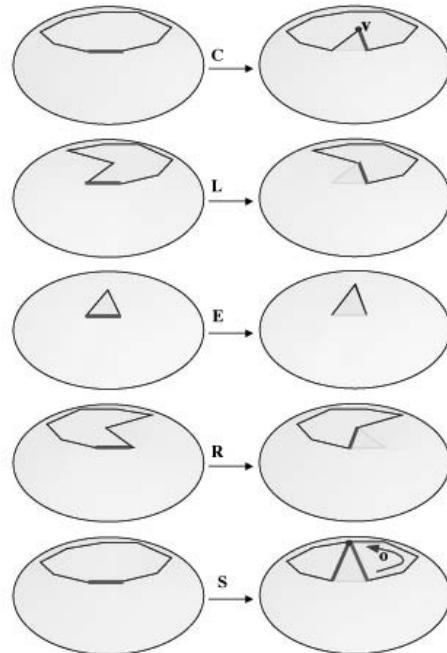
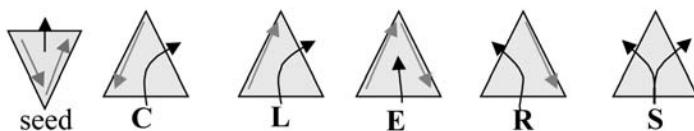


FIGURE 54.3.2

The five Edgebreaker mesh reconstruction operations attach a new triangle to the gate (indicated by a red line on the left column) in the active border loop around a hole in the partly reconstructed mesh. The C operation (top) creates a new vertex (v). The tip of the S-triangle (bottom) may be identified by an offset o . The S operation first puts the gate on the right edge of the new triangle and proceeds to fill the right hole using a recursive call. Then it sets the gate to the left edge of the new triangles and resumes the process. Note that C and S increment the edge count, L and R decrement it, and E reduces it by 3.

FIGURE 54.3.3

The borders of the web are oriented clockwise, except for the seed and the C triangles.



because CCR increments the length of the loop, which must have been at least 3. By exploiting such constraints to better estimate the probability of the next symbol, a more elaborate code was developed [Gum00], which guarantees 1.778t bits when using a forward decoding [RS99], and $1.776t$ bits when using a reverse decoding scheme [IS99]. Hence, the Edgebreaker encoding of the connectivity of any mesh (homeomorphic to a sphere) may be compressed down to $1.78t$ bits. This brings it within 10% of the proven $1.62t$ theoretical lower bound for encoding planar triangular graphs, as established by [Tut62], who by counting all possible planar triangulations of v vertices proved that an optimal encoding uses at least $v \log_2(256/7) \approx 3.245v$ bits, for a sufficiently large v .

STATISTICAL ENCODING

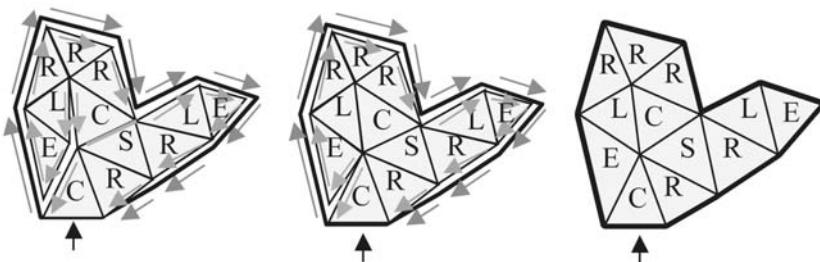
Recent developments in the guaranteed compression ratios discussed above not only have a theoretical importance, but ensure excellent and fast compression and decompression for meshes with irregular connectivity and for large collections of small meshes.

In practice, however, better compression ratios may often be obtained. For example, one may encode CC, CS, and CR pairs as single symbols. Each odd C symbol will be paired with the next symbol. After an even number of C symbols, special codes lead to a guaranteed $2.0t$ -bit encoding [RS99].

Furthermore, by arranging symbols into words that each start with a sequence of consecutive Cs and by using a **Huffman** code, we often reduce storage to less than $1.0t$ bits. For example, 0.85t bits suffice for the Huffman codes of the Stanford Bunny. Including the cost of transmitting the associated 173 word dictionary brings the total cost to 0.91t bits. A **gzip** compression on the resulting bit stream improves it only by 2%.

FIGURE 54.3.4

We assume that the part of the mesh not shown here has already been decoded into a web with properly oriented borders (exterior arrows). Building the TST (shown by the labeled triangles) for the substring CRSRLECRRRLE produces a web whose free borders are oriented clockwise for all non-C-triangles and counterclockwise for C triangles (left). Each time Wrap&Zip finds a pair of edges oriented away from their common vertex, it matches them. The result of the first zip operation (center) enables another zip. Repeating the process zips all the borders and restores the desired connectivity (right).



CONNECTIVITY PREDICTION

In addition to vertex location prediction, the connectivity of the next vertex may also be predicted using the same information. The *Delphi* connectivity prediction scheme [CR02] works as follows. The triangle connectivity, and its *clers* string produced by the Edgebreaker compression, is estimated by *snapping* the tip-vertex to the nearest bounding vertex of the active loop, if one lies sufficiently close. If no bounding vertex lies nearby, the next *clers* symbol is estimated to be a C. If the guess is correct, a single confirmation bit is sufficient. Otherwise, an entropy-based code is received and used to select the correct CLERS symbol from the other four possible ones (or the correct tip of an S-triangle). Experiments indicate that, depending on the model, up to 97% of Delphi's guesses are correct, compressing the connectivity down to $0.19t$ bits. When the probability of a wrong guess exceeds 40% the Delphi encoding ceases to be advantageous.

54.3.2 OTHER CONNECTIVITY COMPRESSION APPROACHES

CUT-BORDER MACHINE

Although invented independently, the *cut-border machine* [GS98] has strong similarities with Edgebreaker. Because it requires encoding the offset of S-triangles and because it was designed to support manifold meshes with boundaries, cut-border is slightly less effective than Edgebreaker. Reported connectivity compression results range from $1.7t$ to $2.5t$ bits. A context-based arithmetic coder further improves them to $0.95t$ bits [Gum99]. Gumhold proposes [Gum00] a custom variable length scheme that guarantees a total of less than $0.94t$ bits for encoding the offsets, thus proving that the cut-border machine has linear complexity.

TOPOLOGICAL SURGERY AND MPEG-4

Turan [Tur84] noted that the connectivity of a planar triangle graph can be recovered from the structure of its VST and TST, which he proposed to encode using a total of roughly $12v$ bits. There is a simple way to reduce this total cost to $6v$ bits by combining two observations: (1) The binary TST may be encoded with $2t$ bits, using two bits to indicate the presence or absence of left and right children. (2) The corresponding (dual) VST may be encoded with $1t$ bits, one bit per vertex indicating whether the node is a leaf and one bit indicating whether it is the last child of its parent. (Recall that $2v = t + 2$.) This scheme does not impose any restriction on the TST. Note that for less than the $2t$ bits budget needed for encoding the TST alone, Edgebreaker encodes the *clers* string, which not only describes how to reconstruct the TST, but also how to orient the borders of the resulting web, so as to define the VST, and hence the complete incidence. This economy comes from the fact that it uses a *spiraling* TST.

Taubin and Rossignac have noticed that a spiraling VST, formed by linking concentric loops into a tree, has relatively few branches. Furthermore, the corresponding dual TST, which happens to be identical to the TST produced by Edgebreaker, has also few branches (Figure 54.3.5). They have exploited this regularity by *Run Length Encoding* the *TST* and the *VST*. The resulting *Topological Surgery* compression technique [TR98] encodes the length of each run, the struc-

ture of the trees of runs, and a marching pattern, which encodes each triangle run as a generalized *triangle strip* [ESV96], using one bit per triangle to indicate whether the next triangle of the run is attached to the right or left.

An IBM implementation of the Topological Surgery compression has been developed for the VRML standard [THLR98] for the transmission of 3D models across the Internet, thus providing a compressed binary alternative to the original VRML ASCII format [VRML97], resulting in a 50-to-1 compression ratio. Subsequently, the Topological Surgery approach has been selected as the core of Three Dimensional Mesh Coding (3DMC) algorithm in *MPEG-4*, the ISO/IEC multimedia standard developed by the Moving Picture Experts Group for digital television, interactive graphics, and interactive multimedia applications.

Instead of linking the concentric rings of triangles into a single TST, the *layered structure* shown in Figure 54.3.5 (left) may be preserved [BPZ99]. The incidence is represented by the total number of vertex layers, and by the triangulation of each layer. When the triangle layer is a simpler closed strip, its triangulation may be encoded as a triangle strip, using one marching bit per triangle. However, in practice, a significant number of overhead bits are needed to encode the connectivity of more complex layers.

HARDWARE DECOMPRESSION IN JAVA 3D

Focusing on hardware decompression, Deering [Dee95] encodes generalized triangle strips using a buffer of 16 vertices. One bit identifies whether the next triangle is attached to the left or right border edge of the previous triangle. Another bit indicates whether the tip of the new triangle is a new vertex, whose location must be encoded in the stream, or a previously processed vertex that is still in the buffer and can hence be identified with 4 bits. Additional bits are used to manage the buffer and to indicate when a new triangle strips must be started. This compressed format is supported by the Java 3D's Compressed Object node.

Chow [Cho97] has provided an algorithm for compressing a mesh into Deering's format by extending the border of the previously visited part of the mesh by a fan of not-yet-visited triangles around a border vertex. When the tip of the new triangle is a previously decoded vertex no longer in the cache, its coordinates, or an absolute or relative reference to them, must be included in the vertex stream, significantly

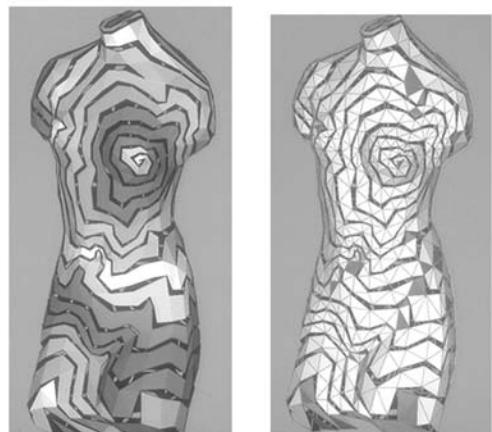


FIGURE 54.3.5

The Topological Surgery approach merges concentric circles of triangles into a single TST (left). That TST and its dual VST have relatively few runs (right).

increasing the overall transmission cost. Therefore, the optimal encoding traverses a TST differently from the spiraling TST of Edgebreaker, so as to reduce cache misses.

VALENCE-BASED INCIDENCE COMPRESSION

A consequence of Euler's theorem is that the average vertex valence is 6. In fact, in most models, the valence distribution is highly concentrated around 6. For example, in a subdivision mesh, all vertices that do not correspond to vertices of the original mesh have valence 6. To exploit these statistics, Touma and Gotsman [TG89] have developed a valance-based encoding, which visits the triangles in the same order as Edgebreaker. They encode the distinction between the C- and the S-triangles as in Edgebreaker, but instead of encoding the symbols for L, R, and E, they encode the valence of each vertex and the offset for each S-triangle. When the number of incident triangles around a vertex is one less than its valence, the missing L, R, or E triangle may be completed automatically. For this scheme to work, the offset must not only encode the number of vertices separating the gate from the tip of the new S-triangle along the border, but also the number of incident triangles on the tip that are part of the right hole.

Only one bit is needed to distinguish a C from an S. Given that 50% of the triangles are of type C and usually about 5% of the triangles are of type S, the amortized entropy cost of that bit is around $0.22t$ bits. Therefore, about 80% of the encoding cost lies in the valence, which has a low entropy for regular and finely tessellated meshes, and in the encoding of the offsets. To fix ideas, consider the example where 80% of the vertices have valence 6. A bit used to distinguish them from the other vertices may be encoded using $0.36t$ bits. The amortized cost of encoding the valence of the other 20% vertices with 2 bits each is $0.40t$ bits. Thus, the valence of reasonably regular meshes may be encoded with $0.76t$ bits. If 5% of the triangles are of type S and each offset is encoded with an average of 5 bits, the amortized cost of the offsets reaches $0.25t$ bits. Thus, offsets add about 25% to the cost of encoding the C/S bits and the valence.

Alliez and Desbrun [AD01a] managed to significantly reduce the number of S-triangles, and thus the cost of encoding the offsets, by using a heuristic that selects a gate with a low probability of producing an S-triangle. They also show that if one could eliminate the S-triangles completely, the valence-based approach would guarantee to compress the mesh with less than $1.62t$ bits, which happens to be Tutte's lower bound [Tut62].

An improved Edgebreaker compression approach was proposed [SKR00] for sufficiently large and regular meshes. It is based on a specially designed context-based coding of the *clers* string and uses the Spirale Reversi decompression. For a sufficiently large ratio of degree-6 vertices and sufficiently large t , this approach guarantees a worst-case storage of $0.81t$ bits.

54.4 SURFACE SIMPLIFICATION

Most of the details are usually far from the viewer. Their perspective projections on the screen appear small and thus need not be displayed at full resolution. Therefore, to avoid transmission delays, only crude approximations (called **Levels**-

(of-Detail) will be transmitted initially. They are produced by single-resolution *simplification* processes discussed in this section. The use of LODs as a technique for accelerating graphics is supported in graphics libraries, such as OpenGL. We do not discuss here the image-based techniques that replace such distant models with *imposters* [DSSD99] made by pasting, as textures, low-resolution images of them onto simple polygons. Instead, we focus on techniques that reduce the triangle count while striving to minimize the error introduced by the simplification. The connectivity and geometry of these simplified models may be compressed using the single-rate compression techniques discussed above. Refinements that upgrade their fidelity may be transmitted later, if at all needed.

We can simplify the mesh by moving one or more vertices to *collapse* one or more triangles, which may then be identified and discarded. Removing a collapsed triangle from the new mesh will not affect the surface, but may create a hole in the connectivity graph. For example, removing a single triangle that has been collapsed by displacing one of its vertices to the mid-point of the opposite edge will create a topological hole sometimes called a “T-junction.” Therefore, we will only remove triangles that have 2 or 3 coincident vertices. The two main simplification approaches, *vertex clustering* and *edge collapse*, are discussed below.

GLOSSARY

Uniform LOD: A simplified model constructed by striving to maintain a uniform distribution of error between the original and the simplified model.

View-dependent LOD: A simplified model created by adjusting the tolerance to a particular set of view parameters, e.g., increasing the accuracy of the approximation close to the viewer and to the visible silhouettes of the model.

Multi-Resolution Model (MRM): A representation from which view-dependent LODs may be efficiently extracted as the viewpoint is displaced.

Vertex-Clustering Simplification: A mesh simplification process that collapses clusters of vertices with identical quantized coordinates to one representative vertex and removes collapsed triangles when redundant.

Edge-collapse: A simplification step that collapses pairs of edge-connected vertices along nearly straight edges or nearly flat regions. Each edge-collapse also collapses two triangles, which may be removed.

Silhouette: The union of the edges of a mesh that are bounded by a front and a back-facing triangles.

Hausdorff distance: The Hausdorff deviation $H(A, B)$, between two sets, A and B , is the largest distance between a point in one of the sets and the other set.

Quadratic error: The sum of the squares of the distances between a new position of a vertex v and planes that contain the triangles incident upon v in the original mesh.

54.4.1 VERTEX CLUSTERING

Vertex clustering [RB93], among the simplest simplification techniques, is based on a crude vertex *quantization*, obtained by imposing a uniform, axis-aligned grid

(Figure 54.4.1) and clustering all vertices that fall in the same grid cell. The simplest implementation of vertex clustering associates each vertex v with a cell number $v.\text{cell}$ computed by concatenating its three quantized coordinates. The quantized version, x' of the x coordinate is the integer part of $s_x(x - x_{\min})/(x_{\max} - x_{\min})$, where s_x is the number of cells in the x direction. Similar expressions are used for quantizing the y and z coordinates. If one wishes to guarantee a uniform error, $\{s_x, s_y, s_z\}$ are chosen so that the $\{x, y, z\}$ dimensions of each cell are nearly identical. The computational cost of this quantization pass is linear and does not require accessing any connectivity or incidence information. One may think of $v.\text{cell}$ as the *cluster name* for the vertices in it.

A second pass selects a *representative vertex* for each cluster. It is often preferable to use one of the vertices of the cluster, rather than to create a new location for the representative vertex. Picking the vertex closest to the average of the cluster vertices has a tendency to shrink the objects. Rossignac and Borrel [RB93] found that the vertex furthest from the center of the model's bounding box is a better choice (Figure 54.4.1). Even better choices may be obtained by using more expensive schemes, which weigh vertices based on the local curvature or on the probability that they would be on a *silhouette* of the object, when seen from a random direction, and then select the closest vertex to the weighted average in each cluster.

Rendering all of the triangles of the original mesh with each vertex replaced by its cluster representative will produce an image of the simplified model; see Figure 54.4.1. To reduce the redundancy of this representation, one may choose to identify and eliminate all degenerate triangles, which may be done in a single pass over all triangles. Note, however, that groups of triangles that model thin branches or small shells may collapse to dangling edges or to isolated points. It would be simple to remove them by not drawing zero-area triangles. However, this option would result in a loss of information about the presence of the object along these thin branches or in the small isolated components. Therefore, these dangling edges and vertices are usually identified and preserved. Consequently, a third step of vertex clustering removes all degenerate triangles that have 2 or 3 vertices in the same cluster, but also creates a list of dangling edges and isolated vertices.

To identify dangling edges and vertices, one can construct a list of six triplets, one per triangle, using the three cluster names of its vertices in all possible permutations. This list may be sorted efficiently by using hashing on the first cluster name in each triplet. All entries $\{a, b, c\}$ that start with $\{a, b\}$ are consecutive. If the third element c in all of them is either a or b , then (a, b) is a dangling edge. Similarly, if all entries that start with $\{a\}$ are of the form $\{a, a, a\}$, then a is an isolated vertex.

This vertex clustering approach has been used in several commercial systems as a simplification tool for accelerating the rendering of 3D models (such as IBM's 3DIX and OpenGL), not only because of its simplicity and speed, but also because it does not use adjacency information beyond the triangle/vertex incidence table, and because it may be applied to arbitrary collections of triangles, edges, and vertices, with no topological restrictions. For example, it may, in a single process, simplify 3D models of solids and 2D models of their drawings, including vector-based text and mark-up.

A second vertex clustering phase with a coarser grid may be applied to the LOD produced by a first pass. Repeating this process produces a series of LODs that all use the same initial vertex set. When, for each LOD, the cell size is reduced by two

in each dimension, the vertex clusters correspond to a hierarchy of clusters, which may be stored in an octree [Sam90]. Luebke [LE97] has used vertex clustering with such an octree to support view-dependent simplification.

Clearly, no vertex has moved by more than the length of the diagonal of a cell, and thus this length offers a bound on the maximum geometric error between the original shape and the simplified one. This is a tight bound on the Hausdorff distance between the two shapes, which may be of considerable importance to manufacturing, robotics, and animation applications where computing clearance and detecting interferences and collisions is important.

Unfortunately, vertex clustering rarely offers the most accurate simplification for a given triangle-count reduction. One of its problems lies in the fact that the result is affected by the grid alignment, which may split nearby vertices into separate clusters, and replace them with distant representatives. Allowing the cells to float a bit [LT97] considerably improves the quality of some models, although at the expense of a slightly higher computational cost.

A more delicate problem may be illustrated by considering the 3D triangulated model of a scanned shape whose vertices have been sampled on a regular grid. If we use a large cell size, important features of the mesh will be simplified. If we use a small cell size, the triangulations of flat or nearly flat portions of the surface will retain an excessive triangle count, because their vertices are not allowed to slide along the surface past their cell boundaries. To overcome this constraint, we would like to have cluster cells that form *slabs* around the surface, with a small thickness in the normal direction that captures the tolerance, but with a wide tangential spread over flat areas. Clearly, if a manifold vertex has all its neighbors in such a slab, then moving it to one of its neighbors will not result in an error that exceeds the tolerance. By progressively adjusting the slab, several techniques identify simply connected groups of coplanar or nearly coplanar triangles and then retriangulate their surface to eliminate interior vertices [KT96]. Instead of local slabs, Cohen et al. [Co⁺96] compute offset surfaces, called *envelopes*, that bound a tolerance zone around the surface, which is used to constrain vertex merging operations.

54.4.2 EDGE COLLAPSING

Collapsed triangles may be created by edge-collapse operations (Figure 54.4.2), which each merge two vertices and collapse two triangles. These collapsed triangles may be easily removed from the connectivity graph. For example, to update a Corner Table, we identify the collapsed edge by a corner c (Figure 54.4.2) and use it to access the corners of the neighboring triangles. We mark the corners of the collapsed triangles as “unused.” Then the V- and O-tables are updated in the natural way. See Figure 54.4.3 for an example.

Note that these updates assume that we have a manifold representation. Hence, most simplification techniques preclude topology-changing edge-collapses, which for example would create dangling edges or nonmanifold vertices. Many of the algorithms also assume that each vertex of the mesh has at least three incident triangles, and that no two edges have identical endpoints. Edge-collapses that violate these restrictions may be easily detected and prevented. Thus, the triangle-count reduction capacity of the restricted simplification may be reduced for objects with many holes or thin branches, such as the chair of Figure 54.4.1.

A simplified mesh may be produced by a sequence of edge-collapse opera-

tions [Hop96]. Most simplification techniques strive to select at each stage the edge whose collapse will have the smallest impact on the total error between the resulting simplified mesh and the original surface. Deciding how the error should be measured is delicate and computing it precisely is time consuming. Therefore, most simplification approaches are based on a compromise where the error is estimated, as discussed below. These estimates are used to select the best candidate for the next edge-collapse. Error estimates for portions of the mesh that have been affected by prior collapses must be updated. A priority queue is used to maintain a sorted list of candidates.

54.4.3 MIXED APPROACHES

FIGURE 54.4.1

Left: vertex clustering simplification on a 2D mesh. Top left: grouping vertices; top right: cluster representative vertex. Bottom left: degenerate triangles removed; bottom right: all vertices replaced by cluster representative, dangling edges and isolated vertices added. Right: simplified coarse approximation.

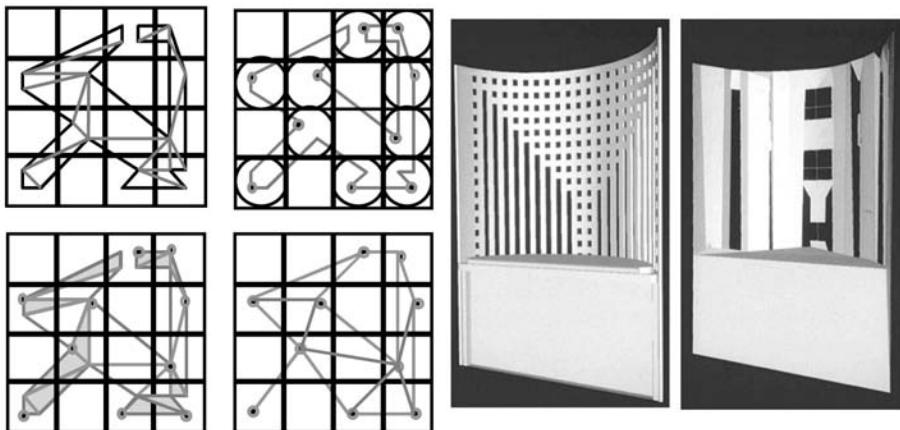
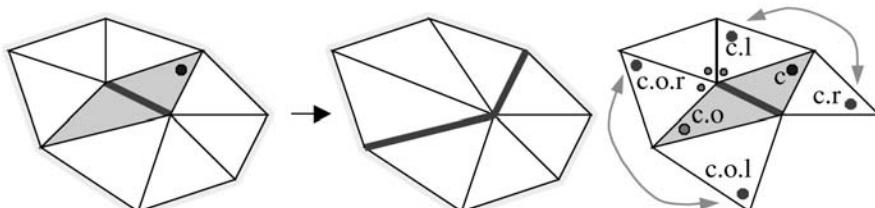


FIGURE 54.4.2

Collapsing the fat edge (left) to one of its vertices collapses its two incident triangles (center). The corner table of the collapsed mesh may be updated using the corner c (right). The reverse of an edge-collapse is a vertex-split.



It is also possible to combine vertex-clustering and edge-collapse approaches. For example, vertex-clustering may be used as a geometric and topological filter to remove small features, such as holes or bumps. Then, the result may be simplified through an edge-collapse process, which is better suited for removing redundant samples along flat portions of the surface and along nearly linear edges. The error resulting from this combination is bounded by the sum of the cell diagonal used in vertex clustering plus the estimate of the worst case error produced by edge collapsing.

In order to prevent topology changes and still exploit the speed and simplicity benefits of vertex clustering, one may split the cluster of vertices within a given cell into edge-connected sub-clusters. Collapsing each sub-cluster into a separate representative vertex will not merge disconnected components, but can still create nonmanifold singularities and dangling faces and edges. A local topology analysis may be conducted to detect and prevent these collapses. Such a combination of vertex clustering and edge-collapse was exploited for performing out-of-core simplification [Lin00] of datasets that are too large to fit in memory and thus would make the random access to the vertex data and connectivity tables, which are performed by edge-collapsing operations, too expensive.

Finally, several authors [PH97] [Lue98] have extended edge-collapsing simplifications by adding *virtual edges*, which are computed using vertex clustering and make it possible to merge components and to deal with nonmanifold meshes.

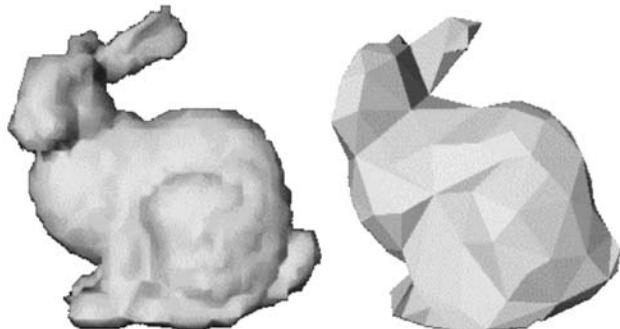
54.4.4 ERROR MEASURES

The error between two shapes could be measured in *image space*, as the discrepancy between the colors of the pixels for a particular set of views [LRC⁺02] [Lin00]. However, such error measures rely on assumptions as to how a particular color change or displacement of the projection of a highlight or color discontinuity on the screen impact the perceived fidelity of the image. They also require a fine sampling of the higher dimensional space of view parameters, and are thus expensive to compute.

The geometric error between the projection of the two shapes on the screen

FIGURE 54.4.3

The 3D model (left) has been simplified by a sequence of edge-collapses to a model with a much lower triangle count (right).



can be formulated more objectively, but is also view-dependent. For example, it may be formulated as the discrepancy between the projections of the silhouettes and sharp edges of both shapes.

Thus, most simplification techniques are based on a view-independent error formulation. The error, $E(A, B)$, between a shape A and a shape B may be formulated as the maximum of the distance, $d(p, S)$, from all points p on A or B , to S , the other shape, (B or A , respectively). This formulation is equivalent to the **Hausdorff** distance $H(A, B)$, which may also be formulated as the smallest radius r , such that $A \subset B \uparrow r$ and $B \subset A \uparrow r$, where $S \uparrow r$ denotes the expanded set S obtained by replacing each point q of S by a ball of center q and radius r . The distance $d(p, S)$ may be computed as the minimum of distances between p and the vertices of S , the normal projections of p onto the edges of S , and the normal projections of p onto the interiors of the triangles of S .

The difficulty in computing $H(A, B)$ lies primarily in the fact that it is not sufficient to test $d(p, S)$ for all vertices p of A and B , because the maximum error may occur at points inside faces, not at vertices or edges. Consequently, the exact Hausdorff measure is often approximated by *super-sampling* the two surfaces and computing the Hausdorff distance between the two dense sets of samples. The popular **Metro** tool [CRS98] super-samples one surface only and computes the maximum of the distance between the samples and the other surface.

Another drawback of the Hausdorff distance is that it does not require a good mapping from one shape to the other and fails to identify for example the fact that nearby and parallel portions of the two surfaces may have opposite orientation.

Thus, most simplification algorithms use a local error estimation. Consider a vertex that has moved from its initial position v to a new position v' , as a result of a vertex clustering step or of a series of edge-collapses. The distance $\|vv'\|$, which is bounded by the cell diagonal in the vertex clustering approach, provides a conservative bound on the Hausdorff error resulting from this displacement. However, it is too conservative when the mesh was nearly planar in the vicinity of v and when the vector vv' is tangent to the surface. Clearly, we want to measure the component of the displacement of that vertex along the normal to the surface.

The error resulting from the collapse of a vertex v_1 to its neighbor v_2 , can be estimated by the dot-product $v_1v_2 \cdot N_1$, where N_1 is the surface normal computed at v_1 . Although simple, this formulation does not guarantee a conservative error bound.

Ronfard and Rossignac [RR96] used the maximum of the distance between the new position of v' and the planes that contain the original triangles incident upon v . The distance between v' and the plane containing vertices (v, a, b) is $vv' \cdot (va \times vb) / \|va \times vb\|$. The right term may be pre-computed and cached for each triangle in the original mesh using its vertices v , a , and b . Note that for very sharp edges and vertices, an additional plane is necessary to measure excursions of v' that would be tangential to all the faces incident upon v and yet would move away from the surface. That normal to that additional plane may be estimated using the surface normal estimation at v . The cost of this approach lies in the fact that, as vertices are merged through series of edge collapses, one needs to keep track of the list of all the planes that contain the triangles incident to these vertices in the original model. Furthermore, for each new edge-collapse candidate, the distance between the new position v' must be computed to all these planes. If the edge collapse is executed, the lists of planes must be merged.

By trading the conservative maximum error bound of [RR96] for a quadratic

error measure, Garland and Heckbert [GH98] have considerably reduced the computational cost of error estimation. The square of the distance $D_1(P)$ between a point P and the plane $\text{Plane}(Q_1, N_1)$ through a point Q_1 with unit normal N_1 , is $(N_1 \cdot Q_1 P)^2$. It is a quadratic polynomial in the coordinates (x, y, z) of P .

Based on this observation, in a preprocessing stage, we pre-compute the 10 coefficients of $D_1(P)$ for each corner c_k , considering Plane $(c_k.p.v.g, N_k)$, where the normal N_k is computed as $(c_k.p.v.g - c_k.v.g) \times (c_k.p.v.g - c_k.v.g)$. Note that N_k is not a unit vector. Its length is proportional to the area of c.t. Then for each vertex v_m , we compute the coefficients by adding the respective coefficients of its corners. They define the function D_m associated with that vertex.

During simplification, we estimate the cost of an edge collapse that would move a vertex v_1 to a vertex v_2 , by $D_1(v_2)$. We perform the collapse with the lowest estimate and add to each coefficients of D_2 the corresponding coefficient of D_1 .

54.4.5 OPTIMAL QUANTIZATION FOR EACH LOD

Assume that simplification produces a mesh A that approximates the original mesh O within a conservative sampling error estimate e_S . The vertex quantization performed during the compression of A will produce a mesh C with a quantization error e_Q with respect to A. Thus, the conservative bound on the total error is $e_S + e_Q$. How should we choose the optimal combination of a triangle count t for A and of the number B of bits used by the quantization? Assume for instance that we have a fixed bit budget. If we use a LOD with too many triangles, we will have very few bits per coordinate, and thus will need to quantize them drastically. Given the magnitude of the resulting quantization error, we may decide that the model is over-sampled, and that we are better off by increasing B and using a lower LOD. Setting $e_S = e_Q$ yields a solution that is in general significantly suboptimal. An approximate solution to this optimization problem has been derived [KRS99] by formulating e_S as $K(t)/t$, and by approximating the shape complexity function $K(t)$ by a constant K , which may be estimated from a curvature integral over the model. In fact, for a sphere, $K(t)$ is constant.

54.5 RE-SAMPLING

The simplification approaches described above reduce the sampling of the mesh, while decreasing its accuracy. They strive to minimize the estimated error for a given vertex count reduction. In contrast, the re-sampling techniques described in this section strive to reduce storage cost while maintaining an acceptable accuracy. They are based on two strategies: (1) increase the regularity of the connectivity of the mesh, so as to increase connectivity compression; (2) constrain each new vertex to lie on a specific curve, so as to reduce to one the number of coordinates that must be specified per vertex.

GLOSSARY

Regular subdivision: A sequence of mesh refinement steps, each of which in-

serts new vertices in the middle of the edges and splits triangles into four. The positions of the new and old vertices are adjusted by a vector computed from the neighboring vertices.

Wavelet compression: A hierarchical encoding of regularly spaced data as a series of residues to values predicted by interpolation from a coarser model.

Normal meshes: The interleaving of mesh refinement steps with displacement steps, which adjust the inserted vertices along estimated surface normal. Statistical properties of the adjustments make them suitable for wavelet compression and progressive transmission.

54.5.1 REPULSION-BASED RETILING

An early simplification technique based on re-sampling [Tur92] first places *samples* on the surface of the mesh and distributes them evenly through an iterative process using *repulsive forces* computed from estimates of the geodesic distances [PS99] between samples. The repulsive forces may be adjusted taking local curvature into account so as to increase sample density in high curvature areas. Then it refines the mesh by inserting these samples as new vertices and hence splitting the triangles that contain them. Finally, the old vertices are removed through edge-collapse operations that preserve topology. This elegant process acts as a low pass filter and produces pleasing simplifications for smooth surfaces. Unfortunately, it may produce unnecessarily large errors near sharp features and does not reduce the cost of encoding the vertices.

54.5.2 NORMAL MESHES

Another approach [KSS00] uses the MAPS algorithm [LSS⁺98] to compute a crude LOD, which can be compressed using any of the single-rate compression schemes discussed above. Once received and restored by the decompression module, the crude LOD is used as the coarse mesh of a subdivision process. Each subdivision stage splits each edge of the mesh into two and each triangle into four, by the insertion of one new vertex per edge. They use the **Loop** subdivision (Chapter 53), which splits edge (c.n.v,c.p.v) by inserting point $(c.v.g+c.o.v.g+3c.n.v.g+3c.o.v.g)/8$ and then moves the old vertices by a fraction toward the average of their old neighbors.

After each subdivision stage, they download a displacement field of corrective vectors and use them to adjust the vertices, to bring the current level subdivision surface closer to the desired surface. The distribution of the coefficients of the corrective vectors is concentrated around zero and their magnitude diminishes as subdivision progresses. They are encoded using a wavelet transform and compressed using a modified version of the SPIHT algorithm [SP96] originally developed for image compression.

Instead of encoding corrective 3D vectors, the **Normal Mesh** approach [GVSS00] restricts each offset vector to be parallel to the surface normal estimated at the vertex. Only one corrective displacement value needs to be encoded per vertex, instead of three coordinates. They use the **Butterfly subdivision** [DLG90], which preserves the old vertices, and for each pair of opposite corners c and c.o splits the edge (c.n.v,c.p.v) by inserting a point computed from 8 vertices.

They encode the corrective displacement values of each new vertex using the unlifted version of Butterfly wavelet transform [DLG90] [ZSS96]. Further subdivision stages will generate a smoother mesh that interpolates these displaced vertices. The difficulty of this approach lies in the computation of a suitable crude LOD and in handling situations where no suitable displacement may be found for a new vertex along the estimated surface normal, because for example the normal does not intersect the original mesh. Furthermore, that original mesh and the connectivity constraint imposed by the regular subdivision process limit the way in which the retiling can adapt to the local shape characteristics, and thus may result in less effective compression ratios. For example, regular meshes may lead to sub-optimal triangulations for surfaces with high curvature regions and saddle points, where vertices of valence different than 6 are more appropriate.

54.5.3 PIECEWISE REGULAR MESHES

The surface of the mesh may be algorithmically decomposed into *reliefs* [SRK02]. Each relief may be re-sampled by a regular grid of parallel rays. Triangles are formed between samples on adjacent rays and also, to ensure the proper connectivity, at the junction of adjacent reliefs.

The sampling rate (i.e., the density of the rays) may be chosen so that the resulting Piecewise Regular Mesh (PRM) has roughly the same number of vertices as the original mesh. In these situations, the PRM typically approximates the original mesh with the mean square error of less than 0.02% of the diameter of the bounding box. Because of the regularity of the sampling for each relief, the PRM may be compressed for both connectivity and geometry down to about $2t$ bits.

The PRM compression algorithm uses a modified version of the Edgebreaker compression and of the Spirale Reversi decompression to encode the global relief connectivity and the triangles which do not belong to the regular regions. First, Edgebreaker compression is used to convert the mesh to be encoded into a CLERS string. Then, the CLERS string is turned into a binary string using the context-based range, which reduces the uncertainty about the next symbol for a highly regular mesh.

The geometry of the reliefs is compressed using an iterated two-dimensional variant of the differential coding [Sal00]. The regular resampling causes the entropy of the parallelogram rule residues to decrease by about 40% when compared to the entropy for the original models, because on reliefs, two out of three coordinates of the residual vectors become zero.

Since this approach does not require global parametrization, it may be used for models with complex topologies. It is faster than the combination of the MAPS algorithm [LSS⁺98] and the wavelet mesh compression algorithm of [GVSS00] [KSS00], while offering comparable compression rates.

54.5.4 SWINGWRAPPER

The *SwingWrapper* approach [AFSR03], shown in Figure 54.5.1, partitions the surface of an original mesh M into simply connected regions, called *triangloids*. From these, it generates a new mesh M' . Each triangle of M' is a linear approximation of a triangloid of M . By construction, the connectivity of M' is fairly regular,

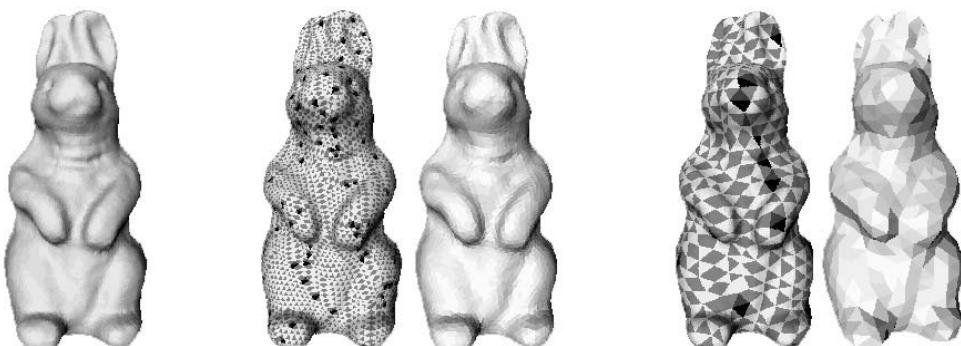
and can be compressed to less than a bit per triangle using EdgeBreaker. The locations of the vertices of M' are compactly encoded with a prediction technique that uses a single correction parameter per vertex. Instead of using displacements along the surface normal or heights on a grid of parallel rays, SwingWrapper requires that all C-triangles be isosceles, with left and right edges having a prescribed length L . SwingWrapper strives to reach a user-defined output file size rather than to guarantee a given error bound. For a variety of popular models, a rate of 0.4t bits yields a mean square error of about 0.01% of the bounding box diagonal.

54.6 PROGRESSIVE TRANSMISSION

When the full resolution model of a mesh is unnecessary or when immediate graphic feedback is needed, a compressed crude model obtained through simplification or resampling is downloaded first. Then, later, if a higher resolution is required, the full-resolution mesh could be downloaded using a compact encoding produced by a single-rate compression. When the storage size of the compressed crude model is small compared to the storage size of the full mesh, the overhead of transmitting both, instead of the full mesh only, is small. Yet, the benefits are considerable if the full model never becomes necessary or if the user is not willing to wait for the full resolution model. However, in this binary mode scenario, when the accuracy of the crude model is no longer sufficient, the delay associated with downloading the full resolution mesh could be avoided if an intermediate resolution model could be downloaded instead and would provide sufficient accuracy. In fact, it may be desired to offer a series of intermediate LODs. Each one could be compressed independently using a single-rate compression scheme.

FIGURE 54.5.1

The original mesh (left) containing 134,074 triangles requires 4,100,000 bytes, when stored in the WRL format. A dense partitioning of its surface into triangloids (2nd) yields a retiled mesh (3rd) which deviates from the original by less than 0.6% of the radius of the smallest ball bounding the model. Its 13642 triangles are encoded with 3.5t bits. The resulting total of 6042 bytes represents a 678-to-1 compression ratio. A coarser partitioning (4th) decomposes the original surface into 1505 triangloids. The corresponding retiled triangle mesh (last) approximating the original model within 4% is encoded with 980 bytes: A 4000-to-1 compression.



The shortcoming of this “independent” approach lies in the fact that the transmission does not take advantage of the information already available at the client side. For example, if the accuracy requirements increase progressively, and the client ends up downloading 10 different LODs, each having twice more vertices than the previous one, the total cost for a mesh will be about $(1+2+4+8+\dots+1024)t/1024$ bytes, which is $2t$ bytes, or twice the cost of transmitting the full-resolution mesh. In fact the total cost will be somewhat higher, because the geometry and connectivity of crude models is less regular and will in general not compress to a byte per triangle.

This shortcoming may be addressed by using a *progressive transmission* approach where the connectivity and geometry of previously decoded LODs is exploited to reduce the transmission cost of the next LOD. Most progressive approaches compress the upgrade, which contains the description of where new vertices should be inserted, i.e., their location and the associated connectivity changes.

GLOSSARY

Vertex split: The inverse of an edge-collapse operation. It is specified by selecting a vertex v and two of its incident edges.

Upgrade: The information permitting to refine an LOD to a higher accuracy LOD.

Compressed Progressive Meshes: A representation combining a single-rate compression of the lowest LOD with compressed encodings of the successive upgrades.

54.6.1 PROGRESSIVE MESHES

The progressive transmission of compressed meshes started with Hoppe’s *Progressive Meshes (PM)* [Hop96]. It encoded the coarsest LOD and a series of vertex-split operations, which when applied to the coarse mesh reverses the sequence of simplifying edge-collapse operations that were used to create it. The advantage of PM is its ability to stop transmission at any desired accuracy, thus offering the finest granularity of upgrades, each one being a vertex-split. The compression effectiveness of PM is limited by its need to encode the absolute index of the vertex at which the vertex-split occurs. That index requires $\log_2(n)$ bits, where n is the number of vertices decoded so far.

54.6.2 COMPRESSED PROGRESSIVE MESHES

By grouping the vertex splits into batches, the *Compressed Progressive Mesh (CPM)* [PR00] eliminates the $\log_2(v)$ cost replacing the vertex indexing by a one-bit-per-vertex mask. When combined with a modified Butterfly geometry predictor estimating the location of each new vertex from its neighbors, it achieves about $11t$ bits for a combined transmission code of the complete geometry ($7.5t$ bits) and connectivity ($3.5t$ bits), while offering between 7 and 12 intermediate LODs for common meshes.

The approach is illustrated in Figure 54.6.1.

54.6.3 EDGE-MASKS

The Wavemesh of Valette and Prost uses a batch strategy similar to CPM, but uses the mask to mark the edges that must be split [VP04]. They use a new simplification algorithm, which removes vertices in an order that permits recreation of the original connectivity with a small number of additional bits (above the cost of the mask). Experimental results suggest that the average cost for encoding the complete connectivity of the progressive mesh ranges from $1.0t$ bits to $2.5t$ bits for commonly used meshes.

54.6.4 KD-TREES

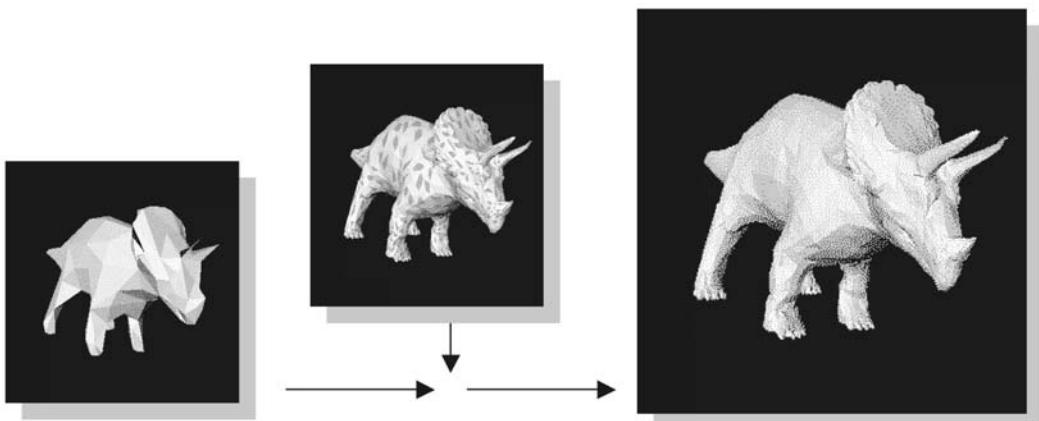
Gandois and Devillers [GD02] use a hierarchy of vertex clustering [RB93]. However, instead of an octree, as in [Lue98] they use a k -d-tree. At an x -split node they split the parent's area in two equal parts by a plane orthogonal to the x -axis and store the number of vertices in the left child. This adaptive organization of the vertices leads to the factorization of the most significant bits of the x -coordinate. The alternating y -split and z -split nodes perform a similar split, but with a different orientation of the splitting plane.

Thus, the coordinates of the centers of the leaves of the k -d-tree are defined implicitly by their position in the tree. The cost of storing them lies in the cost of encoding the numbers of vertices in the left child of each node. When only one vertex lies in a node, its least significant bits that have not been factored in the structure of the tree must be encoded.

When moving down the tree, each parent vertex is split into two vertices, which may, but need not, be connected. Thus, the connectivity may be encoded as a series of possibly nonmanifold vertex-splits, as proposed in [PH97]. The total cost of

FIGURE 54.6.1

The coarse mesh (left) produced by an edge-collapsing simplification is refined by a CPM upgrade which inserts new vertices and new triangles (center). The higher LOD is shown (right).



transmitting a progressive mesh compressed with this approach comprises $1.8t$ bits for the connectivity and about $7.8t$ bits for the geometry, when the original vertex coordinates have been quantified to 12 bits each.

54.6.5 VALENCE DRIVEN CONQUEST

To exploit the regularity of distribution of vertex degrees (or valences), which are concentrated around 6, Alliez and Desbrun [AD01a] use a series of passes to produce decreasing LODs. Each pass reduces the number of triangles by 3 by combining three steps that each traverses the mesh in a breadth-first order. The first one removes the tip vertices of triangles that have a valence of less than 7, leaving polygonal holes in the mesh. It also tags the remaining vertices. The second phase uses these tags to retriangulate the holes, striving to create vertices with valence 3 or 9. The third one removes valence-3 vertices. Because the decimation follows a systematic traversal, it can be reversed by decompression. The upgrade information for each pass contains the degree of the vertices removed during the first step (which must be 3, 4, 5, or 6) and an encoding of the traversal irregularities. These connectivity upgrades may be compressed to an average of $1.9t$ bits, which corresponds to a 40% improvement over CPM [PR00]. However, the selection of which vertices are removed at each phase is only dictated by the connectivity and cannot take into account the geometry, and thus cannot favor vertices whose removal will have the lowest impact on the geometric error. Consequently, this approach will produce intermediate models which for the same triangle count will be less accurate than those produced by CPM.

54.6.6 PROGRESSIVE NORMAL MESHES

By exploiting the hierarchical nature of the wavelet formulation, the normal meshes provide an effective progressive transmission scheme, in which the bits of the coefficients are transmitted in the order of their importance [SP96] [Sha93]. For the same transmission cost, this approach yields fourfold better accuracy than CPM [PR00]. In fact, the total cost of transmitting the highest accuracy mesh is often lower than one offered by the best single-rate compression schemes [TG89]. However, this approach is not capable of restoring the original mesh for two reasons: (1) it has the semiregular connectivity of a subdivision mesh and (2) the constraints on vertex locations make it impossible to restore the original surface.

54.7 SOURCES AND RELATED MATERIAL

SURVEYS

Simplification techniques have been surveyed in [HG97] [PS97] and more recently in [LRC⁺02]. Compression and progressive transmission techniques have been surveyed in [Ros99b].

Topological extensions of simplification and compression not covered here are

discussed in [GBT99] [RC99] [PH97] for nonmanifold models, in [Ros99] [Lo+02] for handles, in [TG89] for holes, and in [KRS99] [IS00] for quadrilateral and polygonal meshes [KRS99] [IS00]. Simplification and compression of meshes with properties are discussed in [GH98] [BPZ99] [IS00]. Compression and progressive transmission of tetrahedral meshes have been addressed in [PRS99] [SR00]. Error correction strategies for progressive transmission are addressed in [AAR02].

RELATED CHAPTERS

- [Chapter 25: Triangulations and mesh generation](#)
- [Chapter 26: Polygons](#)
- [Chapter 49: Computer graphics](#)
- [Chapter 53: Splines and geometric modeling](#)

REFERENCES

- [AAR02] G. Al-Regib, Y. Altunbasak, and J. Rossignac. An unequal error protection method for progressively compressed 3-D meshes. *Internat. Conf. Acoustics Speech Signal Proc. Multimedia Commun. Networking II Session*, 2002.
- [ABK98] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction. In *Proc. ACM Conf. SIGGRAPH 98*, pages 415–421, 1998.
- [AD01a] P. Alliez and M. Desbrun. Progressive encoding for lossless transmission of 3D meshes. *ACM SIGGRAPH Conf. Proc.*, 2001.
- [AD01b] P. Alliez and M. Desbrun. Valence-driven connectivity encoding for 3D meshes. In *Proc. Eurographics*, volume 20, pages 480–489, 2001.
- [AFSR03] M. Attene, B. Falcidieno, M. Spagnuolo, and J. Rossignac. SwingWrapper: Retiling triangle meshes for better EdgeBreaker compression. *ACM Trans. Graphics*, 22:982–996, 2003.
- [BPZ99] C.L. Bajaj, V. Pascucci, and G. Zhuang. Single resolution compression of arbitrary triangular meshes with properties. *Comput. Geom. Theory Appl.*, 14:167–186, 1999.
- [Cho97] M. Chow. Optimized geometry compression for real-time rendering. In *Proc. Conf. Visualization 97*, 1997, pages 347–354.
- [Co⁺96] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P.K. Agarwal, F.P. Brooks, Jr., and W.V. Wright. Simplification envelopes. In *Proc. ACM Conf. SIGGRAPH 96*, pages 119–128, 1996.
- [CR02] V. Coors and J. Rossignac. Guess connectivity: Delphi encoding in Edgebreaker. GVU Tech. Rep., 2002.
- [CRS98] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. In *Proc. Eurographics 98*, volume 17(2), pages 167–174, 1998.
- [Dee95] M. Deering. Geometry compression. In *Proc. ACM Conf. SIGGRAPH 95*, pages 13–20, 1995.
- [DLG90] N. Dyn, D. Levin, and J.A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph.*, 9:160–169, 1990.
- [DSSD99] X. Decoret, G. Schaufler, F.X. Sillion, and J. Dorsey. Multi-layered impostors for accelerated rendering. *Comput. Graph. Forum*, 18:61–73, 1999.

- [ESV96] F. Evans, S.S. Skiena, and A. Varshney. Optimizing triangle strips for fast rendering. In *Proc. IEEE Visualization*, pages 319–326, 1996.
- [GBT99] A. Gueziec, F. Bossen, G. Taubin, and C. Silva. Efficient compression of non-manifold polygonal meshes. In *Proc. IEEE Visualization*, pages 73–80, 1999.
- [GD02] P.M. Gandois and O. Devillers. Progressive lossless compression of arbitrary simplicial complexes. In *Proc. ACM Conf. SIGGRAPH 02*, pages 372–379, 2002.
- [GH98] M. Garland and P.S. Heckbert. Simplifying surfaces with color and texture using quadratic error metric. In *Proc. IEEE Visualization*, pages 287–295, 1998.
- [Gum00] S. Gumhold. Towards optimal coding and ongoing research, 3D Geometry Compression Course Notes. In *Proc. ACM Conf. SIGGRAPH 00*, 2000.
- [Gum99] S. Gumhold. Improved cut-border machine for triangle mesh compression. *Erlangen Workshop Vision, Modeling, Visualization*. IEEE Signal Proc. Soc., 1999.
- [GS98] S. Gumhold and W. Straßer. Real-time compression of triangle mesh connectivity. In *Proc. ACM Conf. SIGGRAPH 98*, pages 133–140, 1998.
- [GVSS00] I. Guskov, K. Vidimce, W. Sweldens, and P. Schröder. Normal meshes. In *Proc. ACM Conf. SIGGRAPH 00*, pages 95–102, 2000.
- [HG97] P.S. Heckbert and M. Garland. Survey of polygonal simplification algorithms. *Multi-resolution Surface Modeling Course*. ACM SIGGRAPH Course Notes, 1997. Tech. Rep. Carnegie Mellon Univ., Dept. of Computer Science.
- [Hop96] H. Hoppe. Progressive meshes. In *Proc. ACM Conf. SIGGRAPH 96*, pages 99–108, 1996.
- [Hop98] H. Hoppe. Efficient implementation of progressive meshes. *Computers and Graphics*, 22:27–36, 1998.
- [Hop99] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Proc. IEEE Visualization*, pages 59–66, 1999.
- [IS99] M. Isenburg and J. Snoeyink. Spirale Reversi: Reverse decoding of the Edgebreaker encoding. Tech. Report TR-99-08, Computer Science, UBC, 1999.
- [IS00] M. Isenburg and J. Snoeyink. Face fixer: Compressing polygon meshes with properties. In *Proc. ACM Conf. SIGGRAPH 00*, pages 263–270, 2000.
- [KR99] D. King and J. Rossignac. Guaranteed 3.67V bit encoding of planar triangle graphs. In *Proc. 11th Canad. Conf. Comput. Geom.*, pages 146–149, Vancouver, 1999.
- [KRS99] D. King, J. Rossignac, and A. Szymczak. Connectivity compression for irregular quadrilateral meshes. Tech. Rep. TR-99-36, GVU, Georgia Tech, 1999.
- [KSS00] A. Khodakovskiy, P. Schröder, and W. Sweldens. Progressive geometry compression. In *Proc. ACM Conf. SIGGRAPH 00*, pages 271–278, 2000.
- [KT96] A.D. Kalvin and R.H. Taylor. Superfaces: Polygonal mesh simplification with bounded error. *IEEE Comput. Graph. Appl.*, 16:64–67, 1996.
- [LE97] D.P. Luebke and C. Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proc. ACM Conf. SIGGRAPH 97*, pages 199–208, 1997.
- [Lin00] P. Lindstrom. Out-of-core simplification of large polygonal models. In *Proc. ACM Conf. SIGGRAPH 00*, pages 259–262, 2000.
- [LRS⁺03] H. Lopes, J. Rossignac, A. Safanova, A. Szymczak, and G. Tavares. Edgebreaker: A simple implemenatation for surfaces with handles. *Computers and Graphics*, 27:553–567, 2003.

- [LSS⁺98] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D.P. Dobkin. MAPS: Multiresolution adaptive parametrization of surfaces. In *Proc. ACM Conf. SIGGRAPH 98*, pages 95–104, 1998.
- [LRC⁺02] D.P. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Hubner. *Levels of Detail for 3D Graphics*. Morgan Kaufmann, San Francisco, 2002.
- [LT97] K-L. Low and T.-S. Tan. Model simplification using vertex clustering. In *Proc. ACM Symp. Interactive 3D Graphics*, pages 75–82, 1997.
- [Lue98] D.P. Luebke. *View-dependent simplification of arbitrary polygonal environments*. Ph.D. thesis, Univ. North Carolina, Chapel Hill, 1998.
- [LW85] M. Levoy and T. Whitted. The use of points as a display primitive. Tech. Rep. TR 85-022, Univ. North Carolina Chapel Hill, 1985.
- [PH97] J. Popovic and H. Hoppe. Progressive simplicial complexes. In *Proc. ACM Conf. SIGGRAPH 97*, pages 217–224, 1997.
- [PR00] R. Pajarola and J. Rossignac. Compressed progressive meshes. *IEEE Trans. Visualization Comput. Graphics*, 6:79–93, 2000.
- [PRS99] R. Pajarola, J. Rossignac, and A. Szymczak. Implant sprays: Compression of progressive tetrahedral mesh connectivity. *IEEE Visualization*, San Francisco, pages 299–305, 1999.
- [PS97] E. Puppo and R. Scopigno. Simplification, LOD and multiresolution: Principles and applications. Tutorial at the *Eurographics 97 Conf.*, Budapest, pages 1–104, 1997.
- [PS99] K. Polthier and M. Schmies. Geodesic flow on polyhedral surfaces. In *Proc. Eurographics Workshop Sci. Visual.*, Vienna, pages 1–14, 1999.
- [RB93] J. Rossignac and P. Borrel. Multi-resolution 3D approximations for rendering complex scenes. *Geometric Modeling in Computer Graphics*, Springer-Verlag, Berlin, pages 445–465, 1993.
- [RC99] J. Rossignac and D. Cardoze. Matchmaker: Manifold Breps for non-manifold r-sets. In *Proc. ACM Symp. Solid Modeling*, pages 31–41, 1999.
- [Ros99] J. Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Trans. Visualization Comput. Graphics*, 5:47–61, 1999.
- [Ros99b] J. Rossignac. Compression and progressive refinement of 3D models. In *Proc. Shape Modeling Internat.*, Aizu, Japan, 1999.
- [RR96] R. Ronfard and J. Rossignac. Full range approximation of triangulated polyhedra. In *Proc. Eurographics 96*, pages 67–76, 1996.
- [RS99] J. Rossignac and A. Szymczak. Wrap&Zip decompression of the connectivity of triangle meshes compressed with Edgebreaker. *Comput. Geom. Theory Appl.* 14:119–135, 1999.
- [RSS03] J. Rossignac, A. Safanova, and A. Szymczak. Edgebreaker on a corner table: A simple technique for representing and compressing triangulated surfaces. In *Hierarchical and Geometrical Methods in Scientific Visualization*, G. Farin, H. Hagen, and B. Hamann, editors, Springer-Verlag, Heidelberg, pages 41–50, 2003.
- [Sal00] D. Salomon. *Data Compression: The Complete Reference*, 2nd edition. Springer-Verlag, Berlin, 2000.
- [Sam90] H. Samet. *Applications of Spatial Data Structures*. Addison-Wesley, Reading, 1990.
- [SP96] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, 6:243–250, 1996.

- [Sch98] M. Schindler. A fast renormalization for arithmetic coding. In *Proc. IEEE Data Compression Conf.*, page 572, 1998.
- [Sha93] J. Shapiro. Embedded image coding using zero-trees of wavelet coefficients, *IEEE Trans. Signal Process.*, 41:3445–3462, 1993.
- [SKR00] A. Szymczak, D. King, and J. Rossignac. An Edgebreaker-based efficient compression scheme for connectivity of regular meshes. *Comput. Geom. Theory Appl.*, 20: 53–68, 2000.
- [SRK02] A. Szymczak, J. Rossignac, and D. King. Piecewise regular meshes: Construction and compression. In *Graphics Models*, 64:183–198, 2002.
- [SR00] A. Szymczak and J. Rossignac. Grow&Fold: Compressing the connectivity of tetrahedral meshes. *Comput. Aided Design*. 32:527–538, 2000.
- [THLR98] G. Taubin, W. Horn, F. Lazarus, and J. Rossignac. Geometry coding and VRML. *Proc. IEEE*, 96:1228–1243, 1998.
- [TG89] C. Touma and C. Gotsman. Triangle mesh compression. In *Proc. Graphics Interface*, pages 26–34, 1998.
- [TR98] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Trans. Graph.*, 17:84–115, 1998.
- [Tur84] G. Turán. On the succinct representations of graphs. *Discrete Appl. Math.*, 8:289–294, 1984.
- [Tur92] G. Turk. Re-tiling polygonal surfaces. In *Proc. ACM Conf. SIGGRAPH 92*, pages 55–64, 1992.
- [Tut62] W. Tutte. A census of planar triangulations. *Canad. J. Math.*, 14:21–38, 1962.
- [VP04] S. Valette and R. Prost. A wavelet-based progressive compression scheme for triangle meshes: Wavemesh. *IEEE Trans. Visualization Comput. Graphics*, 10:123–129, 2004.
- [VRML97] ISO/IEC 14772-1. *The Virtual Reality Modeling Language (VRML)*. 1997.
- [ZSS96] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proc. ACM Conf. SIGGRAPH 96*, pages 189–192, 1996.

55 MANUFACTURING PROCESSES

Ravi Janardan and Tony C. Woo

INTRODUCTION

This chapter surveys some recent work on the application of techniques from computational geometry to geometric problems arising in manufacturing processes such as layered manufacturing, mold design, and numerically controlled machining. Within each topic, we discuss problems that have benefited from the application of geometric techniques, and mention several other problems where such techniques could be used to advantage.

55.1 LAYERED MANUFACTURING

Layered Manufacturing (LM) is a relatively new technology which allows physical prototypes of 3D models to be built directly from their digital representations, using a “3D printer” attached to a computer [Jac92]. LM provides the designer with an additional level of physical verification and facilitates the early detection and correction of design flaws that may have gone unnoticed otherwise. The use of LM has proliferated into a wide variety of areas, including, among others, engineering (e.g., automotive and aerospace design), ergonomic product design (e.g., hand-held devices such as cell phones), medicine (e.g., prosthetics design and tissue engineering), and art (e.g., sculpture) [Cad02, Har01, KF97, Lev02, NLG02].

The basic principle underlying LM is simple: The digital model is oriented suitably and sliced into horizontal layers by a plane. The layers are transmitted over a network to a fabrication device, which “prints” them successively in the vertical direction, each layer on top of the previous one; thus the physical prototype is realized as a vertical stack of two-dimensional layers. The efficiency and accuracy of LM depends, in part, on the efficient solution of a number of geometric problems. For instance, the choice of the model’s orientation determines the number of layers, the surface finish, and the quantity and location of temporary support structures. The problem of printing the layers efficiently reduces to covering the interior of a polygon with a collection of thin rectangles. Other problems include slicing the model efficiently and generating a compact representation of the support structures.

GLOSSARY

STL format: The model is assumed to be given as a surface triangulation. The format specifies the triangles by listing the coordinates of their vertices and the direction cosines of their unit outer normals. This is the *de facto* industry standard for LM; the name is derived from STereoLithography, one of the first

LM processes to be developed.

Model orientation: The rotation of the model from its default orientation in the STL file, prior to being sliced into horizontal layers and built in the vertical direction.

Stairstep error: Stairstep-shaped artifacts introduced on the model's facets due to discretization into layers (similar to antialiasing in computer graphics), which affect surface finish and accuracy. The stairstep error on a facet is the height of the stairstep perpendicular to the facet. It is a function of the (smaller) angle between the vertical direction and the facet's outer normal, hence of the model's orientation.

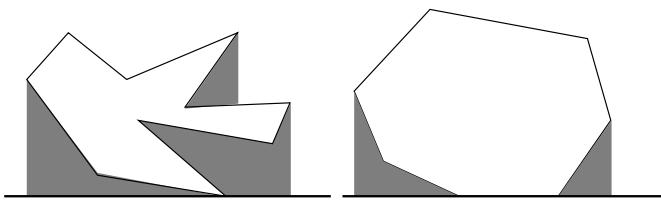
Supports: Temporary structures that are built simultaneously with the model to prop up layers that overhang previously-built layers; these are removed in a postprocessing step. Formally, for a model P , a point $p \in \mathbb{R}^3 \setminus P$ is part of the supports if the upward-directed ray from p intersects P ; thus the membership of p in supports depends on the model's orientation. The supports form a collection of disjoint polyhedra. (Figure 55.1.1.)

Support requirements: Measured in two ways: The support volume is the total volume of the support polyhedra. The support contact-area is the area of that portion of the model's boundary that is in contact with supports. These should be minimized to reduce material costs, build time, and postprocessing time.

Hatching: The process of printing each layer (a polygon) by covering its interior with parallel rectangles of some small width; the width is a process parameter.

FIGURE 55.1.1

Support structures (shown shaded) for a nonconvex polygon (left) and a convex polygon (right). Illustration is in two dimensions for convenience. (Reprinted from [IJM+02], with permission from Elsevier.)



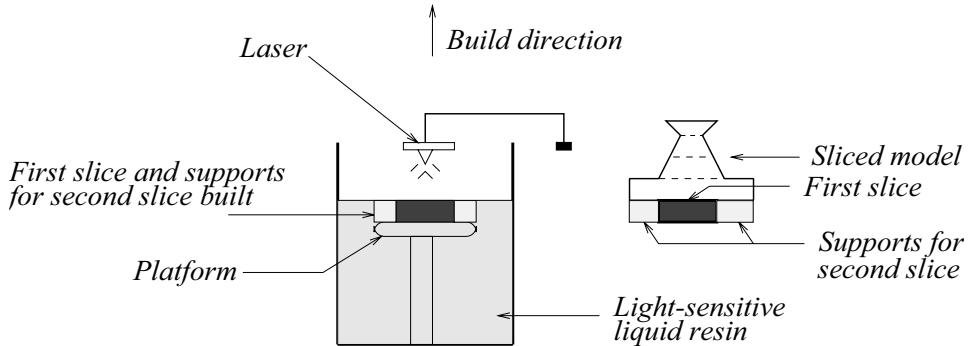
RESULTS

Figure 55.1.2 illustrates schematically a widely-used LM process called Stereolithography, where the printing of layers is achieved by having a laser trace out and hatch each layer on the surface of a liquid resin which hardens when exposed to light. After a layer is built, the platform is lowered by an amount equal to the layer thickness (on the order of a few thousandths of an inch) and the next layer is then built atop the previous one. The need for supports is ascertained beforehand, by analyzing the orientation and geometry of the model, and then generating and merging a description of the layers into the STL file for the model. Representative examples of other LM processes include Fused Deposition Modeling (where layers are printed by extruding and laying down molten plastic via a nozzle), Laminated Object Man-

ufacturing (where the layers are cut out from sheets of adhesive-backed paper), and 3D Printing (where the layers are realized by outlining their shape via a binder fluid and then depositing a special powder onto it).

FIGURE 55.1.2

The Stereolithography process. (Reprinted from [IJM+02], with permission from Elsevier.)



Geometric problems in LM can be grouped loosely into three categories:

Choice of model orientation. Here the goal is to choose an orientation for the model that optimizes some design criterion (or to simply decide if the criterion can be satisfied). In [AB⁺97], $O(n)$ -time algorithms are given for deciding whether an n -vertex polyhedron can be built without supports using two models of Stereolithography—one in which no layer can overhang the previous one, and another where some overhang, controlled by an angle parameter, is allowed. The classes of objects that can be built by these processes are also related to those buildable via NC-machining and casting. In [MJSG99], an algorithm is given to minimize the maximum stairstep error ([BB95]) over all facets of a polyhedron in $O(n \log n)$ time and to minimize the sum of the stairstep errors on all facets in $O(n^2)$ time; the first algorithm even allows facets to be weighted to indicate their relative importance with respect to surface finish. Also given are $O(n^2)$ -time algorithms to minimize the volume and (independently) the contact-area of supports for a convex polyhedron. In [MJSS01], the preceding results are combined to reconcile simultaneously multiple design criteria, including support volume, contact-area, stairstep error, and number of layers. (Minimizing the number of layers is equivalent to finding the width of a polyhedron, and efficient solutions are known for this [HT88, SSMJ99].) Three formulations for reconciling the criteria are considered: optimizing the criteria sequentially, optimizing a weighted combination of the criteria, and allowing the criteria to meet designer-specified thresholds. The methods in [MJSG99, MJSS01] use well-known techniques from computational geometry, such as spherical arrangements, convex hulls, and Voronoi diagrams, in conjunction with constrained optimization methods such as Lagrangian multipliers. In [AD00], an approximation algorithm is given for minimizing the contact-area for a convex polyhedron. This method, based on computing approximate levels in a weighted arrangement of lines, runs in $O((n/\epsilon^3) \log^3 n)$ expected time and has an approximation ratio of $1 + \epsilon$, for any $\epsilon > 0$.

Optimization of supports for a nonconvex polyhedron is much harder due to the complicated structure of the supports. As Figure 55.1.1 illustrates, supports need not extend all the way to the platform, but may instead terminate on the model itself. Furthermore, only a fraction of a facet need be in contact with supports, unlike the convex case where either a facet is entirely in contact with supports or not in contact at all. In [MJS⁺99], algorithms are given for the two-dimensional case, i.e., minimizing support area and contact-length. The approach is based on partitioning the unit-circle of directions into intervals and generating for each interval a formula for the support requirement of interest. The intervals are then scanned in order and the formula for each interval is updated incrementally from that of the previous interval and then optimized within the interval. The running time is $O(n \log n)$ plus the time to perform $O(n)$ minimizations of a certain polynomial of degree $\Theta(n)$. Heuristics for contact-area optimization are described in [AD95], where a subset of the facet normals of the convex hull of the model is used for choosing the orientation. For each orientation, the needed supports and their contact-area are computed approximately, and the best orientation is then output. No analysis of the quality of the approximation is provided. In earlier work, the problem of support optimization is addressed in [FF94], and heuristics are given in the context of an expert system.

Another design consideration in LM is to choose model orientations so that certain functionally-critical surfaces of the prototype (e.g., facets on gear teeth) are not in contact with supports, since the presence and subsequent removal of supports can affect surface finish and accuracy. In [SSJ⁺00, SSJJ], an $O(n^2)$ -time algorithm is given to compute a description of all model orientations for which a prescribed facet is not in contact with supports. The related optimization problem of computing a description of all orientations for which the total area of facets not in contact with supports is maximized is solved in $O(n^4)$ time. These results are based on convex hulls, arrangements, and overlays of subdivision—all on the unit-sphere.

Fixed-orientation problems. Once an orientation has been chosen, several tasks remain. These include computing a description of any needed supports, slicing the model and supports, and deciding on how best to hatch the layers. In commercial software packages for LM, slicing and support generation are usually done in tandem. Specifically, as the model is sliced, the volume subtended under each layer is subtracted from that subtended by the layer above it; the result is the support between the two layers. Thus the supports are generated as a sequence of thin layers. In [Joh99], an alternative approach is pursued, where the goal is to generate a combinatorial description of the supports, as a collection of disjoint polyhedra. The algorithm is based on cylindrical decomposition [Mul93] and runs in $O(n^2 \log n)$ time.

Slicing algorithms used in LM are inefficient in that they compute from scratch the intersection of each slicing plane with the polyhedron, instead of taking advantage of the coherence that exists from layer to layer. This is due, in part, to the lack of any topological information in the STL format. In [MS99], an efficient and robust slicing algorithm is proposed. The algorithm builds a data structure based on a generalization of the well-known winged-edge structure [Bau75] and then uses the plane sweep paradigm to compute and update the layers incrementally, by taking advantage of the topological similarity between closely-spaced layers. A different perspective on slicing is taken in [DM94, KD96], where the focus is on slicing a

model adaptively, with slices of variable thickness, so as to improve surface accuracy and to speed up the build time.

The hatching problem may be viewed as the two-dimensional analog of the model orientation problem. Here the goal is to find a common orientation of all the layer polygons (or, equivalently, a rotation of the model about the vertical axis) so that the total number of times the hatching tool (e.g., the laser in Stereolithography) meets the boundaries of all the polygons is minimized. This, in turn, minimizes the number of starts, stops, and direction changes of the tool and increases tool life. In [HJSS03], the problem is approximated as one of finding a direction in the plane that minimizes the sum of the lengths of the projections of all polygon edges in this direction. The latter problem is reduced to computing the width of a suitably defined convex polygon (see also [Sar99]). The overall running time is $O(n' \log n')$, where n' is the total number of number of polygon edges in all layers. On real-world STL models, the algorithm runs very fast and delivers solutions that are very close to the solution produced by an optimal, but much slower, algorithm [SSHJ02].

Decomposition problems. LM processes generally view the model as a single, monolithic unit. An alternative approach is to decompose the model into a small number of pieces, build the individual pieces, and then glue them back together. This allows large models to be built in parallel on multiple machines (or even simultaneously on the same machine) and also reduces the build time. Moreover, the support requirements of the decomposed parts is usually lower than that of the original. This decomposition-based approach is pursued in [IJM⁺02], where it is shown how to decompose, with a plane, a convex or nonconvex polyhedron in a given orientation into a user-specified number of pieces so that the support volume or contact-area is minimized. The algorithms run in $O(n \log n)$ and $O(n^2 \log n)$ time for convex and nonconvex models, respectively, and are based on cylindrical decomposition and space sweep. In related work [FM01], it has been shown that the problem of deciding whether a polyhedron of genus zero or a polygon with holes can be decomposed into k terrains (hence built with zero supports) is NP-complete; here k is part of the input. In [IJS02], it is shown how to decompose, with a line, a polygon into two smaller polygons such that each is a terrain in the direction normal to the line; the algorithm runs in $O(n \log n)$ or $O(n^2 \log n)$ time, depending on whether or not both terrains have their “base” edge on the dividing line (see also [AB⁺97, RR94] for related work).

Besides the problems described above, a (necessarily incomplete) list of other related work on LM includes: automatic repair of STL files [Bøh95, Bar97]; elimination of support structures for a class of models by selectively thickening the walls of the model [AD98]; the design of a complete software front-end for the LM pipeline, from digital model import, to model repair, to batch scheduling of multiple models [BK98]; new modeling techniques for LM based on voxels [CMP95] and on analytic surfaces such as quadrics [FK96]; and investigation of alternatives to the STL format in LM [KD97, DKPS98].

OPEN PROBLEMS

1. Support optimization for nonconvex polyhedra is a challenging problem and an optimal solution remains elusive. Specifically, given a nonconvex polyhedron, \mathcal{P} , the goal is to compute an orientation which minimizes the support volume or (independently) the contact-area when \mathcal{P} is built in that orientation.

tion. Extending the method in [MJS⁺99] to three dimensions appears difficult and expensive, so a new approach may be needed. Also of interest would be simple and efficient approximation algorithms.

2. The decomposition algorithm in [IJM⁺02] assumes that an orientation is given for the model and then proceeds to find a decomposition which minimizes the support requirements. A natural extension of this is to find an optimal (or near-optimal) decomposition over all possible orientations. Similarly, for the hatching problem, it would be interesting to design an algorithm which computes an optimal or near-optimal hatching direction over all possible orientations of the model.
3. Although the STL format is the *de facto* industry standard for model representation in LM, it is plagued with many problems. It introduces an approximation error when used to represent smooth surfaces, it lacks useful topological information, it is highly redundant and error-prone, and it is very voluminous for surfaces of high curvature. As mentioned earlier, alternatives to STL have been investigated [KD97, DKPS98, CMP95, FK96] recently. In particular, in [FK96] a representation based on quadric surfaces has been considered. It has been shown empirically that for the tasks of slicing and filling in the layers using equidistant offset curves, this analytic representation is superior to STL both in accuracy and in computation time and memory requirements. A natural extension of this work would be to investigate the effect of such representations on other LM tasks such as support generation and minimization, reduction of stairstep error, layer minimization, and hatching.

55.2 MOLD DESIGN

Casting and injection molding processes are used extensively to mass-manufacture a wide variety of products. A key step here is the design of the mold from a digital model of the part, since this affects both the speed of the process and the quality of the finished part. For instance, how the model is decomposed into pieces to make the mold halves determines the number of undercuts in the mold: the greater the number of undercuts, the slower the de-molding process. As another example, the location of venting holes on the mold and the choice of pouring direction determine the extent of air pockets created during mold filling; this ultimately affects the strength and finish of the product.

GLOSSARY

Mold: A cavity in the shape of the part to be manufactured into which molten metal is poured. It consists of two mating parts called **mold halves**. Once the metal has hardened, the mold halves are pulled apart in opposite directions (i.e., **de-molded**) and the part is removed.

Undercut: Any point p on a part's surface such that the outward normal at p makes an angle greater than 90° with the de-molding direction for the mold half containing p . Generally, a group of such points forms a recess or projection in the part that prevents easy de-molding.

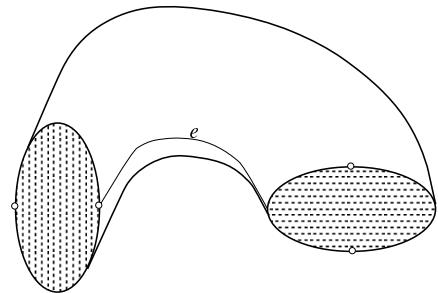


FIGURE 55.2.1

A parting line (e) for the exhaust manifold of an automobile.

Parting line: A continuous closed curve on the surface of the part that defines the two halves; thus it also defines the profile of the contact surface between the two mold halves (Figure 55.2.1).

CH(P): The convex hull of a polyhedron P .

Dent: For a polyhedron P , a connected component of $\text{CH}(P) \setminus P$.

Fillability: The ability to fill a mold from a given pouring direction without creating air pockets. This is a function of the mold geometry, the pouring direction, and the location of air-venting holes.

Part decomposition: The process of dividing a part into smaller pieces and making mold halves for these that satisfy certain optimization criteria.

RESULTS

Geometric problems in mold design generally fall into two categories.

Fillability problems. These are concerned with questions such as whether a mold can be filled from a given pouring direction without creating air pockets, and finding a pouring direction that eliminates air pockets using the smallest number of venting holes. In [BvKT98], several results are presented including: (a) deciding in $O(n)$ time whether an n -vertex polyhedron can be filled from a given pouring gate in a given direction without creating air pockets; (b) enumerating in $O(n^2)$ time all pouring directions that permit such a fill; (c) computing in $O(n^2)$ time a pouring direction which minimizes the number of air pockets; and (d) characterizing classes of polyhedra according to their fillability. The two-dimensional counterparts of these problems are solved in [BT95], with running time $O(n)$ for the decision problem and $O(n \log n)$ for the enumeration and optimization problems. Similar questions are also addressed in [FM93] for different mold-filling strategies and different types of filling material (ranging from gas to liquid to solid).

Part decomposition. This refers to the problem of “cutting” the digital model into smaller pieces and making mold halves for these that meet certain optimization criteria. For instance, how can a 3D part P be divided into two such that the parting line is as “flat” as possible? As noted in the mold-design literature, the flatter the parting line, the more cost-effective and accurate the mold. While the notion of flatness has not been quantified in the literature, it is generally taken to mean that the parting line should lie as nearly in a plane as possible. Although a parting line that lies completely in a plane can always be produced by intersecting P with a plane, this can create undercuts, even if P is a convex polyhedron (Figure 55.2.2).

The problem of computing a flattest undercut-free parting line for an n -vertex

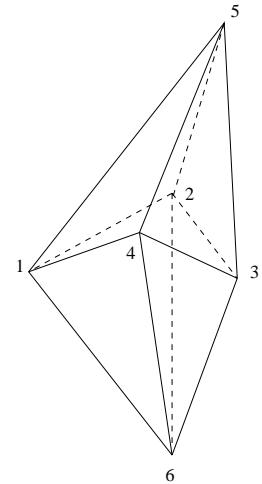


FIGURE 55.2.2

An octahedron that cannot be divided by a plane into two halves without creating undercuts. For example, the plane containing vertices 1, 2, and 3 creates a projection under the chain 1–4–3. Undercuts can be avoided by choosing the parting line to be 1–2–3–4–1 (or 2–5–4–6–2), but this is no longer in a plane. (From [MGJ99], with permission.)

convex polyhedron P is considered in [MGJ99]. That such a line always exists is clear—simply take the boundary, $L(\vec{d})$, of P , as viewed along lines of sight parallel to any direction \vec{d} . The **flatness**, $\rho(\vec{d})$, of $L(\vec{d})$ is defined in [MGJ99] as the sum of the squares of the projected lengths of the segments of $L(\vec{d})$, where the projection is onto a plane normal to \vec{d} , divided by the sum of the squares of the lengths of the segments of $L(\vec{d})$. Thus, $\rho(\vec{d}) \leq 1$, with equality holding if and only if $L(\vec{d})$ lies in a plane. An $O(n^2)$ -time algorithm is given to compute a direction \vec{d} that maximizes $\rho(\vec{d})$. The algorithm blends together geometric techniques such as visibility cones, arrangements, and shortest paths in a simple polygon, with methods from continuous optimization. Algorithms are also given for optimizing other measures of flatness. These include (a) finding a direction which maximizes the flatness criterion defined above, but uses segment lengths rather than squared lengths; and (b) finding a direction which minimizes the width of the parting line, where, for any direction \vec{d} , the width of the parting line $L(\vec{d})$ is defined to be the smallest separation between two parallel planes normal to \vec{d} that enclose $L(\vec{d})$.

In [BBvK97] the problem of deciding if a given n -vertex polyhedron can be parted by a single plane without creating undercuts is addressed. For an n -vertex nonconvex (resp. convex) polyhedron, where the cast parts are to be removed by translation in mutually-opposing directions, the bounds are $O(n^{3/2+\epsilon})$ time and $O(n^{3/2+\epsilon})$ space (resp. $O(n \log^2 n)$ time and $O(n)$ space), where $\epsilon > 0$ is an arbitrarily small constant. A related result is presented in [ADB⁺02], where it is shown that, for an n -vertex polyhedron, all directions that admit an undercut-free parting line (for cast removal in mutually opposing directions) can be computed in $O(n^4)$ time. This is shown to be optimal in the worst case by demonstrating a polyhedron which admits $\Omega(n^4)$ such directions. Finally, in [CC⁺93a], an $O(nd \log d)$ -time algorithm is given to compute a pair of opposing directions maximizing the number of visible dents in an n -vertex polyhedron with d dents. This minimizes the number of undercuts; however, the method does not yield a parting line.

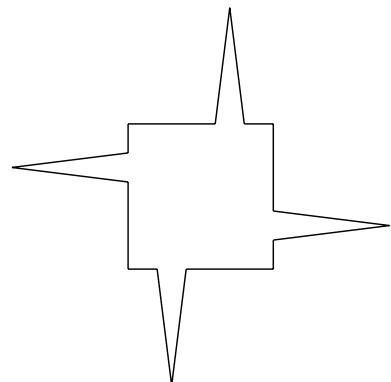
Other related work includes decomposition of two-dimensional molds [RR94], identification of criteria other than parting line shape and number of undercuts [RS90], and heuristics for computing a de-molding direction without too many undercuts [HT92].

OPEN PROBLEMS

1. It is unlikely that the $O(n^2)$ -time algorithm in [BvKT98] for minimizing the number of air-venting holes can be improved (in view of the 3SUM-HARD-based lower bound). However, can a significantly faster algorithm be devised that approximates the minimum number of air-venting holes to within a constant factor?
2. The goals of maximizing the flatness of the parting line and minimizing the number of undercuts are usually at odds. Often, however, meeting specified thresholds suffices: for instance, given parameters u_0 and ρ_0 , design an efficient algorithm to find a parting line with at most u_0 undercuts and flatness at least ρ_0 .
3. A polyhedron P is **1-castable** if it can be parted by a plane without creating undercuts. The results in [BBvK97] allow one to decide 1-castability efficiently. However, there exist polyhedra that are not 1-castable (Figure 55.2.3). To extend the class of polyhedra that can be cast with planes, call a polyhedron P **2-castable** if there is a plane h such that the polyhedra $P \cap h^+$ and $P \cap h^-$ are both 1-castable. (Here h^+ and h^- denote the two halfspaces of h .) Give efficient algorithms to decide 2-castability and characterize the class of 2-castable polyhedra.

FIGURE 55.2.3

Cross-sectional view of a polyhedron that is not 1-castable. The cross section tapers along the length of the polyhedron to a point and then expands again, so that the polyhedron consists of a “double pyramid.” Any casting plane will create an undercut at one (or more) of the spikes or at some of the slanted facets corresponding to the horizontal and vertical segments in the cross section.



55.3 NUMERICALLY CONTROLLED MACHINING

The dominant machining process today is *numerically controlled (NC) machining*, where parts are manufactured under computer control based on information extracted from a digital model. Examples of NC-machines include milling machines and lathes. Typical questions of interest concern accessibility of the tool to the part and generation of toolpaths that satisfy certain optimization criteria.

GLOSSARY

Degrees of freedom (dof): The types of motion permitted of an NC-machine. Specified as a combination of translation and (full or partial) rotation with respect to the coordinate axes.

Visibility map (or VMap): The set of points on the unit sphere representing the directions along which a tool can approach (or “see”; cf. Chapter 28) *all* points on the surface in question without being blocked by other portions of the part. The VMap is a function of the surface geometry and the geometry of the cutting tool, and is in practice usually representable as a (spherical) polygon formed by the intersection of a certain set of hemispheres [GWT94]. For instance, the VMap of a plane is the hemisphere whose pole is the normal to the plane, the VMap of a half-cylinder is a half-great circle, the VMap of a hemisphere is a point, and the VMap of a dent in a polyhedron is the intersection of the set of hemispheres determined by the normals to the dent’s faces.

Pocket: A region bounded by one or more closed curves, which delineates the area on the part from which material must be removed.

Spherical band of width b : The set of all points on the unit sphere that are at a distance of at most b on either side of a great circle, where the distance is measured along a great circle arc.

Part setup: The process of dismounting a part, and re-calibrating and re-mounting it in a new orientation on the worktable of an NC-machine.

Direction-parallel pocket machining: A machining discipline where the tool is constrained to stay within a pocket and, moreover, always moves from left to right with respect to a chosen reference line.

Zigzag pocket machining: Similar to direction-parallel machining, except that the tool moves from left to right, then right to left, and so on.

Contour-parallel pocket machining: The tool is constrained to move along a sequence of closed paths that are parallel to the pocket’s contour.

RESULTS

Two important parameters of an NC-machine are the dof of the machine and the type of cutting tool. The dof include translation along the principal coordinate directions (**3-axis machine**), plus rotation of the worktable about one axis (**4-axis machine**), plus partial swivel of the tool about a second axis (**5-axis machine**). The dof determines global motion of the tool. For example, in a 4-axis machine, the directions in which the tool can move can be represented on the unit sphere as a great circle whose normal is the rotational axis of the worktable. In a 5-axis machine, if the tool can swivel by $\pm b/2$ radians, then the tool motion directions are given by a spherical band of width b , where the great circle associated with the band is as in the 4-axis case. Cutters are classified, according to the maximum angle θ that they can tilt from the local surface normal, as: **flat-end** ($\theta = 0$ radians), **fillet-end** ($\theta < \pi/2$ radians), and **ball-end** ($\theta = \pi/2$ radians). Thus the cutter geometry determines local motion of the tool: a flat-end cutter can approach a point p on a surface only along the surface normal at p , while a ball-end cutter can approach p along any direction lying within the hemisphere with pole p .

Part orientation. In order to machine a surface on a part, the tool must be able to approach (or see) every point on the surface without being blocked by other portions of the part. For a given orientation of the part on the machine’s worktable, only a subset of the surfaces that need to be machined might be so visible to the tool. Therefore, after each such set of visible surfaces has been machined, a part setup is performed to bring a new set of surfaces into view. However, part setup can be quite time-consuming in relation to the actual machining time (hours versus minutes, sometimes). This motivates the following problem. Given the part geometry and the machine parameters, compute a sequence of part orientations that minimizes the number of setups. Unfortunately, this problem is NP-hard, and so attention has focused on obtaining efficient algorithms that approximate closely the minimum number of setups.

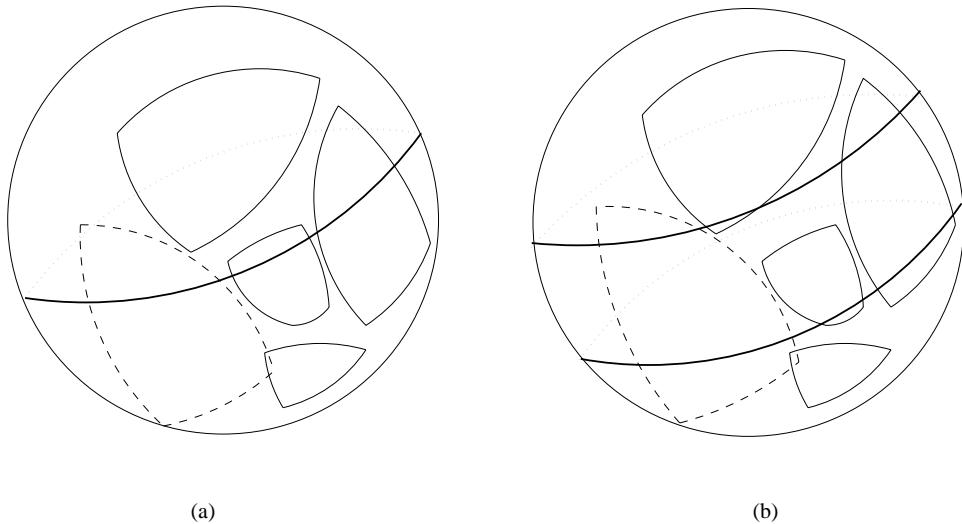
A natural approach is a greedy heuristic which finds repeatedly a part orientation that allows access to the maximum number of as-yet-unmachined surfaces [CC⁺93b, GJM⁺96]. Suppose, for example, that a 4-axis machine equipped with a ball-end cutter is used. Assume further that the VMaps for the part’s surfaces are available; for a ball-end cutter, the VMaps are intersections of certain hemispheres and can be computed as described in [GWT94]. Recall that each VMap represents the directions along which every point on the corresponding surface can be seen by the tool. Therefore, to find an orientation in which the maximum number of surfaces can be seen is equivalent to finding a great circle, C , that intersects the maximum number of VMaps. (Here, C represents the directions in which the tool can move in a 4-axis machine.) Similarly, for a 5-axis (resp. 3-axis) machine, the problem is to find a spherical band B of width b (resp. a point P) that intersects the maximum number of VMaps ([Figure 55.3.1](#)).

Given m VMaps with a total of n vertices, this problem is solved in [CC⁺93b] in $O(nm \log m)$ time and $O(nm)$ space for a 3- and 4-axis machine equipped with a ball-end cutter. In [GJM⁺96], the time bound is improved to $O(n^2)$ in the worst case—when $m = \Theta(n)$ —and, moreover, an $O(nm \log m)$ -time and $O(nm)$ -space algorithm is given for 5-axis machines. These results are based on geometric duality, topological sweep (Section 24.4), and properties concerning intersections and covering of polygons on the unit sphere. In [GJM⁺96], an $O(n^2 + nm \log m)$ -time and $O(nm)$ -space algorithm is also given for fillet-end tools on 4- and 5-axis machines. All of these results imply an $O(\log m)$ -approximation to the minimum number of setups, via the well-known approximation result for the set-cover problem.

Tool paths. A related problem is that of generating tool paths that meet certain optimization criteria, given the pocket geometry, the tool size and geometry, and a machining discipline such as direction-parallel machining, zigzag machining, or contour-parallel machining. The optimization criteria include minimizing the total length traveled by the tool, minimizing the number of *tool retractions* (i.e., the number of times the tool is lifted off the workpiece), and minimizing the number of times any point is machined by the tool. (This problem bears similarities to the hatching problem discussed earlier.) In [AHS00], a zigzag pocket machining algorithm is given and it is proved that the number of retractions is at most $5r + 6h$ for a pocket with $h \geq 0$ holes, where r is the minimum number of retractions. Moreover, no point is machined more than once. (Experiments in [AHS00] indicate a better approximation factor of 1.5.) The approach is based on constructing and processing a so-called machining graph. The algorithm runs in $O(n)$ time, where n is the number of vertices in the machining graph. (Here n is a function of the

FIGURE 55.3.1

A great circle for a 4-axis machine (a) and a spherical band for a 5-axis machine (b) intersecting a set of VMaps. (From [GJM⁺96], with permission.)



pocket geometry and the tool size.)

In [AFM00], the following related optimization problem is shown to be NP-hard: Given a polygonal pocket of size n and a tool represented by a unit disk or a square, find a closed path of minimum length that visits every point of the pocket at least once. It is shown, however, that one can compute a path that is at most a constant times longer than a shortest path in time $O(n \log n)$.

Heuristics have also been investigated for other tool-path generation problems—see, for instance, the references cited in [AHS00]. However, no approximation bounds have been proved.

OPEN PROBLEMS

1. The type of visibility considered in the part setup problem is between two points (one being the tool and the other being a point on the part's surface) along a straight line. Characterizations of such VMMaps and efficient algorithms are given in [GWT94]. Give characterizations and efficient algorithms for VMMaps under point-point visibility along circular trajectories (e.g., as produced by the rotary joints of a robot arm) or along parabolic trajectories (e.g., as executed by droplets under gravity in vapor deposition processes). Also of interest are segment-segment and plane-plane visibility along straight line trajectories.
2. Consider an augmented 4-axis (resp. 5-axis) machine, where the worktable can rotate fully (resp. tilt by $\pi/2$ radians) about a second axis. In the greedy framework described earlier, this reduces to finding a pair of orthogonal great circles (resp. spherical bands) that intersect the maximum number of VMMaps. No algorithms are known for this problem.

-
3. Prove that the zigzag pocket machining problem that calls for the minimum number of retractions and requires that no pocket point is machined more than once ([AHS00]) is NP-hard, or provide a polynomial-time algorithm.
 4. Investigate tool-path generation problems for contour-parallel machining and provide provably good approximation algorithms.
-

55.4 OTHER TOPICS

Besides the three representative topics that we have addressed, there are other areas for fruitful interaction between computational geometry and manufacturing. These include: design of mechanisms and linkages (Section 48.1); geometric constraint systems (Section 56.3); tolerancing of machined parts; interpretation and reconstruction of engineering drawings, assembly and disassembly of components (Section 48.3); geometric software for manufacturing applications, process planning and simulation, mesh generation (Section 25.4); VLSI design and layout, and vision, robotics ([Chapter 48](#)); geometric modeling issues relevant to manufacturing ([Chapter 53](#) and [56](#)); and geometric problems arising in other manufacturing processes such as bending, forming, welding, forging, etc.

55.5 SOURCES AND RELATED MATERIAL

FURTHER READING

The following contain additional discussion and references related to the topics in this chapter.

[Bos95, Maj98]: Provide good expositions of the application of computational geometry techniques to problems in molding, casting, and layered manufacturing.

[Woo94]: Discusses various kinds of visibility in the context of different manufacturing processes.

[Hel91]: Contains a detailed discussion of the application of geometric techniques to problems in pocket machining.

[Bra86]: A good general reference on a variety of design and manufacturing processes, including casting, molding, forging, stamping, machining, etc.

RELATED CHAPTERS

[Chapter 24: Arrangements](#)

[Chapter 28: Visibility](#)

[Chapter 29: Geometric reconstruction problems](#)

[Chapter 48: Robotics](#)

[Chapter 53: Splines and geometric modeling](#)

[Chapter 56: Solid modeling](#)

REFERENCES

- [AB⁺97] B. Asberg, G. Blanco, P. Bose, J. Garcia-Lopez, M.H. Overmars, G.T. Toussaint, G. Wilfong, and B. Zhu. Feasibility of design in stereolithography. *Algorithmica*, 19:61–83, 1997.
- [AD95] S. Allen and D. Dutta. Determination and evaluation of support structures in layered manufacturing. *J. Design Manufac.*, 5:153–162, 1995.
- [AD98] S. Allen and D. Dutta. Wall thickness control in layered manufacturing for surfaces with closed slices. *Comput. Geom. Theory Appl.*, 10:223–238, 1998.
- [AD00] P.K. Agarwal and P.K. Desikan. Approximation algorithms for layered manufacturing. In *Proc. 11th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 528–537, 2000.
- [AdB⁺02] H-K. Ahn, M. de Berg, P. Bose, S.W. Cheng, D. Halperin, J. Matoušek, and O. Schwarzkopf. Separating an object from its cast. *Comput. Aided Design*, 34:547–559, 2002.
- [AFM00] E.M. Arkin, S. Fekete, and J.S.B. Mitchell. Approximation algorithms for lawn mowing and milling. *Comput. Geom. Theory Appl.*, pages 25–50, 2000.
- [AHS00] E.M. Arkin, M. Held, and C. Smith. Optimization problems related to zigzag pocket machining. *Algorithmica*, 26:197–236, 2000.
- [Bar97] G. Barequet. Using geometric hashing to repair CAD models. *IEEE Comput. Sci. Eng.*, 4:22–28, 1997.
- [Bau75] B.G. Baumgart. A polyhedron representation for computer vision. In *Proc. AFIPS National Computer Conf.*, volume 44, pages 589–596, 1975.
- [BB95] M. Bablani and A. Bagchi. Quantification of errors in rapid prototyping processes and determination of preferred orientation of parts. In *Trans. 23rd N. Amer. Manuf. Research Conf.*, 1995.
- [BBvK97] P. Bose, D. Bremner, and M. van Kreveld. Determining the castability of simple polyhedra. *Algorithmica*, 19(1–2):84–113, 1997.
- [BK98] G. Barequet and Y. Kaplan. A data front-end for layered manufacturing. *Comput. Aided Design*, 30:231–243, 1998.
- [Bøh95] J.H. Bøhn. Removing zero-volume parts from CAD models for layered manufacturing. *IEEE Comput. Graphics Appl.*, 15:27–34, 1995.
- [Bos95] P. Bose. *Geometric and computational aspects of manufacturing processes*. Ph.D. thesis, School Comput. Sci., McGill Univ., Montréal, 1995.
- [Bra86] J.G. Bralla. *Handbook of Product Design for Manufacturing*. McGraw-Hill, Boston, 1986.
- [BT95] P. Bose and G.T. Toussaint. Geometric and computational aspects of gravity casting. *Comput. Aided Design*, 27:455–464, 1995.
- [BvKT98] P. Bose, M. van Kreveld, and G.T. Toussaint. Filling polyhedral molds. *Comput. Aided Design*, 30:245–254, 1998.
- [Cad02] CADCAM Net, 2002. <http://www.cadcammnet.com/Sections/rapid%20prototyping/Applications.htm>.
- [CC⁺93a] L.-L. Chen, S-Y. Chou, and T. Woo. Parting directions for mould and die design. *Comput. Aided Design*, 25:762–768, 1993.

- [CC⁺93b] L.-L. Chen, S-Y. Chou, and T. Woo. Separating and intersecting spherical polygons: Computing machinability on three-, four-, and five-axis numerically controlled machines. *ACM Trans. Graph.*, 12:305–326, 1993.
- [CMP95] V. Chandru, S. Manohar, and C. Prakash. Voxel-based modeling for layered manufacturing. *IEEE Comput. Graphics Appl.*, 15:42–47, 1995.
- [DKPS98] D. Dutta, V. Kumar, M. Pratt, and R. Sriram. Towards STEP-based data transfer in Layered Manufacturing. In *Proc. 10th Internat. Conf. PROLOMAT*, 1998.
- [DM94] A. Dolenc and I. Mäkelä. Slicing procedures for layered manufacturing techniques. *Comput. Aided Design*, 26:119–126, 1994.
- [FF94] D. Frank and G. Fadel. Preferred direction of build for rapid prototyping processes. In *Proc. 5th Internat. Conf. Rapid Prototyping*, pages 191–200, 1994.
- [FK96] R. Farouki and T. König. Computational methods for rapid prototyping of analytic solid models. *Rapid Prototyping J.*, 2:41–48, 1996.
- [FM93] S. Fekete and J.S.B. Mitchell. Geometric aspects of injection molding. *Workshop Geometric Comput. Aspects Injection Molding*, Bellairs Research Institute, 1993.
- [FM01] S. Fekete and J.S.B. Mitchell. Terrain decomposition and layered manufacturing. *Internat. J. Comput. Geom. Appl.*, 11:647–668, 2001.
- [GJM⁺96] P. Gupta, R. Janardan, J. Majhi, and T. Woo. Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC-machining. *Comput. Aided Design*, 28:577–587, 1996.
- [GWT94] J. Gan, T. Woo, and K. Tang. Spherical maps: Their construction, properties, and approximation. *J. Mech. Design*, 116:357–363, 1994.
- [Har01] G. Hart. Rapid prototyping of geometric models, 2001. <http://www.georgehart.com/cccg/rpgm.html>. Invited talk at *13th Canad. Conf. Comput. Geom.*, Waterloo, Canada, 2001.
- [Hel91] M. Held. *On the Computational Geometry of Pocket Machining*, volume 500 of *Lecture Notes Comput. Sci.*, Springer-Verlag, New York, 1991.
- [HJSS03] M. Hon, R. Janardan, J. Schwerdt, and M. Smid. Minimizing the total projection of a set of vectors, with applications to layered manufacturing. *Comput. Aided Design*, 35:57–68, 2003.
- [HT88] M.E. Houle and G.T. Toussaint. Computing the width of a set. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10:761–765, 1988.
- [HT92] K.C. Hui and S.T. Tan. Mould design with sweep operations—a heuristic search approach. *Comput. Aided Design*, 24:81–91, 1992.
- [IJM⁺02] I. Ilinkin, R. Janardan, J. Majhi, J. Schwerdt, M. Smid, and R. Sriram. A decomposition-based approach to layered manufacturing. *Comput. Geom. Theory Appl.*, 23:117–151, 2002.
- [IJS02] I. Ilinkin, R. Janardan, and M. Smid. Terrain polygon decomposition with application to layered manufacturing. In *Proc. 8th Internat. Comput. Combin. Conf.*, volume 2387 of *Lecture Notes Comput. Sci.*, pages 381–390, Springer-Verlag, Berlin, 2002.
- [Jac92] P.F. Jacobs. *Rapid Prototyping & Manufacturing: Fundamentals of StereoLithography*. McGraw-Hill, Boston, 1992.
- [Joh99] E. Johnson. Support generation for three-dimensional layered manufacturing. Master’s project report, Dept. of CS&E, Univ. Minnesota, Minneapolis, 1999.
- [KD96] P. Kulkarni and D. Dutta. An accurate slicing procedure for layered manufacturing. *Comput. Aided Design*, 28:683–697, 1996.

- [KD97] V. Kumar and D. Dutta. An assessment of data formats for layered manufacturing. *Advances in Engineering Software*, 28:151–164, 1997.
- [KF97] C.C. Kai and L.K. Fai. *Rapid Prototyping: Principles and Applications in Manufacturing*. John Wiley & Sons, New York, 1997.
- [Lev02] W. Leventon. Synthetic skin. *IEEE Spectrum*, pages 28–33, 2002.
- [Maj98] J. Majhi. *Geometric methods in computer-aided design and manufacturing*. Ph.D. thesis, Dept. of Comput. Sci. & Eng. Univ. Minnesota, Minneapolis, 1998.
- [MGJ99] J. Majhi, P. Gupta, and R. Janardan. Computing a flattest, undercut-free parting line for a convex polyhedron, with application to mold design. *Comput. Geom. Theory Appl.*, 13:229–252, 1999.
- [MJS⁺99] J. Majhi, R. Janardan, J. Schwerdt, M. Smid, and P. Gupta. Minimizing support structures and trapped area in two-dimensional layered manufacturing. *Comput. Geom. Theory Appl.*, 12:241–267, 1999.
- [MJSG99] J. Majhi, R. Janardan, M. Smid, and P. Gupta. On some geometric optimization problems in layered manufacturing. *Comput. Geom. Theory Appl.*, 12:219–239, 1999.
- [MJSS01] J. Majhi, R. Janardan, J. Schwerdt, and M. Smid. Multi-criteria geometric optimization problems in layered manufacturing. *Internat. J. Math. Algorithms*, 2:201–225, 2001.
- [MS99] S. McMains and C. Séquin. A coherent sweep plane slicer for layered manufacturing. In *Proc. 5th Annu. ACM Sympos. Solid Modeling Appl.*, pages 285–295, 1999.
- [Mul93] K. Mulmuley. *Computational Geometry: An Introduction through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, 1993.
- [NLG02] P. Ng, P. Lee, and J. Goh. Prosthetic sockets fabrication using rapid prototyping technology. *Rapid Prototyping J.*, 8:53–59, 2002.
- [RR94] A. Rosenbloom and D. Rappaport. Moldable and castable polygons. *Comput. Geom. Theory Appl.*, 4:219–233, 1994.
- [RS90] B. Ravi and M.N. Srinivasan. Decision criteria for computer-aided parting surface design. *Comput. Aided Design*, 22:11–18, 1990.
- [Sar99] S. Sarma. The crossing function and its application to zig-zag tool paths. *Comput. Aided Design*, 31:881–890, 1999.
- [SSHJ02] J. Schwerdt, M. Smid, M. Hon, and R. Janardan. Computing an optimal hatching direction in layered manufacturing. *Internat. J. Comput. Math.*, 79:1067–1081, 2002.
- [SSJ⁺00] J. Schwerdt, M. Smid, R. Janardan, E. Johnson, and J. Majhi. Protecting critical facets in layered manufacturing. *Comput. Geom. Theory Appl.*, 16:187–210, 2000.
- [SSJJ] J. Schwerdt, M. Smid, R. Janardan, and E. Johnson. Protecting critical facets in layered manufacturing: implementation and experimental results. *Comput. Aided Design*, 35:647–657, 2003.
- [SSMJ99] J. Schwerdt, M. Smid, J. Majhi, and R. Janardan. Computing the width of a three-dimensional point-set: an experimental study. *ACM J. Experimental Algorithms*, volume 4, Art. 8, 1999.
- [Woo94] T. Woo. Visibility maps and spherical algorithms. *Comput. Aided Design*, 26:6–16, 1994.

56 SOLID MODELING

Christoph M. Hoffmann

INTRODUCTION

The objective of solid modeling is to represent, manipulate, and reason about the 3D shape of solid physical objects, by computer.

Solid modeling is an application-oriented field that has a tradition of implementing systems and algorithms. Major applications include manufacturing, computer vision, graphics, and virtual reality. Technically, the field draws on diverse sources including numerical analysis, symbolic algebraic computation, approximation theory, point set topology, algebraic geometry, and computational geometry.

First, the major representations of solids are reviewed in Section 56.1. They include constructive solid geometry, boundary representation, spatial subdivision, medial surface representations, and procedural representations. Then, major layers of abstraction in a typical solid modeling system are characterized in Section 56.2. The lowest level of abstraction comprises a substratum of basic service algorithms. At an intermediate level of abstraction there are algorithms for larger, more conceptual operations. Finally, a yet higher level of abstraction presents to the user a functional view that is typically targeted toward solid design.

Solid design paradigms work with form features and constraints. Often, they define classes of shape instances, and venture into territory that has yet to be plumbed mathematically and computationally. Concurrently, there is also a shift in the system architecture toward modularized confederations of plug-compatible functional components. We explore these trends lightly in Section 56.3.

Open problems are gathered in Section 56.4.

56.1 MAJOR REPRESENTATION SCHEMATA

GLOSSARY

Solid representation: Any representation allowing a deterministic, algorithmic point membership test.

Constructive solid geometry (CSG): The solid is represented as union, intersection, and difference of primitive solids.

Boundary representation (Brep): The solid surface is represented as a quilt of vertices, edges, and faces.

Spatial subdivision: The solid is decomposed into a set of nonintersecting primitive volumes.

Medial surface transformation: Closure of the locus of centers of maximal inscribed spheres, and a function giving the minimum distance to the solid boundary. Usually called the ***MAT*** for “medial axis transformation.”

Procedural representation: The solid is described by a scripting language or a notational schema that must be evaluated.

A solid representation must allow the unambiguous, algorithmic determination of point membership: given any point $p = (x, y, z)$, there must be an algorithm that determines whether the point is inside, outside, or on the surface of the solid. Moreover, restrictions are placed on the topology of the solid and its embedding, excluding, for example, fractal solids.

These restrictions are eminently reasonable. Increasingly, however, solid modeling systems depart from this strict notion of solid and permit representing a mixture of solids, surfaces, curves, and points, for example, in surface modeling in graphics via “particle systems.” The additional geometric structures are useful for certain design processes, for interfacing with applications such as meshing solid volumes, and for abstracting solid features, to name a few.

56.1.1 CONSTRUCTIVE SOLID GEOMETRY

GLOSSARY

Primitive solids: Traditionally: block, sphere, cylinder, cone, and torus. More general primitives are possible.

Sweep: Volume covered by sweeping a solid or a closed contour in space.

Extrusion: Sweep along a straight line segment.

Revolution: Circular sweep.

Regularized Boolean operation: The closure of the interior of a set-theoretic union, intersection, or difference.

Algebraic halfspace: Points such that $f(x, y, z) \leq 0$ where f is an irreducible polynomial.

Irreducible polynomial: Polynomial that cannot be factored over the complex numbers.

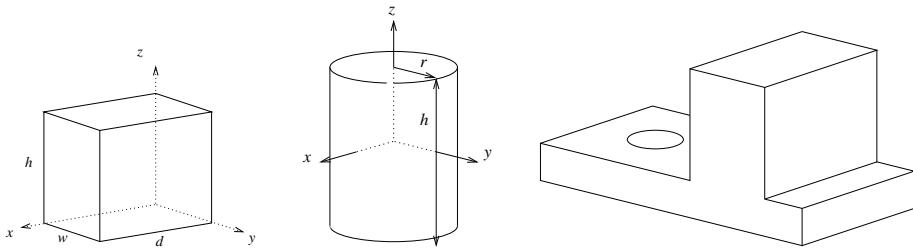
Classical Constructive Solid Geometry (CSG) represents a solid as a set-theoretic Boolean expression of *primitive* solid objects, of a simpler structure. Both the surface and the interior of the final solid are thereby defined, albeit implicitly. The CSG representation is valid if the primitives are valid. A solid’s surface is closed and orientable and encloses a volume. The traditional CSG primitives are block, sphere, cylinder, cone, and torus.

A solid is represented as an algebraic expression that uses rigid motions and regularized set operations. The traditional operations are regularized union, intersection, and difference. A regularized set operation is obtained by taking the closure of the interior of the set-theoretic result. The effect is to obtain solids that do not contain lower-dimensional parts, such as interior (or dangling exterior) faces, edges, and vertices.

Each solid has a default coordinate system that can be changed with a rigid body transformation. A Boolean operation identifies the two coordinate systems of the solids to be combined and makes it the default coordinate system of the resulting solid.

FIGURE 56.1.1

Left and middle: CSG primitives `block(w, d, h)` and `cylinder(r, h)` with default coordinate systems. Right: T-bracket as union of two blocks minus a cylinder.



As an example, consider Figure 56.1.1. Using the coordinate system conventions shown, the CSG representation of the bracket is the expression

$$\begin{aligned} \text{block}(8, 3, 1) \cup^* \text{move}(\text{block}(2, 2.5, 3), (0, 4.5, 1)) \\ -^* \text{move}(\text{cylinder}(0.75, 1), (1.5, 1.5, -0.5)) \end{aligned}$$

where the * indicates a regularized operation. (See also [Figure 38.4.1](#).)

The basic operations one wishes to perform on CSG representations are classifying points, curves, and surfaces with respect to a solid; detecting redundancies in the representation; and approximating CSG objects systematically.

More general primitives are obtained by considering the volume covered by sweeping a solid along a space curve, or sweeping a planar contour bounding an area. Defining a sweep is delicate, requiring many parameters to be exactly defined, but simple cases are widely used. They are extrusion, i.e., sweep along a straight line; and revolution, i.e., a sweep about an axis. The evaluation of general sweeps can be accomplished by a number of methods. An even more general set of primitives is algebraic halfspaces, point sets defined by

$$P = \{(x, y, z) \in \mathbb{R}^3 \mid f(x, y, z) \leq 0\},$$

where $f(x, y, z)$ is an irreducible polynomial in x , y , and z .

More general operations are obtained by using nonregularizing Boolean operations or by defining a nonstandard semantics for Boolean operations on surfaces and curves.

56.1.2 BOUNDARY REPRESENTATION

In boundary representation (Brep), the solid surface is represented as a quilt of faces, edges, and vertices. A distinction is drawn between the topological entities, vertex, edge, and face, related to each other by incidence and adjacency, and the geometric location and shape of these entities. See also [Figure 56.1.2](#). For example, when polyhedra are represented, the faces are polygons described geometrically by a face equation plus a description of the polygon boundary. Geometrically, the entities in a Brep are not permitted to intersect anywhere except in edges and vertices that are explicitly represented in the topology data structure. In addition to the classification operations mentioned for CSG, Boolean union, intersection, and difference operations are usually implemented for Brep systems. Both regularized and nonregularizing Boolean operations may occur.

Different Brep schemata appear in the literature, divided into two major families. One family restricts the solid surfaces to oriented manifolds. Here, every edge

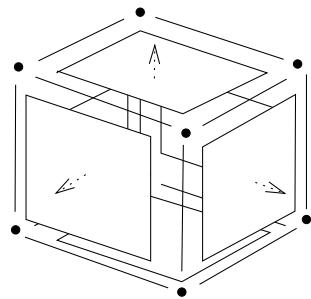


FIGURE 56.1.2

Topological entities of a box. Adjacency and incidence are recorded in Brep. Dotted arrows indicate face orientation.

is incident to two faces, and every vertex is the apex of a single cone of incident edges and faces. The second family of Brep schemata allows oriented nonmanifolds in which edges are adjacent to an even number of faces. When these faces are ordered radially around the common edge, consecutive face pairs alternatingly bound solid interior and exterior. See Figure 56.1.3 for examples.

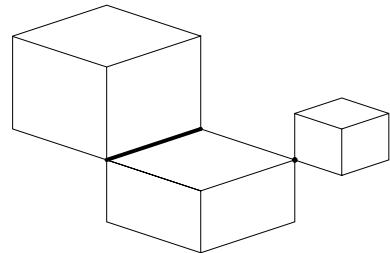


FIGURE 56.1.3

A nonmanifold solid without dangling or interior faces, edges, and vertices; the nonmanifold edges and vertices are drawn with a thicker pen.

More general nonmanifold Breps are used in systems that combine surface modeling with solid modeling. In such representation schemata, a solid may have interior (two-sided) faces, dangling edges, and so on. The current trend is to incorporate surface modeling capabilities into solid modelers.

The topology may be restricted in other ways. For instance, the interior of a face may be required to be homeomorphic to a disk, and edges required to have two distinct vertices. In that case, the Brep of a cylinder would have four faces, two planar and two curved. This may be desirable because of the geometric surface representation, or may be intended to simplify the algorithms operating on solids.

56.1.3 SPATIAL SUBDIVISION REPRESENTATIONS

GLOSSARY

Boundary conforming subdivision: Spatial subdivision of a solid that represents the boundary of the solid exactly.

Boundary approximating subdivision: Spatial subdivision that represents the boundary of the solid only approximately.

Regular subdivision: A subdivision whose cells are congruent. Grids are regular subdivisions.

Irregular subdivision: A subdivision with noncongruent cells.

Octree: Recursive selective subdivision of a cuboid volume into eight subcuboids.

Binary space partition (BSP) tree: Recursive irregular subdivision of space, traditionally by halfplanes. See also [Sections 28.8.2](#) and [38.5](#).

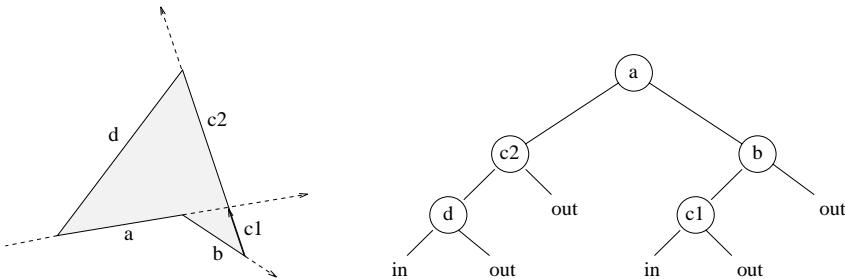
Spatial subdivision decomposes a solid into cells, each with a simple topological structure and often also with a simple geometric structure. Subdivision representations are divided into *boundary conforming* and *boundary approximating*.

Important boundary conforming subdivision schemata are *meshes* and the *BSP tree*. Mesh representations are used in finite element analysis, a method for solving continuous physical problems. The mesh elements can be geometric tetrahedra, hexahedra, or other simple polyhedra, or they can be deformations of topological polyhedra so that curved boundaries can be approximated exactly. See [Sections 25.4–5](#).

Binary space partition trees are recursive subdivisions of 3-space. Each interior node of the tree separates space into two disjoint point sets. In the simplest case, the root denotes a separator plane. All points of \mathbb{R}^3 below or on the plane are represented by one subtree, all points above the plane are represented by the other subtree. The two point sets are recursively subdivided by halfplanes at the subtree nodes. The leaves of the tree represent cells that are labeled IN or OUT. The (half) planes are usually face planes of a polyhedron, and the union of all cells labeled IN is the polyhedron. For an example in \mathbb{R}^2 see Figure 56.1.4. Note that algebraic halfspaces can be used as separators, so that curved solids can be represented exactly.

FIGURE 56.1.4

A polygon and a representing BSP tree.



Boundary approximating representations are *grids* and *octrees*. In grids, space is subdivided in conformity with a coordinate system. For Cartesian coordinates, the division is into hexahedra whose sides are parallel to the coordinate planes. In cylindrical coordinate systems, the division is into concentric sectors, and so on. The grids may be regular or adaptive, and may be used to solve continuous physical problems by differencing schemes. Rectilinear grids that are geometrically deformed can be boundary-conforming. Otherwise, they approximate curved boundaries.

An octree divides a cube into eight subcubes. Each subcube may be further subdivided recursively. Cubes and their subdivision cubes are labeled white, black, or grey. A grey cube is one that has been subdivided and contains both white and black subcubes. A subcube is black if it is inside the solid to be represented, white if it is outside. Quadtrees, the two-dimensional analogue of octrees, are used in many geographical information systems. See [Figure 38.5.1](#).

56.1.4 MEDIAL SURFACE REPRESENTATIONS

GLOSSARY

MAT: Medial axis transform, the two-dimensional version of the medial surface representation. Some authors use “medial axis transform” regardless of the dimension of the domain.

Maximal inscribed disk: Disk inscribed in a domain and not properly contained in another inscribed disk.

Medial axis and medial surface can unambiguously represent two-dimensional domains and 3D solids, respectively. The representations are not widely used for this purpose at this time; more frequently they are used for shape recognition (see [Section 51.4](#)). However, as explained below, some sophisticated meshing algorithms are based on the medial axis and the medial surface.

The medial axis of a two-dimensional domain is defined as the closure of the locus of centers of disks inscribed within the domain. A disk is maximal if no other disk properly contains it. An example is shown in Figure 56.1.5 along with some maximal disks.

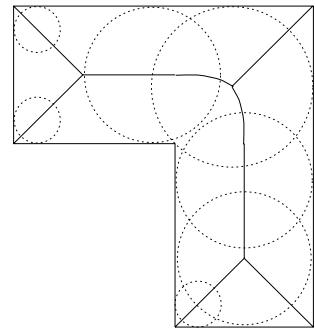


FIGURE 56.1.5

L-shaped domain and associated medial axis. Some maximal inscribed circles contributing to the medial axis are shown.

The medial surface of a solid is the closure of the locus of centers of maximal inscribed spheres. When we know the radius (the limit radius in case of closure points) of the corresponding sphere for each point on the medial surface, then an unambiguous solid representation is obtained that is sometimes called the medial axis transform (MAT). The MAT has a number of intriguing mathematical properties. For example, by enlarging the radius values by a constant, the MAT of a dilatation of the solid is obtained.

Originally, solid modeling has investigated the MAT for the purpose of constructing shell solids (obtained by subtracting a small inset), for organizing finite element meshing algorithms, and for recognizing form features. More recently, the role of the MAT in surface reconstruction has begun to impact solid modeling; see [Chapter 30](#). Surface reconstruction arises in solid modeling for its application in reverse engineering where a model is to be constructed from a physical object by an automated measuring strategy.

56.1.5 PROCEDURAL REPRESENTATIONS

Procedural solid representations fall into two families: script language representations that have a strong programming language character, and descriptive representations evaluated by a program or system.

The PADL system used FORTRAN as script language to specify solids. CSG expressions and directives were embedded into the Fortran program. The solid was evaluated into an internal format. Alpha_1 originally used Lisp as script language and evaluated the solid so described into a boundary representation. Subsequently, a direct manipulation interface was added to the system. The recent SGDL system uses Scheme as script language, evaluating it into an internal proprietary data format. Since such script languages are based on a general programming language, the solid evaluation can be highly complex and may include any computation. Unless the evaluated solid is represented in one of the other representation schemata, it is in general not possible to reason about solids using the procedural representation directly.

Descriptive representations, including the Erep notation are data representations by nature. Their procedural nature derives from the need to evaluate and instantiate parameters, based on (computed) geometric relationship and, in many cases, geometric constraints. Once the parameters are determined, the shape is evaluated in steps, where the major steps typically correspond to form features. Usually, an entire family of solids can be so described and instance solids are obtained by evaluating parameters and dimensional constraints. A semantic characterization of the family remains largely an open problem, as discussed later.

56.1.6 CONVERSION BETWEEN REPRESENTATIONS

Most solid modeling systems use Brep. Conversion from CSG to Brep is well understood and is implemented as regularized Boolean operations on Brep solids. An extensive literature addresses these complex algorithms.

The conversion from Brep to CSG is not completely understood. In the polyhedral case, the conversion is essentially the same as the conversion from Brep to BSP tree. Pure CSG solids, using the PADL primitives, can also be converted. Conversion involving higher degree surfaces is largely open.

Some progress has been made by Naylor and Rogers in the case of Bézier curves and B-splines (for definitions, see [Section 53.1](#)). Roughly speaking, a coarse BSP tree is constructed that encloses sections of the curve in convex polygonal regions. On demand, the tree can be extended dynamically, thereby refining the enclosing regions. In this way, points may be classified efficiently with respect to the curve to a required resolution.

There are several algorithms for converting from CSG or Brep to the MAT. Some are based on geometric principles, some on a Delaunay triangulation of an approximated boundary, and some on a grid subdivision of ambient space. Because simple boundary geometry elements can produce very complicated curves and surface elements in the MAT, approximation approaches are favored in practice. The conversion from MAT to Brep has been addressed by Vermeer [Ver94] and later by Amenta [ACK01]. Note that a polyhedral MAT produces a solid boundary that can contain spherical, conical, and cylindrical elements.

The conversion from CSG or Brep to mesh representations is a partially solved problem when the conversion is done for finite element analysis or other numerical treatment of continuum problems. In that context, the problem is not a geometric problem alone: the quality of the subdivision must also be judged by nongeometric criteria that derive from the nature of the physical problem and the numerical algorithm used to solve it. Many approaches are based on octree subdivision, on Delaunay triangulation, and on MAT computations.

The conversion relationships are summarized in Table 56.1.1.

TABLE 56.1.1 Representation conversion.

CONVERSION	REMARKS
CSG → Brep	Many methods, e.g., [Chi88, Hof89, Män88]. Active research seeks better tradeoff between speed, accuracy, and geometric coverage.
Brep → CSG	Largely open. Polyhedral case similar to BSP tree construction [Hof93b]; quadric cases treated in [Sha91, SV93]. See also [NR95] for parametric case.
Brep, CSG → MAT	[CHL91] uses grid approximation, [SAR95] uses Delaunay approximation of domain.
MAT → Brep	[Ver94] converts polyhedral MAT.
Brep, CSG → spatial subdivision	Many approaches; see, e.g., [Hof95, TWM85, SERB99]. Active research seeks improved techniques.

GEOMETRIC COVERAGE

The range and geometric representation of solid surfaces is referred to as *geometric coverage*. Polyhedral modeling restricts to planes. Classical CSG allows only planes, cones, cylinders, spheres, and tori. Experimental modelers have been built allowing arbitrary algebraic halfspaces. SGDL uses implicit algebraic surfaces of degree up to 4.

Most commercial and many research modelers use B-splines (uniform or nonuniform, nonrational or rational) or Bézier surfaces. The properties and algorithmic treatment of these surfaces is studied by computer-aided geometric design. See Chapter 53, as well as the monographs and surveys [Far88, Hos92, HL93].

Subdivision surfaces have also been proposed but, despite their success in graphics, have thus far not gained wide acceptance in solid modeling. There are many connections between certain kinds of subdivision curves and surfaces and certain classes of spline curves and surfaces. See also Chapter 53.

SPATIAL RELATIONSHIPS

In many applications one would like to understand spatial relationships. Some of the solid representations reviewed have been considered for this purpose. For instance, the MAT has been used to guide meshing algorithms globally and some attempts have been made to devise simplifications for isolating specific features of a shape. Attempts have been made to define suitable simplifications and variations of the MAT; e.g., [FLM03].

Shape simplification ([Chapter 54](#)) is fundamental for many tasks, including in collision detection ([Chapter 35](#)). When a shape is offset by a large distance, smaller features tend to disappear; hence offsetting, a close relative of the MAT, can be used to explore shape simplification [BDG97]. Other approaches have constructed hierarchical representations in which shape is approximated by a hierarchy of simple bounding volumes that at the tree root enclose the entire shape, and in the interior refine the shape estimate by alternatingly subtracting and adding smaller bounding volumes; e.g., [GLM96, KGL⁺98]. Such trees of bounding volumes have similarity with CSG trees.

56.2 LEVELS OF ABSTRACTION

GLOSSARY

Substratum: Basic computational primitives of a solid modeler, such as incidence tests, vector arithmetic, etc.

Algorithmic infrastructure: Major algorithms implementing conceptual operations, such as surface intersection, edge blending, etc.

Graphical user interface (GUI): Visual presentation of the functionality of the system.

Application procedural interface (API): Presentation of system functionality in terms of methods and routines that can be included in user programs.

Substratum problem: Unreliability of logical decisions based on floating-point computations.

Large software systems should be structured into layers of abstraction. Doing so simplifies the implementation effort because the higher levels of abstraction can be compactly programmed in terms of the functionality of the lower levels. Thereby, the complexity of the system is reduced. A solid modeling system spans several levels of abstraction:

1. On the lowest level, there is the substratum of arithmetic and symbolic computations that are used as primitives by the algorithmic infrastructure. This level contains point and vector manipulation routines, incidence tests, and so on.
2. Next, there is an intermediate level comprising the algorithmic infrastructure. This level implements the conceptual operations available in the user interface, as well as a wide range of auxiliary tools needed by these operations. There is often an application programming interface available with which programs can be written that use the algorithmic infrastructure of the modeling system.
3. A graphical user interface (GUI) presents to the user a view of the functional capabilities of the system. Interaction with the GUI exercises these functions, for instance, for solid design. Tools for editing and archiving solids are included.

Ideally, the levels of abstraction should be kept separate, with the higher levels leveraging the functionality of the lower levels. However, this separation is fundamentally limited by the interaction of numeric and symbolic computation.

56.2.1 THE SUBSTRATUM

The substratum consists of many low-level computations and tests; for example, vector computations, simple incidence tests, and computations for ordering points along a simple curve in space. Ideally, these operations create an abstract machine whose functionality simplifies the algorithms at the intermediate level of abstraction. But it turns out that this abstract machine is unreliable in a subtle way when implemented using floating-point arithmetic. Exact arithmetic would remedy this unreliability, but is held by many to be unacceptably inefficient when dealing with solids that have curved boundaries. See [Section 41.4](#). Problems include input accuracy.

To illustrate how inexact arithmetic at the substratum level can impact the geometric computation, consider modeling polyhedral solids, the simplest possible situation for solid modeling. All computational decisions that arise in the course of a regularized Boolean operation on polyhedra can be reduced to determining the sign of 4×4 determinants. Geometrically, this is a test of whether a point is above, on, or below a plane. When the determinant's value is nearly zero, floating-point evaluation will decide based on a tolerance. But the decision is unreliable because logically equivalent tests may arise as different determinants in the course of the algorithm: some of the determinants could have small, others large values, thus necessitating different tolerances to arrive at consistent decisions. This gives an opportunity for the algorithm to build inconsistent data structures and fail. The problems are magnified when dealing with curved solids.

Recent academic solid modeling systems adopt exact arithmetic either outright, or use exact arithmetic on demand. In the latter approach, an error bound is evaluated along with the predicate on whose value a logical decision depends. If the decision is unambiguous based on floating-point arithmetic, no further action is taken. Otherwise, an exact evaluation is done. If an exact evaluation is to be made, the input is understood to be exact as given, and the predicate must be evaluated from the input data without using intermediate, possibly inaccurate, data. The assumption of exact input data is problematic for Brep solids. Unless the input solid is very simple, or it was computed using exact arithmetic, it is an open problem how to interpret the data such that a valid solid is obtained. For a deeper evaluation of the problem, and for some approaches to solving it, see [Chapter 41](#).

56.2.2 ALGORITHMIC INFRASTRUCTURE

Algorithmic infrastructure is a prominent research subject in solid modeling. Among the many questions addressed is the development of efficient and robust algorithms for carrying out the geometric computations that arise in solid modeling. The problems include point/solid classification, computing the intersection of two solids, determining the intersection of two surfaces, interpolating smooth surfaces to eliminate sharp edges on solids, and many more. See the reference section for a sampling of the literature.

Recent academic work considers structuring *application procedural interfaces*

(API's) that encapsulate the functional capabilities of solid modelers so they can be used in other programs; [ABC⁺00]. Such API's play a prominent role in applications because they allow building on existing software functionality and constructing different abstraction hierarchies than the one implemented by a full-service solid modeling system. The work attempts to give a system-independent specification of basic API functionality for solid modeling.

An important consideration when devising infrastructure is that the algorithms are often used by other programs, whether or not there is an API. Therefore, they must be ultra-reliable and in most cases must not require user intervention for exceptional situations.

The major geometric computations implemented at the infrastructure level have to balance the conflicting goals of efficiency, accuracy, and robustness. For this reason, many operations continue to be researched in efforts to seek new perceived optima. Moreover, new variants of surface representations continue to be devised that necessitate different approaches. Some of the major operations on which research continues are the following.

Surface intersection. Given two bounded areas of two surfaces, determine all intersection curve components. A major difficulty of the problem is to identify correctly all components of the intersection, including isolated points and singularities. Since this computation is done in \mathbb{R}^3 , classical algebraic geometry is of limited help. The other difficulty is to address properly the substratum unreliabilities.

Surface intersection remains a key problem with continuing attempts at balancing efficiency, accuracy, and stability of the algorithms.

Offsetting. Given a surface, its *offset* is the set of all points that have fixed minimum distance from the surface. Offsets can have self-intersections that must be culled, and there is a technical relationship between offsetting and forming the MAT. Namely, when offsetting a curve or surface by a fixed distance, the self-intersections must lie on the medial axis. Offsetting is used to determine certain blending surfaces, and is also used in the solid operation of *shelling* that creates thin-walled solids.

Blending. Given two intersecting surfaces, a third surface is interpolated between them to smooth the intersection edge. A simple example is shown in [Figure 56.2.1](#). A locally convex blend surface is often called a *round*, and a locally concave one a *fillet*. The blend surface in Figure 56.2.1 is a fillet.

Blending has been considered almost since the beginning of solid modeling, and some intuitive and interesting techniques have been developed over the years. For example, consider blending two primary surfaces f and g . Roll a ball of fixed radius r along the intersection such that it maintains contact with both f and g . Then the surface of the volume swept by the ball can be used as a blending surface, suitably trimmed. Note that the center of the ball lies on the intersection of the offsets, by r , of both f and g . In more complicated schemes the radius of the ball is varied along the intersection.

A less well-understood issue for blending solids arises from the global problem of how to devise the contact curves and blending surfaces, so that the surfaces connect properly at adjacent faces, behave correctly at vertices, and so on. [Figure 56.2.2](#) shows the problem of overlapping blends. The fillet and round

constructed separately do not meet in the region of overlap. The problem is that then there is no closed surface defining the blended solid. A resolution could modify the round, or the fillet, or could insert a separate surface in the overlap region after suitably cutting back both primary blends.

When the primary surfaces meet at a vertex tangentially, blending surfaces must “dissipate.” Figure 56.2.3 shows several methods to dissipate round and

FIGURE 56.2.1

Left: two cylinders intersecting in a closed edge. Right: edge blended with a constant-radius, rolling-ball blend; the bounding curves of the blend are shown.

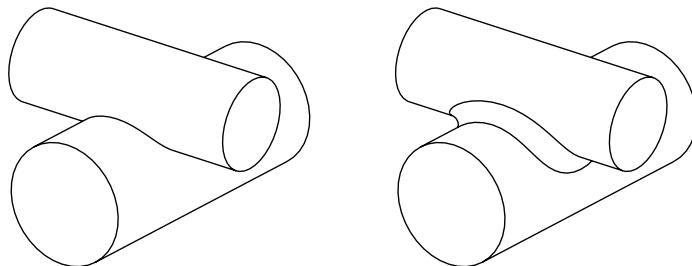


FIGURE 56.2.2

Global blend interference [Bra97]: The round of the front edge overlaps with the fillet of the cylinder edge on top (left). Without further action, the two blends do not connect, leaving a gap in the surface. The solution shown in the middle modifies the front round. Other possibilities include modifying the fillet or inserting a separate blend in the overlap region (right).

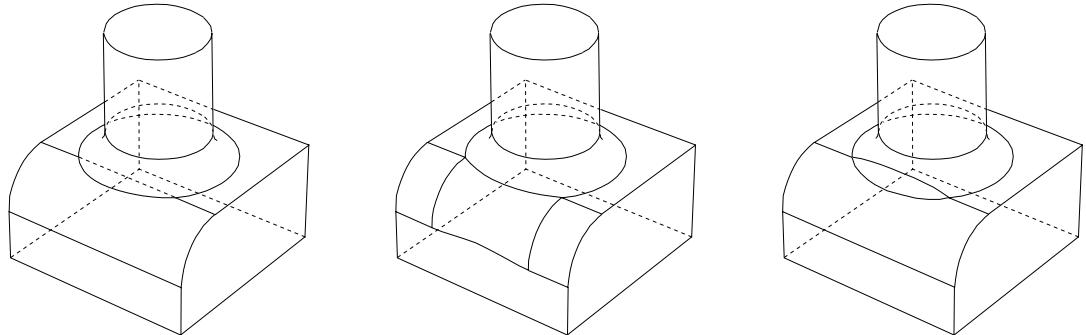
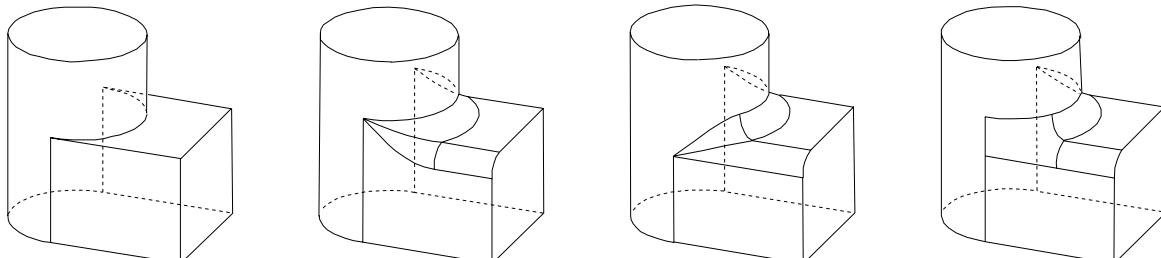


FIGURE 56.2.3

Global blend interference [Bra97]: At ending vertex, the round and the fillet must be merged into a compatible structure. Several solutions are illustrated.



fillet at the end vertices. The examples are from [Bra97] and point out the dimensions of the global problem.

Deformations. Given a solid body, deform it locally or globally. The deformation could be required to obey constraints such as preserving volume or optimizing physical constraints. For example, we could deform the basic shape of a ship hull to minimize drag in fluids of various viscosities.

Shelling. Given a solid, hollow out the volume so that a thin-wall solid shape remains whose outer surface is part of the boundary of the input solid. The wall thickness is a parameter of the operation. Variations include designating parts of the solid surface as “open.” For instance, taking a solid cylinder and designating both flat end faces as open the operation creates a hollow tube of the same outside diameter. Conceptually, the operation subtracts an inset of the solid, obtained by shrinking the original solid, an offset operation.

56.2.3 USER INTERFACES

Ultimately, the functional capabilities of a solid modeling system have to be presented to a user, typically through a *graphical user interface* (GUI). It would be a mistake to dismiss GUI design as a simple exercise. If the GUI merely presents the functionality of the infrastructure literally, an opportunity for operational leveraging has been lost. Instead, the GUI should conceptualize the functionalities an application needs. As in programming language design, this conceptual view can be convenient or inconvenient for a particular application. Research on GUI’s therefore is largely done with a particular application area in mind.

For example, in mechanical engineering product design, an important aspect of the GUI might be to allow the user to specify the shape conveniently and precisely. This might be accomplished using geometric constraints and constraints of length, radius, and angle. In GUI’s for virtual environment definition and navigation, on the other hand, approximate constraints and direct manipulation interfaces would be better.

56.3 FEATURES AND CONSTRAINTS

GLOSSARY

Form feature: Any stereotypical shape detail that has application significance.

Geometric constraint: Prescribed distance, angle, collinearity, concentricity, etc.

Generic design: Solid design with constraints and parameters without regard to specific values.

Design instance: Resulting solid after substituting specific values for parameters and constraints.

Parametric constraint solving: Solving a system of nonlinear equations that has a fixed triangular structure.

Variational constraint solving: Solving a system of nonlinear simultaneous equations.

In solid modeling, two design paradigms have become standard for manufacturing applications, *feature-based design* and *constraint-based design*. The new paradigms expose a need to reconsider solid representations at a different level of abstraction. The representations reviewed before are for individual, specific solids. However, we need to represent entire *classes* of solids, comprising a generic design. Roughly speaking, solids in a class are built structurally in the same way, from complex shape primitives, and are instantiated subject to constraints that interrelate specific shape elements and parameters. How these families should be defined precisely, how each generic design should be represented, and how designs should be edited are all important research issues of considerable depth.

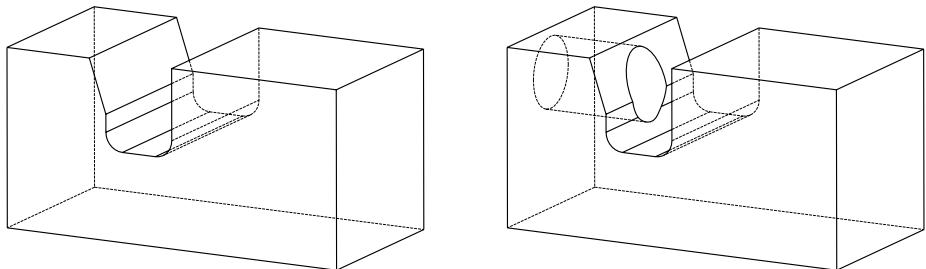
56.3.1 FEATURE-BASED DESIGN

Feature-based design is usually understood to mean designing with shape elements such as slots, holes, pockets, etc., that have significance to manufacturing applications relating to function, manufacturing process, performance, cost, and so on. Focusing on shape primarily, we can conceptualize solid design in terms of three classes of features: generative, modifying, and referencing features. A feature is added to an existing design using attachment attributes and placement conditions. Subsequent editing may change both types of attachment information.

As an example, consider the solid shown to the right in Figure 56.3.1. A hole was added to the design on the left, and this could be specified by giving the diameter of the hole, placing its cross section, a circle, on the side face, and requiring that the hole extend to the next face. Should the slot at which the hole ends be moved or altered by subsequent editing, then the hole would automatically be adjusted to the required extent.

FIGURE 56.3.1

Left: solid block with a profiled slot. Right: After adding a hole with the attribute “through next face,” an edited solid is obtained. If the slot is moved later, the hole will adjust automatically.



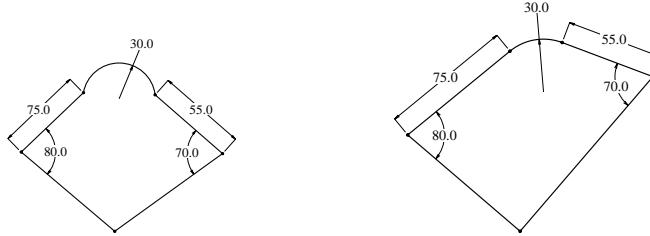
56.3.2 CONSTRAINT-BASED DESIGN

Constraint-based design refers to specifying shape with the help of constraints, when placing features or when defining shape parameters. For instance, assume that we are to design a cross section for use in defining a solid of revolution. A rough topological sketch is prepared (Figure 56.3.2, left), annotated with constraints, and instantiated to a sketch that satisfies the constraints exactly (Figure 56.3.2, right).

Auxiliary geometric structures can be added, such as an axis of rotation. There is an extensive literature on constraint solving, from a variety of perspectives.

FIGURE 56.3.2

Geometric constraint solving. Input to the constraint solver shown on the left. Here, the arc should be tangent to the adjacent segments, and the two other segments should be perpendicular. Output of the constraint solver shown on the right.

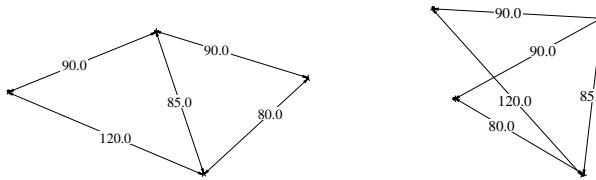


Most solid modeling systems use both features and constraints in the design interface. Often, the constraints on cross sections and other two-dimensional structures are unordered, but the constraints on 3D geometry are usually considered in a fixed sequence. Solving systems of unordered constraints is sometimes referred to as *variational constraint solving*. Mathematically, it is equivalent to solving a system of nonlinear simultaneous equations. Solving constraints in a fixed sequence is also known as *parametric constraint solving*. The latter is equivalent to solving a system of nonlinear equations that has a fixed, triangular structure where each equation introduces a new variable.

A well-constrained geometric constraint problem corresponds naturally to a system of nonlinear algebraic equations with a finite set of solutions. In general, there will be several solutions of a single, well-constrained geometric problem. An example is shown in Figure 56.3.3. This raises the interesting question of exactly how a constraint solver should select one of those solutions efficiently, and why.

FIGURE 56.3.3

The well-constrained geometric problem of placing 4 points by 5 distances has two distinct solutions.



From symbolic computation we know that there are algorithms to convert a nontriangular system of nonlinear equations into a triangular system. The distinction between parametric and variational constraint solving is therefore artificial in theory. However, full-scale triangularization of systems of nonlinear equations is not tractable in many cases, so the distinction is relevant in practice. Moreover, a predetermined sequential evaluation of constraints is simple to implement and can be interfaced easily with conditional constraint evaluation, thereby increasing the expressive power of the constraint system without raising new semantic issues. For these reasons, many developers of solid modeling systems leverage core modeling

capabilities by such (simple) extensions.

Spatial constraint solving is very much more demanding than planar constraint solving. In the planar case, simple (simultaneous) subsystems can be identified and isolated using straightforward graph algorithms, and result in practically important solvers. Furthermore, in the planar case, there are not many such subsystems needed. In contrast, no simple simultaneous spatial subsystems exist. When lines are allowed as geometric primitives, then the systems become very much harder and there are many such subsystems even when restricting to only five or six geometric elements. The number of basic cases number in the hundreds; [GHY02]. This structural barrier seems to preclude the emergence of truly spatial constraint solvers, and with it, of spatial design paradigms. In practice, CAD systems skirt the issue by building interrelated planar constraint problems which are variational in each plane but follow a clear, parametric sequence for elaborating the spatial relationship between the various planar problems.

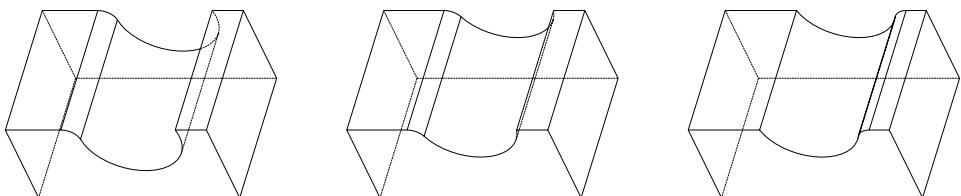
56.3.3 SEMANTIC PROBLEMS

When constraints and parameters are used in solid design, a *generic design* is obtained. Generic designs are instantiated by constraint values, and may be edited by changing the constraint values, the constraint schema, and the feature attributes. A design so edited can then be automatically re-instantiated by the solid modeler. A central difficulty in implementing this scenario, however, is that the generic design is usually defined visually on the basis of a particular instance, and when the design changes, the instance geometry is no longer present. Thus, visually identified instance structures must be suitably described, so that re-instantiation can be carried out correctly.

As an example, consider the solid shown in Figure 56.3.4, left. It was constructed as follows. First, a rectangle was drawn and extruded into a block. On the front face of the block, a circle was drawn as a profile of a slot across the top of the block. Then, an edge was visually identified for rounding. This design is edited by altering the position of the circular slot profile. The edge to be rounded is not an explicit design entity, however. Hence, the edge has to be described implicitly, perhaps by the intersection of the circle and the top edge of the face on which the circle has been drawn. This description does not distinguish between the two straight edges of the slot, however, so additional information has to be used. Such information would have to allow a consistent identification under all possible constraint values, and is called the *persistent naming* problem.

FIGURE 56.3.4

A block with a slot and round on the left edge is shown left. After editing, in this case decreasing the depth of the slot, re-instantiation should produce the solid shown in the middle. However, some systems may re-instantiate as shown to the right, an error.



There has been a small stream of academic work on this topic, although it is of intense interest in applications. In particular, the formalization of the design information has profound implications on system architectures because it formalizes, in effect, the information flow between functional components. Whenever such formalization seeks independence from the specific implementation of the system components, system modularization is facilitated. Ultimately, this will accelerate the current trend of decomposing solid modeling systems into standardized components that can function interchangeably and can be combined in a variety of ways.

56.4 OPEN PROBLEMS

Most major problems in solid modeling contain a conceptualization aspect. That is, a precise, technical formulation of the problem commits to a specific conceptualization of the larger context that may be contentious. For example, consider the following technical problem. *Given an implicit algebraic surface S and a distance d , find the “offset” of S by d .* Assuming a precise definition of offset, and a restriction to irreducible algebraic surfaces S , the problem statement ignores the fact that a solid model is not bounded by a single, implicit surface, and that implicit surfaces of high algebraic degree may cause severe computational problems when used in a solid modeler.

CONSTRAINT SOLVING

Geometric constraint solvers trade efficiency for generality. Some very interesting techniques have been developed for planar problem that are fast but not very general. They could be extended in various ways without substantially impacting on efficiency. Such extensions, for constraint solving in the plane, include the incorporation of parametric curve segments as geometric elements, more general constraint configurations, relations among distances, and angles.

Spatial geometric constraint solving poses a number of open problems, including determining whether a constraint problem is generically well-constrained. The problem of how many lines can be found at prescribed distance from four fixed points has been solved, one of the sequential construction problems for lines. Other construction problems for lines are not completely solved. The smaller simultaneous problems involving points and planes have been solved. Most simultaneous problems involving lines require numerical treatment, however, and are not well understood.

FEATURES

Manufacturing applications need cogent definitions of features to accelerate design. Such definitions ought to be in terms of generic mechanisms of form and of function. Also needed are mapping algorithms interrelating different feature schemata.

A set of features, say those conceptualizing machining a shape from stock, is called a *design view*. In manufacturing applications there are many views, including machining, tolerancing, design view, etc. Work has begun to address the problem of altering a design in one view with an automatic update of the other views. To

do so requires reasoning about shape and is a hard problem. Some approaches have been based on subdividing the shape by superimposing all feature boundaries, and then tracking how the subdivision is affected by changes to one of the features.

SEMANTICS OF CONSTRAINT-BASED DESIGN

A solid shape design in terms of constraints can be changed simply by changing constraint values. To date, all such changes have been specified in terms of the procedures and algorithms that effect the change. What is needed is an abstract definition of shape change under such constraint changes to obtain a semantic definition of generic design and constraint-based editing. Such a definition must be visually intuitive.

MODEL RECTIFICATION

Because of the substratum problem, Brep data structures can be invalid in the sense that the geometric description does not agree fully with the topological description. For instance, there may be small cracks between adjacent faces, the edge between two adjacent faces may not be where the curve description would place it, and so on. This has motivated work to “heal” the defective surface by closing cracks, eliminating overlaps, and so on. Some approaches sew up cracks with smaller faces, and in the case of polyhedra with triangles. Optimal healing is known to be NP-hard.

An intuitive idea is to assign a thickness to faces, edges and vertices, and enlarge the thickness so that the surface closes up. The difficulty is to work out what happens when nonadjacent faces merge into adjacent ones. The natural geometric enlargement creates mathematically difficult surfaces; for instance, the offset surface of an ellipsoid increases the algebraic degree by a factor of 4. So, an interval based approach has also been proposed in which there is no closed-form description of the enlarged geometric elements.

56.5 SOURCES AND RELATED MATERIAL

FURTHER READING

Monographs on solid modeling. Monographs and surveys provide an excellent entry into solid modeling. Major monographs on solid modeling are [Chi88, Hof89, Män88]. Books on the related field of CAGD (computer-aided geometric design) may also contain material on solid modeling but concentrate primarily on curve and surface design and manipulation. Surface interrogation from a solid modeling point of view is explored in [Hos92, PM02].

Solid representations and conversion. There is a large and diverse literature on representations and representation conversion. Classical work focused primarily on the semantic foundations of CSG and Brep and includes [Req77, Wei86]. Maintaining Brep and CSG simultaneously has been explored in [RS00]. The mesh and octree representations are treated in [BN90, Hof95, Sam89a, Sam89b, TWM85], including the associated conversion problems. The medial axis representation of solids, and how to compute with it, are considered in [Hof92, SAR95, Ver94]. Implicit algebraic

halfspaces as solid primitives are discussed in [BDL⁺91]. The conversion between boundary representation and CSG can be considered a generalization of the binary space partition tree and is explored in [Hof93b, Nay90, NR95, Sha91, SV93]. Curve and surface representations, and their manipulation, are the subject of [Far88, Hos92, HL93, PM02]. More specialized treatment of offsets and sweeps is found in [BL90, CHL91]. Procedural script language representations are discussed in [Bro82, UU94, SS01] for PADL, Alpha_1, and SGDL. Data representations that neutrally describe form features and constraints are developed in [HJ92].

Substratum, infrastructure, and user interfaces. The substratum robustness issue is presented in greater depth in Chapter 41; [SI89, For97] explores the use of exact arithmetic in polyhedral modeling. Manocha and Keyser work with exact arithmetic for curved solids; [KKM99a, KKM99b]. A recent survey is found in [Hof01].

Infrastructure work is traditionally quite extensive. Surface intersection is treated in [Hoh92]; this thesis contains an excellent summary of previous work. A recent monograph on the subject is [PM02]. Global solid operations are considered in [BW89, For95, PS95, RSB96]; local solid operations are discussed in [HH87, Pet92]. Much work has been done in blending. The local problem is often addressed in the context of CAGD, and the monographs on that subject contain much material. The global blending problem is treated extensively in [Bra97]. Work in symbolic algebraic computation (Chapter 33) has foundational importance, for instance in regard to converting between surface representations. Some of the applications of symbolic computation are explored in [BCK88, Cho87, Hof90].

Features and constraints. Neither topic is new, so there is a sizable literature on both. The confluence of the two issues in recent solid modeling systems, however, is new. It raises a number of questions that have only recently been articulated and addressed. [SHL92, KRU94] discuss feature work. Constraints are the subject of [BFH⁺95, HV94, Kra92]. The confluence of the two strands and some of the implications are discussed in [HJ92]. Some of the technical issues that must be addressed are explained in [Hof93a, CH95], and there is more work emerging on this subject. In particular, Shapiro and Raghothama propose several criteria for defining a family of solids; [RS02, RS98].

RELATED CHAPTERS

- [Chapter 25: Triangulations and mesh generation](#)
- [Chapter 30: Curve and surface reconstruction](#)
- [Chapter 38: Geometric intersection](#)
- [Chapter 41: Robust geometric computation](#)
- [Chapter 49: Computer graphics](#)
- [Chapter 53: Splines and geometric modeling](#)

REFERENCES

- [ACK01] N. Amenta, S. Choi, and R.K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Internat. J. Comput. Geom. Appl.*, 19: 127–153, 2001.
- [ABC⁺00] C. Armstrong, A. Bowyer, S. Cameron, J. Corney, G. Jared, R. Martin, A. Middleditch, M. Sabin, and J. Salmon. *Djinn, a Geometric Interface for Solid Modelling*. Information Geometers, Winchester, 2000.

- [BCK88] B. Buchberger, G.E. Collins, and B. Kutzler. Algebraic methods for geometric reasoning. *Annu. Reviews in Computer Science*, 3:85–120, 1988.
- [BDG97] G. Barequet, M.T. Dickerson, and M.T. Goodrich. Voronoi diagrams for polygon-offset distance functions. In *Workshop Algorithms Data Struct.*, pages 200–209, volume 1272 of *Lecture Notes Comput. Sci.*, Springer-Verlag, Berlin, 1997.
- [BDL⁺91] A. Bowyer, J.H. Davenport, D.A. Lavender, P.S. Milne, and A.F. Wallis. The design of a geometric algebra system. In D. Kapur, editor, *Integration of Symbolic and Numeric Methods*. MIT Press, Cambridge, 1991.
- [BFH⁺95] W. Bouma, I. Fudos, C. Hoffmann, J. Cai, and R. Paige. A geometric constraint solver. *Comput. Aided Design*, 27:487–501, 1995.
- [BL90] D. Blackmore and M. Leu. A differential equations approach to swept volume. In *Proc. Rensselaer 2nd Internat. Conf. Computer-Integrated Manuf.*, pages 143–149, Troy, 1990.
- [BN90] P. Brunet and I. Navazo. Solid representation and operation using extended octrees. *ACM Trans. Graph.*, 9:170–197, 1990.
- [Bra97] I. Braid. Non-local blending of boundary models. *CAD*, 29:89–100, 1997.
- [Bro82] C.M. Brown. PADL-2: a technical summary. *IEEE Comput. Graph. Appl.*, 2:69–84, 1982.
- [BW89] M.I.G. Bloor and M.J. Wilson. Generating blending surfaces with partial differential equations. *Comput. Aided Design*, 21:165–171, 1989.
- [CH95] X. Chen and C. Hoffmann. On editability of feature-based design. *CAD*, 27:905–914, 1995.
- [Chi88] H. Chiyokura. *Solid Modeling with Designbase*. Addison-Wesley, Reading, 1988.
- [CHL91] C.-S. Chiang, C. Hoffmann, and R. Lynch. How to compute offsets without self-intersection. In *Proc. SPIE Conf. Curves Surfaces Comput. Vision Graphics*, volume 1610, pages 76–87. Internat. Soc. for Optical Engineering, Bellingham, 1991.
- [Cho87] C.-S. Chou. *Mechanical Theorem Proving*. Reidel, Dordrecht, 1987.
- [Far88] G. Farin. *Curves and Surfaces for Computer-Aided Geometric Design*. Academic Press, Orlando, 1988.
- [FLM03] M. Foskey, M.C. Lin, and D. Manocha. Efficient computation of a simplified medial axis. In *Proc. 8th Annu. ACM Symp. Solid Modeling Appl.*, pages 96–107. ACM Press, New York, 2003.
- [For95] M. Forsyth. Shelling and offsetting bodies. In *Proc. 3rd Annu. ACM Symp. Solid Modeling*. ACM Press, New York, 1995.
- [For97] S.J. Fortune. Polyhedral modeling with multi-precision integer arithmetic. *CAD*, 29:123–133, 1997.
- [GHY02] X.-S. Gao, C. Hoffmann, and W.-Q. Yang. Solving spatial basic geometric constraint configurations with locus intersection. In *Proc. 7th Annu. ACM Symp. Solid Modeling Appl.*, ACM Press, 2002.
- [GLM96] S. Gottschalk, M.C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Proc. ACM Conf. SIGGRAPH 96*, pages 171–180, 1996.
- [HH87] C. Hoffmann and J.E. Hopcroft. The potential method for blending surfaces and corners. In G. Farin, editor, *Geometric Modeling*, pages 347–365. SIAM, 1987.
- [HJ92] C. Hoffmann and R. Juan. Erep, an editable, high-level representation for geometric design and analysis. In P. Wilson, M. Wozny, and M. Pratt, editors, *Geometric Modeling for Product Realization*, pages 129–164. North Holland, Amsterdam, 1992.

- [HL93] J. Hoschek and D. Lasser. *Comput. Aided Geom. Design*. A.K. Peters, Wellesley, 1993.
- [Hof89] C. Hoffmann. *Geometric and Solid Modeling*. Morgan Kaufmann, San Francisco, 1989.
- [Hof90] C. Hoffmann. Algebraic and numerical techniques for offsets and blends. In S. Micali, M. Gasca, and W. Dahmen, editors, *Computations of Curves and Surfaces*, pages 499–528. Kluwer Academic, Dordrecht, 1990.
- [Hof92] C. Hoffmann. Computer vision, descriptive geometry, and classical mechanics. In B. Falcidieno and I. Herman, editors, *Computer Graphics and Mathematics*, Eurographics Series, pages 229–244. Springer-Verlag, Berlin, 1992.
- [Hof93a] C. Hoffmann. On the semantics of generative geometry representations. In *Proc. 19th ASME Design Automation Conf.*, volume 2, pages 411–420, 1993.
- [Hof93b] C. Hoffmann. On the separability problem of real functions and its significance in solid modeling. In *Computational Algebra*, pages 191–204. Marcel Dekker, New York, 1993. *Lecture Notes Pure Appl. Math.*, 151.
- [Hof95] C. Hoffmann. Geometric approaches to mesh generation. In I. Babuska, J. Flaherty, W. Henshaw, J.E. Hopcroft, J. Oliker, and T. Tezduyar, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*. Springer-Verlag, Berlin, 1995.
- [Hof01] C. Hoffmann. Robustness in geometric computations. *J. Comput. Info. Sci. Engr.*, 1:143–155, 2001.
- [Hoh92] M. Hohmeyer. *Surface Intersection*. Ph.D. thesis, Univ. California, Berkeley, Dept. Comput. Sci., 1992.
- [Hos92] M. Hosaka. *Modeling of Curves and Surfaces in CAD/CAM*. Springer-Verlag, New York, 1992.
- [HV94] C. Hoffmann and P. Vermeer. Geometric constraint solving in R^2 and R^3 . In D.Z. Du and F. Hwang, editors, *Computing in Euclidean Geometry*, second edition. World Scientific, Singapore, 1994.
- [KGL⁺98] S. Krishnan, M. Gopi, M.C. Lin, D. Manocha, and A. Pattekar. Rapid and accurate contact determination between spline models using ShellTrees. *Comput. Graph. Forum*, 17:C315–C326, 1998.
- [KKM99a] J. Keyser, S. Krishnan, and D. Manocha. Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: I—representations. *CAGD*, 16:841–859, 1999.
- [KKM99b] J. Keyser, S. Krishnan, and D. Manocha. Efficient and accurate B-rep generation of low degree sculptured solids using exact arithmetic: II—computation. *CAGD*, 16:861–882, 1999.
- [Kra92] G. Kramer. *Solving Geometric Constraint Systems*. MIT Press, Cambridge, 1992.
- [KRU94] F.-L. Krause, E. Rieger, and A. Ulbrich. Feature processing as kernel for integrated CAE systems. In *Proc. IFIP Internat. Conf.: Feature Modeling Recogn. Advanced CAD/CAM Systems Vol II*, pages 693–716, Valenciennes, 1994.
- [Män88] M. Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, 1988.
- [Nay90] B. Naylor. Binary space partitioning trees as an alternative representation of polytopes. *Comput. Aided Design*, 22, 1990.
- [NR95] B. Naylor and L. Rogers. Constructing binary space partitioning trees from piecewise Bézier curves. In *Proc. Graphics Interface*, pages 181–191, 1995.
- [UU94] University of Utah. *Alpha_1 advanced experimental CAD modeling system*, 1994. <http://www.cs.utah.edu/gdc/projects/alpha1/>.

- [Pet92] J. Peters. Joining smooth patches around a vertex to form a C^k surface. *Comput. Aided Geom. Design*, 9:387–411, 1992.
- [PM02] N.M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer-aided Design and Manufacture*. Springer-Verlag, Berlin, 2002.
- [PS95] A. Pasko and V. Savchenko. Algebraic sums for deformation of constructive solids. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling*. ACM Press, New York, 1995.
- [Req77] A. Requicha. Mathematical models of rigid solids. Tech. Rep. PAP Tech. Memo 28, Univ. Rochester, 1977.
- [RS98] S. Raghotama and V. Shapiro. Boundary representation deformation in parametric solid modeling. *ACM Trans on Graphics*, 17:259–286, 1998.
- [RS00] S. Raghotama and V. Shapiro. Consistent updates in dual representation systems. *CAD*, 32:463–477, 2000.
- [RS02] S. Raghotama and V. Shapiro. Topological framework for part families. In *Proc. ACM Sympos. Solid Modeling and Applic*, pages 1–12, 2002.
- [RSB96] A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solids. *IEEE Trans. Visualization Comput. Graph.*, 2:19–27, 1996.
- [Sam89a] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, Reading, 1989.
- [Sam89b] H. Samet. *Design and Analysis of Spatial Data Structures: Quadtrees, Octrees, and Other Hierarchical Methods*. Addison-Wesley, Reading, 1989.
- [SAR95] D. Sheehy, C. Armstrong, and D. Robinson. Computing the medial surface of a solid from a domain Delaunay triangulation. In *Proc. 3rd Annu. ACM Sympos. Solid Modeling*, pages 201–212, 1995.
- [SERB99] A. Sheffer, M. Etzion, A. Rappoport, and M. Bercovier. Hexahedral mesh generation using the embedded voronoi graph. *Engineering with Computers*, 15:248–262, 1999.
- [Sha91] V. Shapiro. *Representations of Semialgebraic Sets in Finite Algebras Generated by Space Decompositions*. Ph.D. thesis, Cornell Univ., Sibley School Mech. Engr., 1991.
- [SHL92] J. Shah, D. Hsiao, and J. Leonard. A systematic approach for design-manufacturing feature mapping. In P. Wilson, M. Wozny, and M. Pratt, editors, *Geometric Modeling for Product Realization*, pages 205–222. North Holland, Amsterdam, 1992.
- [SI89] K. Sugihara and M. Iri. A solid modeling system free from topological inconsistency. *J. Information Processing*, 12:380–393, 1989.
- [SV93] V. Shapiro and D. Vossler. Separation for boundary to CSG conversion. *ACM Trans. Graph.*, 12:35–55, 1993.
- [SS01] SGDL Systems. *The SGDL language*, 2001. <http://www.sgdl-sys.com>.
- [TWM85] J. Thompson, Z. Warsi, and W. Mastin. *Numerical Grid Generation*. North Holland, Amsterdam, 1985.
- [Ver94] P. Vermeer. *Medial Axis Transform to Boundary Representation Conversion*. Ph.D. thesis, Purdue Univ., 1994. Comput. Sci.
- [Wei86] K. Weiler. *Topological Structures for Geometric Modeling*. Ph.D. thesis, Rensselaer Polytechnic Inst., Comput. Syst. Engr., 1986.

57 COMPUTATION OF ROBUST STATISTICS: DEPTH, MEDIAN, AND RELATED MEASURES

Peter J. Rousseeuw and Anja Struyf

INTRODUCTION

As statistical data sets grow larger and larger, the availability of fast and efficient algorithms becomes ever more important in practice. Classical methods are often easy to compute, even in high dimensions, but they are sensitive to outlying data points. Robust statistics develops methods that are less influenced by abnormal observations, often at the cost of higher computational complexity. Many robust methods, especially those based on ranks, are closely related to geometric or combinatorial problems. An early overview of relations between statistics and geometry was given in [Sha76].

Recently many other (mostly multivariate) statistical methods have been developed that have a combinatorial or geometric character and are computationally intensive. Techniques of computational geometry appear to be very well suited for the development of fast algorithms. Over the last decade, the notion of statistical depth especially received considerable attention from the computational geometry community. We mainly concentrate on depth and multivariate medians in this chapter, and in Section 57.3 we list other areas of statistics where computational geometry has recently been of use in constructing efficient algorithms.

57.1 MULTIVARIATE RANKING

A data set consisting of n univariate points is usually ranked in ascending or descending order. Univariate order statistics (i.e., the ‘ k th smallest value out of n ’) and derived quantities have been studied extensively. The median is defined as the order statistic of rank $(n + 1)/2$ when n is odd, and as the average of the order statistics of ranks $n/2$ and $(n + 2)/2$ when n is even. The median and any other order statistic of a univariate data set can be computed in $O(n)$ time. Generalization to higher dimensions is, however, not straightforward.

Alternatively, univariate points may be ranked from the outside inward by assigning the most extreme data points depth 1, the second smallest and second largest data points depth 2, etc. The deepest point then equals the usual median of the sample. The advantage of this type of ranking is that it can be extended to higher dimensions more easily. This section gives an overview of several possible generalizations of depth and the median to multivariate settings. A comprehensive survey of statistical applications of multivariate data depth may be found in [LPS99].

GLOSSARY

Bagplot: Bivariate generalization of the boxplot based on depth regions.

Breakdown value: The smallest fraction of contaminated data points that can move the estimator arbitrarily far away.

Centerpoint: Any point with halfspace depth $\geq \lceil n/(d+1) \rceil$.

Deepest fit: Median hyperplane based on regression depth.

Depth: The outside-inward “rank” of a point (not necessarily a data point).

Depth region: The set of all points with depth $\geq k$ is called the k th depth region D_k .

Median: The point with maximal depth. When this point is not uniquely defined, the median is taken to be the centroid of the depth region with highest depth.

Tukey median: Median based on halfspace depth.

HALFSPACE LOCATION DEPTH

Let $X_n = \{x_1, \dots, x_n\}$ be a finite set of data points in \mathbb{R}^d . The *Tukey depth* or *halfspace depth* (introduced by [Tuk75] and further developed by [DG92]) of any point θ in \mathbb{R}^d (not necessarily a data point) determines how central the point is inside the data cloud. The halfspace depth of θ is defined as the minimal number of data points in any closed halfspace determined by a hyperplane through θ :

$$ldepth(\theta; X_n) = \min_{\|\mathbf{u}\|=1} \#\{i; \mathbf{u}^\top \mathbf{x}_i \geq \mathbf{u}^\top \theta\}.$$

Thus, a point lying outside the convex hull of X_n has depth 0, and any data point has depth at least 1. Figure 57.1.1 illustrates this definition for $d = 2$.

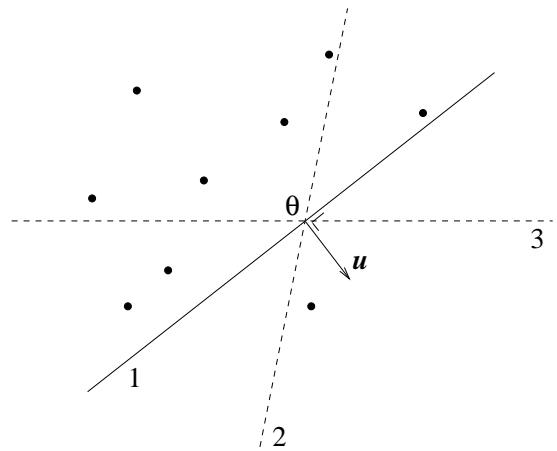


FIGURE 57.1.1

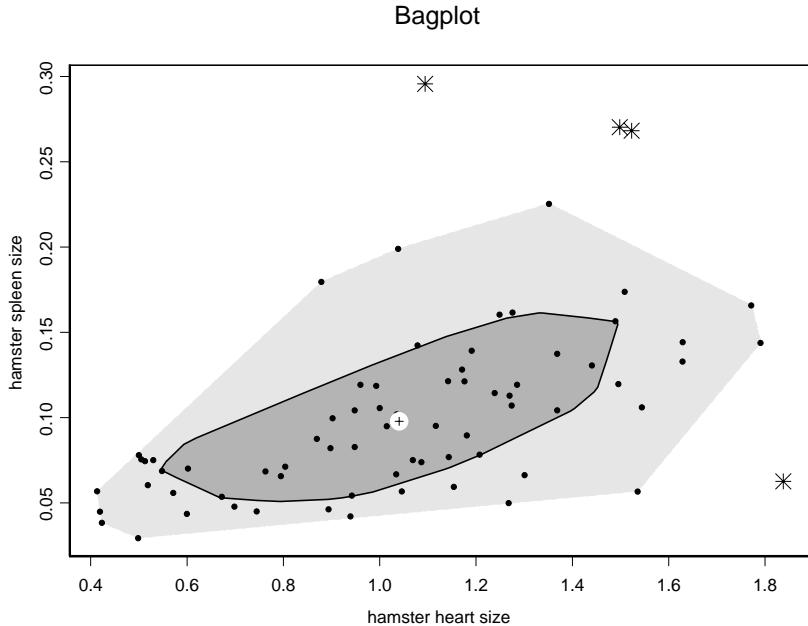
Illustration of the bivariate halfspace depth. Here θ (which is not a data point itself) has depth 1 because the halfspace determined by \mathbf{u} contains only one data point.

The halfspace depth regions form a sequence of nested polyhedra. Each D_k is the intersection of all halfspaces containing at least $n - k + 1$ data points. Moreover,

every data point must be a vertex of one or more depth regions. The median for halfspace depth is called the *Tukey median*. When the innermost depth region is larger than a singleton, the Tukey median is defined as its centroid. This makes the Tukey median unique by construction.

Note that the depth regions give an indication of the shape of the data cloud. Based on this idea one can construct the *bagplot* [RRT99], a bivariate version of the univariate boxplot. Figure 57.1.2 shows such a bagplot. The cross in the white disk is the Tukey median. The dark area is an interpolation between two subsequent depth contours, and contains 50% of the data. This area (the “bag”) gives an idea of the shape of the majority of the data cloud. Inflating the bag by a factor of 3 relative to the Tukey median yields the “fence” (not shown), and data points outside the fence are called outliers and marked by stars. Finally, the light gray area is the convex hull of the non-outlying data points.

FIGURE 57.1.2
Bagplot of the heart and spleen size of 73 hamsters.



An often used criterion to judge the robustness of an estimator is its *breakdown value*. The breakdown value is the smallest fraction of data points that we need to replace in order to move the estimator of the contaminated data set arbitrarily far away. The classical mean of a data set has breakdown value zero since it will already explode when we move one observation far out. Note that for any estimator which is equivariant for translation (which is required to call it a location estimator) the breakdown value can be at most $1/2$. (If we replace half of the points by a far-away

translation image of the remaining half, the estimator cannot distinguish which were the original data.)

The Tukey depth and the corresponding median have good statistical properties. The Tukey median T^* is a location estimator with breakdown value $\epsilon_n(T^*; X_n) \geq 1/(d+1)$ for any data set in general position. This means that it remains in a predetermined bounded region unless $n/(d+1)$ or more data points are moved. At an elliptically symmetric distribution the breakdown value becomes $1/3$ for large n , irrespective of d . Moreover, the halfspace depth is invariant under all nonsingular affine transformations of the data, making the Tukey median affine equivariant. Since data transformations such as rotation and rescaling, are very common in statistics, this is an important property. The statistical asymptotics of the Tukey median have been studied in [BH99].

CENTERPOINTS

There is a close relationship between the Tukey depth and centerpoints, which have been long studied in computational geometry. In fact, Tukey depth extends the notion of centerpoint. A *centerpoint* is any point with halfspace depth $\geq \lceil n/(d+1) \rceil$. A consequence of Helly's theorem is that there always exists at least one centerpoint, so the depth of the Tukey median cannot be less than $\lceil n/(d+1) \rceil$.

OTHER LOCATION DEPTH NOTIONS

1. **Simplicial depth** ([Liu90]). The depth of θ equals the number of simplices formed by $d+1$ data points that contain θ . Formally,

$$sdepth(\theta; X_n) = \#\{(i_1, \dots, i_{d+1}); \theta \in S[x_{i_1}, \dots, x_{i_{d+1}}]\}.$$

The simplicial median is affine equivariant with a breakdown value bounded above by $1/(d+2)$.

2. **Oja depth** ([Oja83]). This is also called simplicial volume depth:

$$odepth(\theta; X_n) = \left(1 + \sum_{(i_1, \dots, i_d)} \{volume\ S[\theta, \mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_d}]\}\right)^{-1}.$$

The corresponding median is also affine equivariant, but has zero breakdown value.

3. **Projection depth.** We first define the *outlyingness* ([DG92]) of any point θ relative to the data set X_n as

$$O(\theta; X_n) = \max_{\|\mathbf{u}\|=1} \frac{|\mathbf{u}^\tau \theta - \text{med}_i\{\mathbf{u}^\tau \mathbf{x}_i\}|}{\text{MAD}_i\{\mathbf{u}^\tau \mathbf{x}_i\}},$$

where the median absolute deviation (MAD) of a univariate data set $\{y_1, \dots, y_n\}$ is the statistic $\text{MAD}_i\{y_i\} = \text{med}_i|y_i - \text{med}_j\{y_j\}|$. The outlyingness is small for centrally located points and increases if we move toward the boundary of the data cloud. Instead of the median and the MAD, also another pair (T, S) of a location and scatter estimate may be chosen. This leads to different notions of projection depth, all defined as

$$pdepth(\theta; X_n) = (1 + O(\theta; X_n))^{-1}.$$

General projection depth is studied in [Zuo03]. As with the median and the MAD, the projection depth has breakdown value 1/2 and is affine equivariant.

4. ***Spatial median*** ([Gow74]). This median maximizes the function

$$L^1 \text{depth}(\boldsymbol{\theta}; X_n) = (1 + \sum_{i=1}^n \|\mathbf{x}_i - \boldsymbol{\theta}\|)^{-1}.$$

It has breakdown value 1/2, but is not affine equivariant (it is only equivariant with respect to translations, multiplication by a scalar factor, and orthogonal transformations).

5. ***Convex hull peeling***. Here the depth of a point $\boldsymbol{\theta}$ is defined as the level of the convex layer of X_n to which $\boldsymbol{\theta}$ belongs. The convex hull of the data set has level 1. By removing these points and repeating the procedure on the remaining points we obtain a sequence of nested convex layers which define the higher levels. The resulting depth has no population analog, unlike the other definitions given above. Moreover, its robustness properties are not good, hence it will not be considered further in this chapter.

A comparison of the main properties of the different location depth medians is given in Table 57.1.1.

TABLE 57.1.1 Comparison of several location depth medians.

MEDIAN	BREAKDOWN VALUE	AFFINE EQUIVARIANCE
Tukey	worst-case $1/(d+1)$ typically $1/3$	yes
Oja	$2/n \approx 0$	yes
Simplicial	$\leq 1/(d+2)$	yes
Projection	$1/2$	yes
Spatial	$1/2$	no

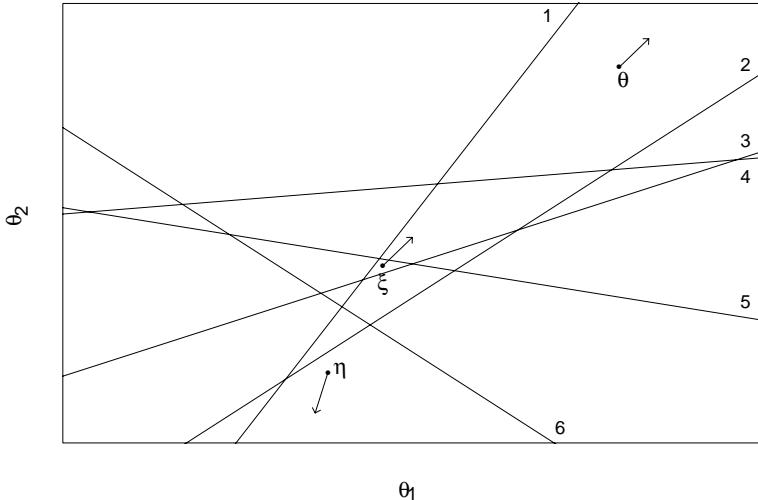
REGRESSION DEPTH

Following [RH99b] we now define the depth of a point relative to an arrangement of hyperplanes (see [Chapter 24](#)). A point $\boldsymbol{\theta}$ is said to have depth 0 if there exists a ray $\{\boldsymbol{\theta} + \lambda\mathbf{u}; \lambda \geq 0\}$ that does not cross any of the hyperplanes h_i in the arrangement. (A hyperplane parallel to the ray is counted as intersecting at infinity.) The depth of any point $\boldsymbol{\theta}$ is then the minimum number of hyperplanes intersected by any ray from $\boldsymbol{\theta}$. [Figure 57.1.3](#) shows an arrangement of lines. In this plot, the points θ and η have depth 0 and the point ξ has depth 2. The depth is always constant on open cells and on cell edges. It was shown ([RH99b]) that any arrangement of lines in the plane encloses a point with depth at least $\lceil n/3 \rceil$, giving rise to a new type of “centerpoints.”

This notion of depth was originally defined ([RH99]) in the dual, as the depth of a regression hyperplane H_θ relative to a point configuration of the form $Z_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ in \mathbb{R}^{d+1} . Regression depth ranks hyperplanes according to how well they fit the data in a regression model, with \mathbf{x} containing the predictor variables and y the response. A vertical hyperplane ($\mathbf{x} = \text{constant}$), which cannot be used to predict future response values, is called a “nonfit” and has depth 0. The regression depth of a hyperplane H_θ is found by rotating H_θ in a continuous movement until it becomes vertical. The minimum number of data points that is passed in such a rotation is called the **regression depth** of H_θ . Figure 57.1.4 is the dual representation of Figure 57.1.3. (For instance, the line θ has slope θ_1 and intercept θ_2 and corresponds to the point (θ_1, θ_2) in Figure 57.1.3.) The lines θ and η have depth equal to 0, whereas the line ξ has depth 2.

FIGURE 57.1.3

Example of the regression depth of a point in an arrangement of lines (see Figure 57.1.4 for the dual plot).

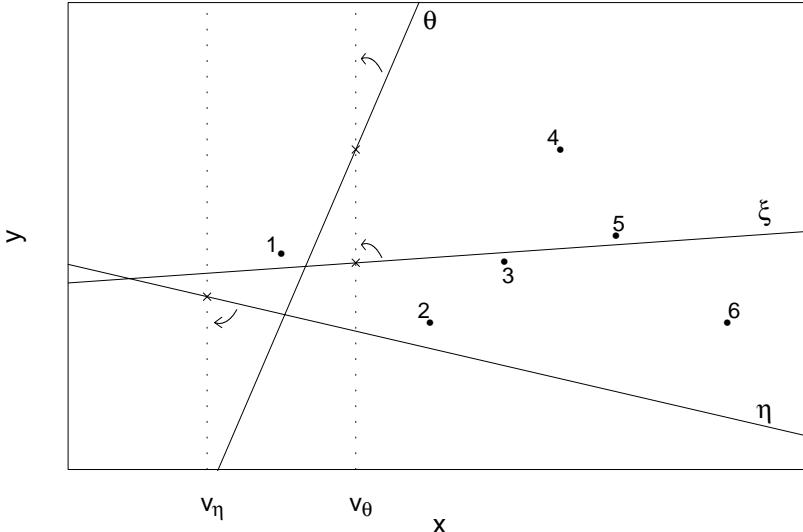


In statistics one is interested in the *deepest fit* or regression depth median, because this is a line (hyperplane) about which the data are well-balanced. The statistical properties of regression depth and the deepest fit are very similar to those of the Tukey depth and median. The bounds on the maximal depth are almost the same. Moreover, for both depth notions the value of the maximal depth can be used to characterize the symmetry of the distribution ([RS04]). The breakdown value of the deepest fit is at least $1/(d+1)$ and under linearity of the conditional median of y given \mathbf{x} it converges to $1/3$. In the next section, we will see that the optimal complexities for computing the depth and the median are also comparable. For a detailed comparison of the properties of halfspace and regression depth, see [HRV01].

The regression depth region D_k is defined in the primal, as the set of points

FIGURE 57.1.4

Example of the regression depth of a line in a bivariate configuration of points (this is the dual of Figure 57.1.3).



with arrangement depth at least k . Contrary to the Tukey depth, these depth regions need not be convex. But nevertheless it was proved that there always exists a point with arrangement depth at least $\lceil n/(d+1) \rceil$ ([ABE⁺00]).

ARRANGEMENT LEVELS

Regression depth is undirected (isotropic) in the sense that it is defined as a minimum over all possible directions. If we restrict ourselves to vertical directions \mathbf{u} (i.e., up or down), we obtain the usual levels of the arrangement (cf. Section 24.2). The absence of preferential directions makes regression depth invariant under affine transformations.

57.2 COMPUTING DEPTH

Although the definitions of depth are intuitive, the computational aspects can be quite challenging. The calculation of depth regions and medians is computationally intensive, especially for large data sets in higher dimensions. In statistical practice, such data are quite common and therefore reliable and efficient algorithms are needed. For the bivariate case several algorithms have been developed. Unfortunately, some are complex and have yet to be implemented. The computational aspects of depth in higher dimensions are still mostly unexplored.

Algorithms for depth-related measures are often more complex for data sets which are not in general position than for data sets in general position. For example, the boundaries of subsequent halfspace depth regions are always disjoint when

the data are in general position, but this does not hold for nongeneral position. Preferably, algorithms should be able to handle both the general position case and the nongeneral position case directly. As a quick fix, algorithms which were made for general position can also be applied in the other case if one first adds small random errors to the data points. For large data sets, this dithering will have a limited influence on the results.

BIVARIATE ALGORITHMS

Table 57.2.1 gives an overview of algorithms, each of which has been implemented, to compute the depth in a given point θ in \mathbb{R}^2 . These algorithms are time-optimal, since the problem of computing these bivariate depths has an $\Omega(n \log n)$ lower bound ([ACG⁺02], [LS00b]).

The algorithms for halfspace and simplicial depth are both based on the same technique. First, data points are radially sorted around θ . Then a line is rotated through θ . The depth is calculated by counting the number of points that are passed by the rotating line in a specific manner. The planar regression depth algorithm is easiest to visualize in the regression setting. To compute the depth of a hyperplane H_θ with coefficients θ , the data are first sorted along the x -axis. A vertical line L is then moved from left to right and each time a data point is passed, the number of points above and below H_θ on both sides of L is updated.

TABLE 57.2.1 Computing the depth of a bivariate point.

DEPTH	TIME COMPLEXITY	SOURCE
Tukey depth	$O(n \log n)$	[RR96]
Regression depth	$O(n \log n)$	[RH99]
Simplicial depth	$O(n \log n)$	[RR96]

In general, computing a median is harder than computing the depth in a point, because typically there are many candidate points. For instance, for the simplicial median the currently best algorithm requires $O(n^4)$ time, whereas its corresponding depth needs only $O(n \log n)$. The simplicial median seems difficult to compute because there are $O(n^4)$ candidate points (namely, all intersections of lines passing through two data points) and the simplicial depth regions have irregular shapes, but of course a faster algorithm may yet be found.

Fortunately, in several important cases the median can be computed without computing the depth of individual points. Table 57.2.2 gives an overview of algorithms to compute bivariate depth-based medians. For the regression depth median, an $\Omega(n \log n)$ lower bound was established by [LS00b], and the same lower bound holds for computing the Tukey median ([LS00]). The currently best algorithm for the Tukey median is based on Matoušek's algorithm to find the median in $O(n \log^5 n)$ time ([Mat91]). This algorithm first finds a region D_k with $k \geq \lceil n/3 \rceil$.

Then, a binary search is used to find the largest k for which $D_k \neq \emptyset$. Unfortunately this procedure seems too complex to implement. (An actual implementation is available for a slower algorithm in [RR98].)

A linear-time algorithm to compute a bivariate centerpoint is described in [JM94].

TABLE 57.2.2 Computing the bivariate median.

MEDIAN	TIME COMPLEXITY	SOURCE
Tukey median	$O(n \log^4 n)$	[LS00]
Regression depth median	$O(n \log n)$	[LS00b]
Simplicial median	$O(n^4)$	[ALS ⁺ 03]
Oja median	$O(n \log^3 n)$	[ALS ⁺ 03]

The computation of bivariate halfspace depth regions has also been studied. The first algorithm [RR96b] required $O(n^2 \log n)$ time per depth region. An algorithm to compute all regions in $O(n^2)$ time is constructed and implemented in [MRR⁺01]. This algorithm thus also yields the Tukey median in $O(n^2)$ time. It is based on the dual arrangement of lines where topological sweep is applied. A completely different approach is implemented in [KMV02]. They make direct use of the graphics hardware to approximate the depth contours of a set of points in $O(nW + W^3) + nCW^2/512$ time, where the pixel grid is of dimension $(2W + 1) \times (2W + 1)$.

ALGORITHMS IN HIGHER DIMENSIONS

Algorithms to compute the halfspace and regression depth of a given point in \mathbb{R}^d in $O(n^{d-1} \log n)$ time are constructed in [RS98], where also faster approximate algorithms are given.

The simplicial depth of a point in \mathbb{R}^3 can be computed in $O(n^2)$ time, and in \mathbb{R}^4 the fastest algorithm needs $O(n^4)$ time [CO01]. For higher dimensions, no better algorithm is known than the straightforward $O(n^{d+1})$ method to compute all simplices.

The currently best available algorithms for computing the halfspace, regression, and simplicial depth in higher dimensions all use projections onto a lower-dimensional space. This reduces the problem to computing bivariate depths, for which optimal algorithms exist.

Very little is known about the computation of high-dimensional depth medians and regions. A steepest descent algorithm to approximate the Tukey median in any dimension was developed in [SR00]. In [VRH⁺02] an algorithm is described to approximate the deepest fit in any dimension. No efficient implementable algorithm for the depth regions in 3 dimensions is yet available. The algorithm of [MRR⁺01] can theoretically be generalized to higher dimensions, but the sweeping method they use has not yet been implemented for more than two dimensions.

OPEN PROBLEMS

Aside from the fact that many of the above described algorithms are clearly not optimal, and that the optimal possible complexity for most remains unknown, three important open problems stand out:

1. The projection depth has better statistical properties than most other location depth notions. However, its practical use is severely limited by the absence of an efficient algorithm to compute the projection depth and the associated median.
2. An efficient and implementable algorithm for 3D depth contours is needed. This would allow for a natural extension of the bagplot to three dimensions.
3. Most data sets have more than two variables. Algorithms to compute the medians for location as well as regression depth in any dimension are therefore needed in practice. For large data sets, good approximate algorithms can be a valuable alternative to optimal exact algorithms, which may be quite slow.

57.3 OTHER STATISTICAL TECHNIQUES

Computational geometry has provided fast and reliable algorithms for many other statistical techniques, especially for bivariate problems.

Linear regression is a frequently used statistical technique. The ordinary least squares regression, minimizing the sum of squares of the residuals, is easy to calculate, but produces unreliable results whenever one or more outliers are present in the data. Robust alternatives are often computationally intensive. We here give some examples of regression methods for which geometric or combinatorial algorithms have been constructed.

1. **L^1 regression.** This well-known alternative to least squares regression minimizes the sum of the absolute values of the residuals, and is robust to vertical outliers. Algorithms for L^1 regression may be found in, e.g., [YKI⁺88].
2. **Least median of squares (LMS) regression** ([Rou84]). This method minimizes the median of the squared residuals and has a breakdown value of 1/2. To compute the bivariate LMS line, an $O(n^2)$ algorithm using topological sweep has been developed [ES90]. An approximation algorithm for the LMS line is constructed in [MN⁺97].
3. **Median slope regression** ([The50], [Sen68]). This bivariate regression technique selects the line with median slope of all lines through two data points. An optimal $O(n \log n)$ algorithm is given in [BC98], and a more practical randomized algorithm in [DMN92].
4. **Repeated median regression** ([Sie82]). Median slope regression takes the median over all couples (d -tuples in general) of data points. Here, this median is replaced by d nested medians. For the bivariate repeated median regression line, [MMN98] provide an efficient randomized algorithm.

Algorithms for higher-dimensional generalizations of the median slope and repeated median regression estimators are discussed in [MN94].

The aim of cluster analysis (Section 51.1) is to divide a data set into clusters of similar objects. Partitioning methods divide the data into k groups. Hierarchical methods construct a complete clustering tree, such that each cut of the tree gives a partition of the data set. A selection of clustering methods with accompanying algorithms is presented in [SHR97]. The general problem of partitioning a data set into groups such that the partition minimizes a given error function f is NP-hard. However, for some special cases efficient algorithms exist. For a small number of clusters in low dimensions, exact algorithms for partitioning methods can be constructed. Constructing clustering trees is also closely related to geometric problems (see e.g., [Epp97], [Epp98]).

57.4 SOURCES AND RELATED MATERIAL

SURVEYS

All results not given an explicit reference above may be traced in these surveys.

[LPS99]: A survey of multivariate data depth and its statistical applications.

[Sha76]: An overview of the computational complexities of basic statistics problems like ranking, regression, and classification.

[Sma90]: An overview of several multivariate medians and their basic properties.

[ZS00]: A classification of multivariate data depths based on their statistical properties.

RELATED CHAPTERS

[Chapter 24: Arrangements](#)

[Chapter 51: Pattern recognition](#)

REFERENCES

- [ACG⁺02] G. Aloupis, C. Cortés, F. Gómez, M. Soss, and G.T. Toussaint. Lower bounds for computing statistical depth. *Comput. Statist. Data Anal.*, 40:223–229, 2002.
- [ALS⁺03] G. Aloupis, S. Langerman, M. Soss, and G.T. Toussaint. Algorithms for bivariate medians and a Fermat-Torricelli problem for lines. *Comput. Geom. Theory Appl.*, 26:69–79, 2003.
- [ABE⁺00] N. Amenta, M. Bern, D. Eppstein, and S.-H. Teng. Regression depth and center points. *Discrete Comput. Geom.*, 23:305–323, 2000.
- [BH99] Z.-D. Bai and X. He. Asymptotic distributions of the maximal depth estimators for regression and multivariate location. *Ann. Statist.*, 27:1616–1637, 1999.

- [BC98] H. Brönnimann and B. Chazelle. Optimal slope selection via cuttings. *Comput. Geom.*, 10:23–29, 1998.
- [CO01] A.Y. Cheng and M. Ouyang. On algorithms for simplicial depth. In *Proc. 13th Canadian Conf. on Comp. Geom.*, pages 53–56, Waterloo, 2001.
- [DMN92] M.B. Dillencourt, D.M. Mount, and N.S. Netanyahu. A randomized algorithm for slope selection. *Internat. J. Comput. Geom. Appl.*, 2:1–27, 1992.
- [DG92] D.L. Donoho and M. Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann. Statist.*, 20:1803–1827, 1992.
- [ES90] H. Edelsbrunner and D.L. Souvaine. Computing least median of squares regression lines and guided topological sweep. *J. Amer. Statist. Assoc.*, 85:115–119, 1990.
- [Epp97] D. Eppstein. Faster construction of planar two-centers. In *Proc. 8th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 131–138, New Orleans, 1997.
- [Epp98] D. Eppstein. Fast hierarchical clustering and other applications of dynamic closest pairs. In *Proc. 9th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 619–628, San Francisco, 1998.
- [Gow74] J.C. Gower. The mediancenter. *J. Roy. Statist. Soc. Ser. C*, 32:466–470, 1974.
- [HRV01] M. Hubert, P.J. Rousseeuw, and S. Van Aelst. Similarities between location depth and regression depth. In L.T. Fernholz, editor, *Statistics in Genetics and in the Environmental Sciences*, pages 153–162. Birkhäuser Verlag, Basel, 2001.
- [JMH94] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. *Discrete Comput. Geom.*, 12:291–312, 1994.
- [KMF02] S. Krishnan, N.H. Mustafa, and S. Venkatasubramanian. Hardware-assisted computation of depth contours. In *Proc. 13th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 558–567, 2002.
- [LS00] S. Langerman and W. Steiger. Computing a maximal depth point in the plane. In *Proc. Japan Conf. on Discrete and Computational Geometry*, page 46, 2000.
- [LS00b] S. Langerman and W. Steiger. An optimal algorithm for hyperplane depth in the plane. In *Proc. 11th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 54–59, San Francisco, 2000.
- [Liu90] R.Y. Liu. On a notion of data depth based on random simplices. *Ann. Statist.*, 18:405–414, 1990.
- [LPS99] R.Y. Liu, J. Parelius, and K. Singh. Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Ann. Statist.*, 27:783–840, 1999.
- [Mat91] J. Matoušek. Computing the center of planar point sets. In J.E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete Computational Geometry: Papers from the DIMACS Special Year*, volume 6 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 221–230, 1991.
- [MMN98] J. Matoušek, D.M. Mount, and N.S. Netanyahu. Efficient randomized algorithms for the repeated median line estimator. *Algorithmica*, 20:136–150, 1998.
- [MRR⁺01] K. Miller, S. Ramaswami, P.J. Rousseeuw, T. Sellares, D.L. Souvaine, I. Streinu, and A. Struyf. Fast implementation of depth contours using topological sweep. In *Proc. 12th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 690–699, Washington, 2001.
- [MN94] D.M. Mount and N.S. Netanyahu. Computationally efficient algorithms for high-dimensional robust estimators. *Graphical Models Image Proc.*, 56:289–303, 1994.

- [MN⁺97] D.M. Mount, N.S. Netanyahu, K. Romanik, R. Silverman, and A.Y. Wu. A practical approximation algorithm for the LMS line estimator. In *Proc. 8th Annu. ACM-SIAM Sympoz. Discrete Algorithms*, 473–482, New Orleans, 1997.
- [Oja83] H. Oja. Descriptive statistics for multivariate distributions. *Statist. Probab. Lett.*, 1:327–332, 1983.
- [Rou84] P.J. Rousseeuw. Least median of squares regression. *J. Amer. Statist. Assoc.*, 79:871–880, 1984.
- [RH99] P.J. Rousseeuw and M. Hubert. Regression depth. *J. Amer. Statist. Assoc.*, 94:388–402, 1999.
- [RH99b] P.J. Rousseeuw and M. Hubert. Depth in an arrangement of hyperplanes. *Discrete Comput. Geom.*, 22:167–176, 1999.
- [RR96] P.J. Rousseeuw and I. Ruts. Algorithm AS 307: Bivariate location depth. *J. Roy. Statist. Soc. Ser. C*, 45:516–526, 1996.
- [RR98] P.J. Rousseeuw and I. Ruts. Constructing the bivariate Tukey median. *Statistica Sinica*, 8:827–839, 1998.
- [RRT99] P.J. Rousseeuw, I. Ruts, and J.W. Tukey. The bagplot: A bivariate boxplot. *Amer. Statist.*, 53:382–387, 1999.
- [RS98] P.J. Rousseeuw and A. Struyf. Computing location depth and regression depth in higher dimensions. *Statist. Comput.*, 8:193–203, 1998.
- [RS04] P.J. Rousseeuw and A. Struyf. Characterizing angular symmetry and regression symmetry. *J. Statist. Plann. Inference*, to appear.
- [RR96b] I. Ruts and P.J. Rousseeuw. Computing depth contours of bivariate point clouds. *Comput. Statist. Data Anal.*, 23:153–168, 1996.
- [Sen68] P.K. Sen. Estimates of the regression coefficient based on Kendall’s tau. *J. Amer. Statist. Assoc.*, 63:1379–1389, 1968.
- [Sha76] M.I. Shamos. Geometry and statistics: problems at the interface. In J.F. Traub, editor, *Algorithms and Complexity: New Directions and Recent Results*, pages 251–280. Academic Press, Boston, 1976.
- [Sie82] A. Siegel. Robust regression using repeated medians. *Biometrika*, 69:242–244, 1982.
- [Sma90] C.G. Small. A survey of multidimensional medians. *Internat. Statistical Review*, 58:263–277, 1990.
- [SHR97] A. Struyf, M. Hubert, and P.J. Rousseeuw. Integrating robust clustering techniques in S-PLUS. *Comput. Statist. Data Anal.*, 26:17–37, 1997.
- [SR00] A. Struyf and P.J. Rousseeuw. High-dimensional computation of the deepest location. *Comput. Statist. Data Anal.*, 34:415–426, 2000.
- [The50] H. Theil. A rank-invariant method of linear and polynomial regression analysis (parts 1-3). *Nederl. Akad. Wetensch. Ser. A*, 53:386–392, 521–525, 1397–1412, 1950.
- [Tuk75] J.W. Tukey. Mathematics and the picturing of data. In *Proc. Internat. Congr. of Math.*, 2, pages 523–531, Vancouver, 1975.
- [VRH⁺02] S. Van Aelst, P.J. Rousseeuw, M. Hubert, and A. Struyf. The deepest regression method. *J. Multivariate Anal.*, 81:138–166, 2002.
- [YKI⁺88] P. Yamamoto, K. Kato, K. Imai, and H. Imai. Algorithms for vertical and orthogonal L1 linear approximation of points. In *Proc. 4th Sympos. Comput. Geom.*, pages 352–361, 1988.

- [Zuo03] Y. Zuo. Projection based depth functions and associated medians. *Ann. Statist.*, 31:1460–1490, 2003.
- [ZS00] Y. Zuo and R. Serfling. General notions of statistical depth function. *Ann. Statist.*, 28:461–482, 2000.

58 GEOGRAPHIC INFORMATION SYSTEMS

Marc van Kreveld

INTRODUCTION

Geographic information systems (GIS) facilitate the input, storage, manipulation, analysis, and visualization of geographic data. Geographic data generally has a location, size, shape, and various attributes, and may have a temporal component as well. Geographical analysis is important for a GIS. It includes combining different spatial themes, relating the dependency of phenomena to distance, interpolating, studying trends and patterns, and more. Without analysis, a GIS could be called a spatial database.

Not all aspects of GIS are relevant for computational geometers. Human-computer interaction, and legal aspects of GIS, are also considered part of GIS research. This chapter focuses primarily on those aspects that are susceptible to algorithms research. Even here, the approach taken within GIS research is different from the approach a computational geometer would take, with much less initial abstraction of the problem, and less emphasis on theoretical efficiency. The GIS research field is multi-disciplinary: it includes researchers from geography, geodesy, cartography, and computer science. The research areas geodesy, surveying, photogrammetry, and remote sensing primarily deal with the data input, storage, and correction aspects of GIS. Cartography mainly concentrates on the visualization aspects.

Section 58.1 deals with spatial data structures important to GIS. Section 58.2 discusses the most common spatial analysis methods. Section 58.3 discusses the visualization of spatial data, also called automated cartography. Section 58.4 deals with Digital Elevation Models (DEMs) and their algorithms. Section 58.5 discusses the most important contributions that can be made from the computational geometry perspective to research problems in GIS. Section 58.6 lists several open problems.

58.1 SPATIAL DATA STRUCTURES

GIS store different types of data separately, such as land cover, elevation, and municipality boundaries. Therefore, each such data set is stored in a separate data structure that is tailored to the data, both in terms of representation and searching efficiency.

Geometric data structures for intersection, point location, and windowing are a mainstream topic in computational geometry, and are treated at length in Chapters 34, 36, and 38. This section concentrates on concepts and results that are specific and important to GIS. We overview raster and vector representation of data, problems that appear in data input and correction in a GIS, and a well-known spatial indexing structure.

GLOSSARY

Thematic map layer: Separately stored and manipulatable component of a map that contains the data of only one specific theme or geographic variable.

(Geographic) feature: Any geographically meaningful object.

Raster structure: Representation of geometric data based on a subdivision of the underlying space into a regular grid of square pixels.

Vector structure: Representation of geometric data based on the representation of points with coordinates, and line segments between those points.

Digitizing: Process of transforming cartographic data such as paper maps into digital form by tracing boundaries with a mouse-like device.

Conflation: Process of rectifying a digital data set by comparison with another digital data set that covers the same region (cf. rubber sheeting).

Topological vector structure: Vector structure in which incidence and adjacency of points and line segments is explicitly represented.

Quadtree: Tree where every internal node has four children, and which corresponds to a recursive subdivision of a square into four subsquares. The standard quadtree is for raster data. Leaves correspond to pixels or larger squares that appear in the recursive subdivision and are uniform in the thematic value.

R-tree: Tree based on a recursive partitioning of a set of objects into subsets, where every node stores the axis-parallel bounding box of all objects that appear in its subtree. All leaves have the same depth. R-trees usually have high (but constant) degree and are well-suited for secondary memory.

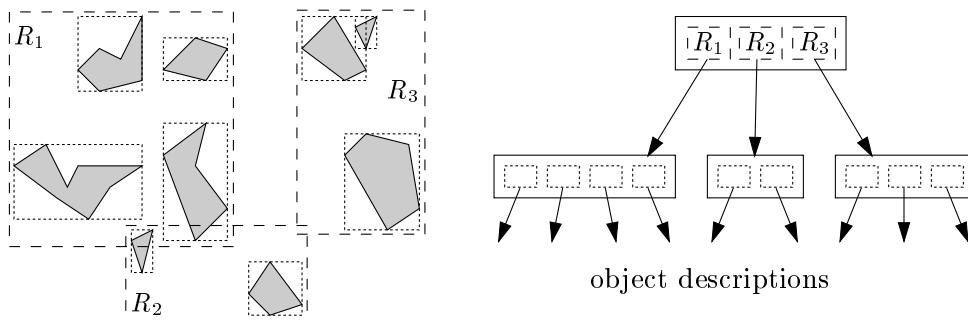
58.1.1 RASTER AND VECTOR STRUCTURES

Geographic data is composed of geometry, topology, and attributes. The attributes contain the semantics of a geographic feature. There are two essentially different ways to represent the geometric part of geographic data: raster and vector. This distinction is the same as representation in image space and object space in computer graphics ([Chapter 49](#)).

Data acquisition and input into a GIS often cause error and imprecision in the data, which must be corrected either manually or in an automated way. Also, the digitizing of paper maps yields unstructured collections of polygonal lines in vector format, to which topological structure is usually added using the GIS.

The topological vector structure obtained could be represented as, for example, a doubly-connected edge list or a quad-edge structure. But for maps with administrative boundaries, where long polygonal boundary lines occur where all vertices have degree 2, such a representation is space-inefficient. It is undesirable to have a separate object for every vertex and edge, with pointers to the incident features. The following variation gives better efficiency. Group maximal chains of degree-2 vertices into single objects, and treat them like an edge in the doubly-connected edge list. More explicitly, a chain stores pointers to the origin vertex (junction or endpoint), the destination vertex (junction or endpoint), the left face, the right face, and the next and previous chains in the two cycles of the faces incident to this

FIGURE 58.1.1
A set of polygons and an example of an R-tree for it.



chain. Such a representation allows retrieval of k adjacent faces of a face with m vertices to be reported in $O(k)$ time rather than in $O(m)$ time.

Relatively recent trends in geographic data modeling and representation include multi-scale models, temporal and spatio-temporal models, fuzzy models, and qualitative representations of location.

58.1.2 R-TREES

The most widely used spatial data structure in GIS is the R-tree of Guttmann [Gut84]. It is a type of box-tree (see [BCG⁺96]) that has high (but constant) degree internal nodes, with all leaves at the same depth. It permits any type of object to be stored, and supports several types of queries, such as windowing, point location, and intersection. Insertions and deletions are both supported. The definition of R-trees does not specify which objects go in which subtree, and different heuristics for grouping give rise to different versions [BKSS90, KF94, LLE97].

R-trees generally do not have nontrivial worst-case query time bounds, so different versions must be compared experimentally. Only the version of Agarwal et al. [ADB⁺02] has a query time bound better than linear (close to $O(\sqrt{n})$) when the stored objects are rectangles, but this structure cannot be maintained dynamically while retaining the query time.

58.2 SPATIAL ANALYSIS

Spatial analysis is the process of discovering information implicitly present in one or more spatial data sets. This includes common GIS operations such as map overlay, buffer computation, and shortest paths on road networks, but also geostatistical and spatial data analysis functions such as cluster detection, spatial interpolation, and spatial modeling. We discuss the most common forms and results in this section. For cluster analysis and classification, see [Chapter 51](#).

GLOSSARY

Map overlay: The operation of combining two thematic map layers of the same region in order to obtain one new map layer, often with a refinement of the subdivisions used for the input map layers.

Buffer: The region of the plane within a certain specified distance to a geographic feature.

Neighborhood analysis: The study of how relations between geographic features depend on the distance.

Network analysis: The study of distance, reachability, travel time, and similar geographic functions that can be defined for network data (graphs with a geographic meaning).

Cluster analysis: The study of the grouping in sets of geographic features by proximity.

Trend analysis: The study of time-dependent patterns in geographic data.

Spatial interpolation: The derivation of values at locations based on values at other (nearby) locations.

Geostatistics: Statistics for data associated with locations in the plane.

58.2.1 MAP OVERLAY

With map overlay, two or more thematic map layers are combined into one. For example, if one map layer contains elevation contours and another map layer (of the same region) forest types, then their overlay reveals which types of forest occur at which elevations. One layer can also serve as a mask for the other layer. Overlay is essential to locating a region that has various properties that appear in different thematic map layers. In the spatial database literature, map overlay is also called *spatial join*.

Map overlay is commonly solved using a plane sweep like the Bentley-Ottmann algorithm [BO79] for line segment intersection. This leads to an $O((n + k) \log n)$ time algorithm for two planar subdivisions of complexity $O(n)$, and output complexity $O(k)$. However, map overlay of two thematic map layers is essentially an extension of a red-blue line segment intersection problem, and can therefore be solved in optimal $O(n \log n + k)$ time [CEGS94, Cha94, PS94]. In case each subdivision is simply connected, the given result can be improved to $O(n + k)$ [FH95].

Map overlay in GIS must handle imprecise data as well, and therefore overlay methods that include sliver removal have been suggested. Essentially, boundaries that are closer than some pre-specified value are identified in the overlay. This is also called *epsilon filter* or *fuzzy tolerance* [Chr97]. The idea of using an epsilon band around a cartographic line is due to Perkal [Per66].

Since R-trees can also be used as access structures for subdivisions, map overlay can also be performed efficiently using R-trees [BKS93, KBS91, vO94].

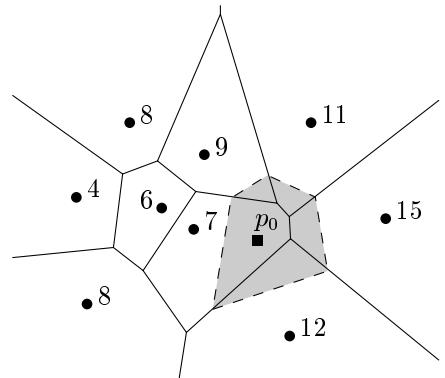


FIGURE 58.2.1

Natural neighbor interpolation at p_0 ; five measurements determine the interpolated value with weights proportional to the areas of the five grey regions.

58.2.2 BUFFER COMPUTATION

The buffer of a geographic feature of width ϵ is the same as the Minkowski sum of that feature with a disk of radius ϵ , centered at the origin; see [Chapter 47](#). Computation can be done with the algorithms mentioned in that chapter.

Buffer computation and map overlay are two main ingredients for urban planning. As an example, tree requirements for a new factory may be the proximity of a river, at least some distance to houses, and not in nature areas. A map with suitable locations is obtained after computing buffers for two of the thematic map layers, and then combining these with each other and the third layer.

58.2.3 SPATIAL INTERPOLATION

Spatial interpolation is one of the main operations in geostatistics. It is the operation of defining values at locations when only values at other locations are known. For example, when ground measurements are taken at various locations, we only know values at a finite set of points, but we would like to know the values everywhere. Several methods exist for this version of the spatial interpolation problem, including triangulation, moving windows, natural neighbors, and Kriging. Triangulation is discussed in the next section. Moving windows is simply weighted averaging of known (or observed) values within a window around the point with unknown value.

Natural neighbor interpolation is a method based on Voronoi diagrams [Sib81, SBM95]. Suppose the Voronoi diagram of the points with known values is given, and we want to obtain an interpolated value at another location p_0 . We determine what Voronoi cell p_0 would “own” if it were inserted in the point set defining the Voronoi diagram. Let A be the area of the Voronoi cell of p_0 , and let A_1, \dots, A_k be the areas removed from the Voronoi cells of the points p_1, \dots, p_k , due to the insertion of p_0 . Then, by natural neighbor interpolation, the interpolated value at p_0 is $\sum_{i=1}^k (A_i/A) \cdot V(p_i)$, where $V(p_i)$ denotes the known or observed value at p_i . The bivariate function obtained is continuous everywhere, and differentiable except at the points with known values.

Kriging is an interesting method that also applies weighted linear combinations of the known (or observed) values, that is, $V(p_0) = \sum_{i=1}^n \lambda_i \cdot V(p_i)$. The λ_i are the weights, which sum to 1. Furthermore, the weights are chosen so that

the estimation variance is less than for any other linear combination of known values. One additional advantage of Kriging is that it provides an estimation error as well [BM98].

Splines, discussed extensively in [Chapter 53](#), can also be used for interpolation. A version used in GIS are the thin-plate splines. They do not necessarily pass through the known values of the points, and can therefore reduce artifacts. The spline function minimizes the sum of two components, one representing the smoothness and the other representing the proximity to the known values of the points [BM98].

58.3 VISUALIZATION OF SPATIAL DATA

Various tasks traditionally performed manually by cartographers can be automated. GIS allow users to select their own combination of themes and data sets, and their own way of visualizing this information. It is therefore necessary that certain cartographic tasks be done in an automated manner, such as non-overlapping label placement. Since it is not known beforehand which information is shown on a map, it is impossible to pre-compute a good label placement for the geographic features of each thematic map layer separately: it must be done after it is known which layers are selected for visualization.

GLOSSARY

Choropleth map: Type of map in which the regions of an administrative subdivision are shown using a particular color scheme to represent some specified geographic variable.

Isoline map: Type of map for a continuous spatial phenomenon where lines of equal value for that phenomenon are displayed.

Cartogram: Type of map in which the area of the regions of an administrative subdivision represent some specified geographic variable (also called **value-by-area map**).

Schematic map: Type of map where important locations and connections between them (direct transportation) are shown highly stylized, and where location is preserved only approximately.

Label placement problem: The problem of placing text to annotate features on a map, according to various constraints and optimization criteria.

Line simplification problem: The problem of computing a polygonal line with fewer vertices from another polygonal line, while satisfying given constraints of distance.

Cartographic generalization problem: The problem of computing a map at a coarser (smaller) scale from a data set whose detail would be appropriate for a map at a finer (larger) scale.

58.3.1 LABEL PLACEMENT

Automated label placement has been the topic of considerable research, both within cartography and within the field of algorithms. One can distinguish three types of labels: labels for point objects, labels for line objects, and labels for polygonal objects. Imhof [Imh75] provides many examples of well-placed and poorly-placed labels, demonstrating the several different requirements for practical, high-quality label placement.

The point-label placement problem is the following optimization problem. Given a set of points, each with a specified label (name or other text), place as many labels as possible adjacent to their point, but without overlap between any two labels. One can extend the problem by restricting, or not allowing, overlap with other map features, avoiding ambiguity, and so on. Another version of point labeling is to maximize label size under the condition that all points be labeled. Label placement for point objects is usually approached as follows. A label is modeled by an axis-parallel rectangle, the bounding box of the text. For each point, define a restricted set of positions considered for its label, the candidates. Typically, the four positions where a corner of the label coincides with the point are chosen. In the label number maximization problem, the problem is abstracted to maximum independent set in a graph where edges represent intersections of two candidate label positions. Tables 58.3.1 and 58.3.2 contain some results.

TABLE 58.3.1 Point-label placement: size maximization; selected results.

TYPE OF LABEL	POSITIONS	APPROX. FACTOR	TIME	SOURCE
Equal-size square	4	2	$O(n \log n)$	[FW91, WW97]
Equal-size square	2	1	$O(n \log n)$	[FW91]
Arbitrary rectangle	2	1	$O(n \log^2 n)$	[FW91]
Equal-size circle	constant	$3.6 + \epsilon$	$O(n \log n + n \log(OPT^*/\epsilon))$	[DMM02].

TABLE 58.3.2 Point-label placement: number maximization; selected results.

TYPE OF LABEL	POSITIONS	APPROX. FACTOR	TIME	SOURCE
Rectangle	constant	$O(\log n)$	$O(n \log n)$	[AvKS98]
Fixed-height rectangle	constant	2 (or PTAS)	$O(n \log n)$	[AvKS98]
Fixed-height rectangle	touching	2 (or PTAS)	$O(n \log n)$	[vKSW99]

Combinatorial optimization approaches have also been applied frequently to the point-label placement problem [CMS95, vDTdB02]. However, experiments show that simple heuristics work well in practice [CMS95, WWKS01].

We next discuss the labeling of linear features. Here we distinguish streets and rivers. The labeling of street patterns yields a combinatorial optimization problem

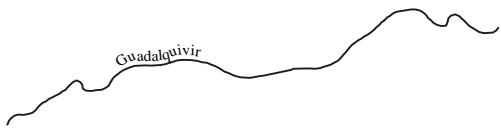


FIGURE 58.3.1

River labeling due to Wolff et al. [WK⁺00].

similar to point labeling [NW00, Str01]. River labeling is quite different, because there are several different criteria that constitute a good river label placement. The label should be close to the river, it should follow the shape of the river, it should not have too high curvature, it should be as horizontal as possible, and it should have few inflection points. The algorithm of Wolff et al. [WK⁺00] includes all of these criteria; Figure 58.3.1.

The labeling of polygonal features appears for instance when placing the name of a country or lake inside that feature. It is common to either choose horizontal and straight placement, or let the shape follow the main shape of the polygonal feature. In the first case, one can place the label in the middle of the largest scaled copy of the label that fits inside the region. In the second case, one can use the medial axis to retrieve the main shape and place the label along it.

58.3.2 CARTOGRAPHIC GENERALIZATION

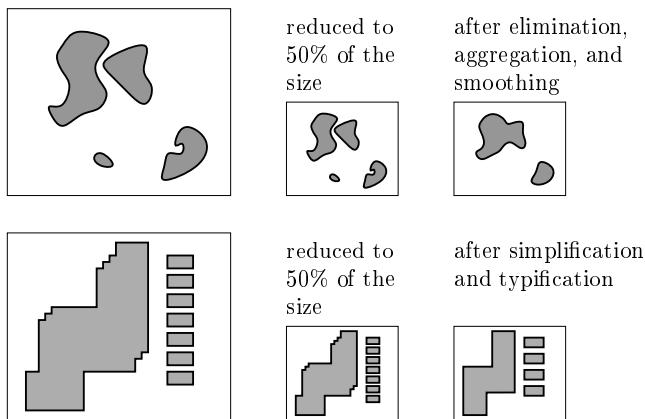
Cartographic generalization is the process of transforming and displaying cartographic data with less detail and information (i.e., on a coarser, smaller scale) than the input data contains. Examples include omitting small towns and minor roads, using only one color for nature regions rather than a distinction in forest, heath, moor, etc., aggregating several buildings into one block, and exaggerating the width of a road on a small-scale map. Generalization is a very important research topic in automated cartography [BM91, MLW95, WJ98].

The changes to the map data for generalization are done by *generalization operators*. They include selection, aggregation, typification, reclassification, smoothing, displacement, exaggeration, symbolization, collapse, and many others. See Figure 58.3.2 for some examples. Detecting a need for generalization is accomplished by computing certain geometric measures on distance, density, and detail. This will trigger the generalization operators to perform transformations. It may happen that one operator causes the need of a change somewhere else on the map, possibly leading to a domino effect. For example, one of the common generalization operators is displacement, to assure a certain distance between two map features that should not appear adjacent. Moving one feature may cause the need to move another, leading to iterative displacement algorithms [Høj98, LJ01, MP01]. The same effect can occur when smoothing a single polygonal line that represents a curved road [BM97, BB00]. Also the operators aggregation (of two or more polygons into one) and exaggeration can cause displacement of other features [BW97, Har99].

A complicating issue is the preservation of consistency in generalization. For example, generally one wants to avoid omitting a large town that is close to several cities while retaining a small town that has no cities or towns in its immediate neighborhood. This motivates the need for a global selection mechanism that models how important features are based on attributes, geometry, and neighborhood. For point features this problem is called *settlement selection* [LP86, vKvOS97].

FIGURE 58.3.2

Examples of cartographic generalization.



Several studies have also been done for selection in road networks [MM97, TR95] and in polygonal subdivisions [vO95, Jaa98].

The problem of (polygonal) line simplification (cf. Section 51.3) is often considered a cartographic generalization problem, too. However, if the motivation for line simplification is only data reduction, then line simplification cannot be considered generalization. But since line simplification methods automatically reduce detail in polygonal lines, we will discuss some methods here.

The best-known cartographic line simplification method is due to Douglas and Peucker [DP73]. Starting with a line segment between the endpoints of the polygonal line, it selects the most distant vertex to be added to the simplification, and then continues recursively on the two parts that appear. This process continues until the most distant vertex is closer than some chosen threshold value to the approximating line segment. Theoretically, the method is highly unsatisfactory because it can create self-intersections in the output, requires quadratic time in the worst case, and may need many more segments in the approximation than the optimal approximation. However, it is very simple and usually works well in practice. Hershberger and Snoeyink devised a different algorithm to compute the same approximation which runs in $O(n \log^* n)$ time [HS98].

Weibel [Wei97b] and van der Poorten and Jones [vdPJ99] demonstrate that many aspects are involved in practical line simplification for GIS, and that many different criteria may be used. The GIS literature contains several more practical approaches.

Guibas et al. [GHMS93] and Estkowsky and Michell [EM01] show that minimum vertex simplification is NP-hard when self-intersections are not allowed. Selected algorithmic results are listed in [Table 58.3.3](#).

58.3.3 SPECIAL-PURPOSE MAPS

Topographic maps are general-purpose maps that display a variety of themes of general interest together, like roads, towns, forests, and elevation contours. Special-purpose maps, on the other hand, concentrate on a particular theme, and may use

TABLE 58.3.3 Line simplification: selected results.

OUTPUT VERTICES	COMPLEXITY	ERROR CRITERION	NOTES AND PROBLEMS	SOURCE
From input	$O(n^2)$	distance	min. link, self-inter.	[CC96]
From input	$O(n^{4/3+\delta})$	vertical distance	min. link, self-inter.	[AV00]
From input	$O(n^2 \log n)$	distance	no self-inter.	[dBvKS98]
From input	$O(n^3)$	distance	no self-inter.	[EM01]
Arbitrary	$O(n^2 \log n)$	distance	min. link, self-inter.	[GHMS93]

alternative methods of visualizing the information. A choropleth map could, e.g., show the population densities of the states of the U.S. by coloring each with a color from a well-chosen set of colors, for instance, five saturation values of red. The geographic theme of population density can be seen as a function from the plane (the Earth's surface) to the nonnegative reals. Here the points of the plane are aggregated by state.

There are other ways of visualizing functions from the plane to the reals cartographically, including isoline maps, dot maps, and cartograms. The latter again applies to aggregated regions of the plane. Flow maps visualize a presence and quantity of flow from one (aggregated) region to another. Schematic maps visualize connections between locations, such as subway maps. Dent provides a useful overview of several special-purpose map types [Den99].

Cartograms show values for regions by shrinking and expanding those regions, so that the area of each region corresponds to the value represented. The most important usage is the population cartogram, where a region A with a population twice that of region B will be shown twice as large as B . Necessarily, cartograms show a distortion of the geographic space. To keep the regions more or less recognizable, they should keep their shape, location, and adjacency as much as possible.

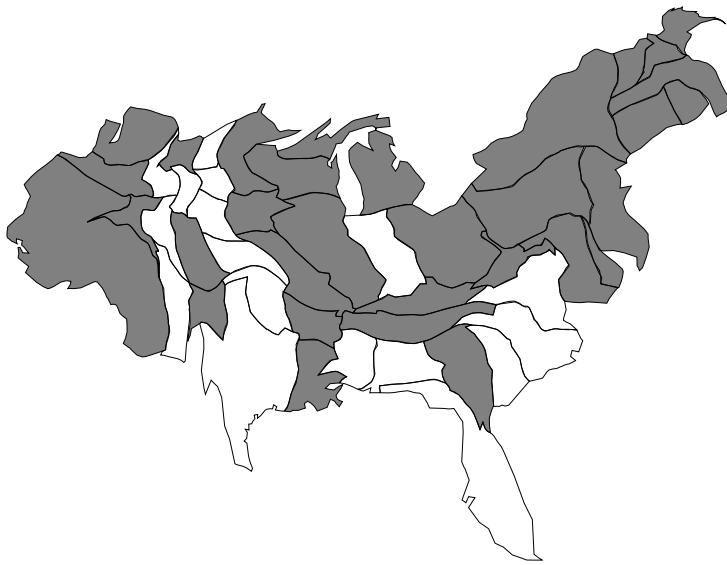
Several algorithms have been proposed to construct cartograms, given an administrative subdivision and a value for each region. Tobler [Tob86] simply uses scaling on the x - and y -coordinates, which may prevent regions from being shown at the correct size. Dougenick et al. [DCN85] compute a centroid for each region, which is assigned a repelling force if the region should grow, and an attracting force if the region should shrink. The forces of all centroids on all boundaries of the map then result in new positions of these boundaries. This is used in an iterative algorithm. Kocmoud and House's approach [KH98] is constraint programming. They also attempt to preserve the main orientations of the boundaries. Edelsbrunner and Waupotitsch [EW97] give a cartogram construction algorithm based on simplicial complexes in the plane, where paths of triangles are used to define deformations that let one region grow at the cost of the size of another. See [Figure 58.3.3](#).

Dot maps show values by dots, where one dot represents, e.g., ten thousand people. This allows the distribution of the population to be shown better than in cartograms, but the relative populations for two regions are more difficult to compare. De Berg et al. [dBB⁺02] show the connection between dot maps and discrepancy, and compare various heuristics to construct dot maps.

Schematic maps are commonly used for public transportation systems. Direct lines, or connections between major stations, are shown with a polygonal line that is highly abstracted: it has only a few segments, and these segments are horizontal,

FIGURE 58.3.3

Cartogram showing electoral votes in the 1992 presidential elections, where shaded States indicate a majority for Bill Clinton (from Edelsbrunner and Waupotitsch [EW97]).



vertical, or have slope 1 or -1 . Avelar and Müller [AM00] give an iterative algorithm that moves vertices so that the segments become more and more oriented in one of the desired orientations. Cabello et al. [CdBv⁺01] place the connections incrementally in a pre-computed order, leading to an $O(n \log n)$ time algorithm that does not require iteration. Neyer [Ney99] views the problem as a line simplification problem and approximates each connection with the minimum number of segments in the specified orientations. The three approaches to computing schematic maps are quite different and each has its own advantages and drawbacks.

Brandes and Wagner [BW98] show connections between stations by circular arcs and address the visualization problem as a graph layout problem ([Chapter 52](#)).

58.3.4 DYNAMIC AND ANIMATED MAPS

Besides computations for traditional paper maps, a more recent trend is to study dynamic maps, animated maps, interactive maps, Web maps, and multimedia maps [KB01, CPG99]. This area leads to a number of new computational issues, where efficiency is very important and quality is less critical. For example, a label placement on the screen must be updated when the user starts panning or zooming. Only as few changes as possible should be made; see, e.g., [PPH99]. Related labeling problems arise in user interface design [BFH01].

Zooming out on a map also makes real-time cartographic generalization necessary. The problem is that not only the size of features must be changed, but also the way of visualization. On large-scale maps, cities are shown by polygonal outlines, but on small-scale maps, they are shown by point symbols. The process is called ***dynamic*** or ***on-the-fly generalization*** [vO95, MG99]. Ideally, the changes made

during zooming should be made in a continuous manner, with no major, sudden changes on the map [vK01]. In static generalization, the objective is to compute a new representation. But in dynamic generalization, the problem is the computation of the transition. The relation to morphing should be clear.

58.4 DIGITAL ELEVATION MODELS

We have concentrated on types of data based on subdivisions with well-marked boundaries. Another important type of data is the scalar function in two variables. The most common example from geography is elevation above sea level, also called terrain. Three other examples are annual precipitation, nitrate concentration per cubic meter, and average noise level.

There are two common representations for elevation: the regular square grid, or *elevation matrix*, which is a raster representation, and the triangular irregular network (TIN), which is a vector representation. For the latter representation, the Delaunay triangulation is often used.

GLOSSARY

Digital elevation model (DEM): Representation of a scalar function in two variables. Sometimes specifically used for the raster-based representation.

Triangular irregular network (TIN): Vector-based representation of a digital elevation model defined by a triangulation of a point set. Also called polyhedral terrain.

Drainage network: Collection of linear features that represent the locations where water on a terrain has formed rivers.

Viewshed analysis: The study of visibility on a terrain.

58.4.1 CONSTRUCTION AND SIMPLIFICATION OF DEMS

The problem of simplifying a digital elevation model, or performing raster to vector conversion for a digital elevation model, is a higher-dimensional version of line simplification. The best algorithm known is similar in approach to the Douglas-Peucker algorithm for line simplification given in Section 58.3.2. Assume that the outer boundary of the DEM is rectangular, a set of points with their elevation is given (e.g., based on a regular square grid), and assume that a maximum allowed vertical error $\epsilon > 0$ is specified. An initial coarse simplification of the DEM is a triangulation of the four corners of the rectangle. If that simplification is no vertically further than ϵ at all points, then it is accepted. Otherwise, the point with largest vertical distance is selected and added to the triangulation, which is restored by flipping to the Delaunay triangulation. The process is then repeated.

The method requires quadratic time in the worst case, but an implementation can be given which, under natural assumptions, takes $O(n \log n)$ time in practice [Hel90, Fjä91, HG95].

Agarwal and Suri [AS98] show that a corresponding optimization problem is

NP-hard, and give an approximation algorithm that requires $O(n^8)$ time. If m is the size of the optimal piecewise linear ϵ -approximation of the n given points, then the computed approximation has size $O(m \log m)$. Agarwal and Desikan [AD97] give a cubic time ϵ -approximation algorithm with a worse size bound on the approximation, but with some assumptions the approximation has the same size asymptotically and runs in near-quadratic time.

When a TIN is constructed for modeling terrains, various geometric computation problems arise. When the input is a set of (digitized) contour lines, a triangulation between the contour lines such as the constrained Delaunay triangulation can be used [DP89, Sch98]. Care must be taken that no triangle with all three vertices on the same contour line is created, as this gives undesirable artifacts. Thibault and Gold [TG00] provide a solution that avoids flat triangles by adding vertices on the medial axis or skeleton, which are given intermediate elevations (see also [GD02]). If information on rivers is present too, then these can be included as edges of the TIN using a constrained Delaunay triangulation [MS99]. Two other approaches of interest are by Silva et al. [SMK95] and by Little and Shi [LS01]. Both methods construct a TIN from gridded data and can preserve important features like valleys and ridges of the terrain. Gudmundsson et al. [GHvK02] define a class of triangulations, called higher-order Delaunay triangulations, and use them to create TINs with fewer local minima in the terrain, because minima generally do not occur.

Multi-resolution terrain modeling has attracted considerable research, largely covered by Puppo and Scopigno [PS97]. See [Chapter 54](#) for more information on surface simplification and multi-resolution representations.

58.4.2 VISUALIZATION OF DEMS

Digital elevation models may be visualized in several ways. A traditional way is by contour maps, and the process of deriving a contour map is called *contouring* [Wat92].

A perspective view of a digital elevation model can be obtained by back-to-front rendering of the grid elements or triangles. If a vector representation of a perspective view is needed, an algorithm of Katz et al. [KOS92] achieves this for a TIN in $O((n + k) \log n \log \log n)$ time, where n is the number of triangles and k is the complexity of the visibility map.

58.4.3 DERIVED MAPS AND PRODUCTS

In the analysis of terrain—e.g., for land suitability studies—slope and aspect maps are important derived products of a digital elevation model. They are straightforward to compute. Similarly, the plan and profile curvature can be of importance, for example for waterflow and erosion modeling.

The computation of the drainage network, based on the shape of the terrain, has been frequently studied, most often for grid data. Besides the drainage network, watersheds also provide important terrain information. A surprising combinatorial result of de Berg et al. [dB⁺96] is that if water always follows the direction of steepest descent, and the drainage network consists of all points that receive flow from some region with positive area, then triangulations exist with n triangles for which the drainage network has complexity $\Theta(n^3)$.

Viewshed analysis is the study of visibility in the terrain. For gridded DEMs, the visibility index of a grid cell c is used as a measure of visibility for c . It is defined as the number of other grid cells visible from grid cell c . Viewshed analysis has applications in urban and touristical planning, and for telecommunication. The shortest watchtower problem is to compute a point with smallest vertical distance above a terrain which has visibility to all points of the terrain. It can be computed in $O(n \log n)$ time [Zhu97]. Another interesting problem is to place a small number of antennas so that any point on the terrain has direct visibility to at least one antenna [BMM⁺02, Fra02]. Other visibility results for terrains may be found in Section 28.8.3.

The computation of shortest paths between two points on a terrain is a problem of both theoretical and practical interest. The approach of Lanthier et al. [LMS01, ALMS98] is significant both theoretically and practically, also because it deals with the weighted version. The main idea is to place Steiner points on edges to convert the problem into a graph problem, and then apply Dijkstra's algorithm. This gives a simple approximation algorithm for least-cost paths.

58.5 ALGORITHMIC CHALLENGES IN GIS

The application area GIS is the source of a number of interesting research problems. Many of these are simply-stated algorithmic problems, such as finding the most efficient algorithm for a well-defined problem, or finding the best approximation factor for some computationally hard problem. But from the application perspective, more important is the study of relatively simple solutions for problems in which a number of different requirements must be satisfied or optimized simultaneously. For example, label placement with high cartographic quality has to be achieved with no overlap between different labels, no or little overlap of a label with other map features, clear association between a feature and its label, and avoidance of areas that are too dense with text. As a second example, in realistic terrain reconstruction, seven constraints have been listed [Sch98]. Such constraints cannot be formulated straightforwardly as algorithmic optimization. It is usually more important which requirements can be handled simultaneously than how efficient a solution is.

Challenges for algorithms research on GIS problems are methods that deal with multiple criteria simultaneously, either as a whole solution or as part of an optimization approach such as genetic or evolutionary algorithms. The appropriate formulation of the GIS problem itself, and comparison of results based on different formulations, are also issues of major importance.

58.6 OPEN PROBLEMS

The references in the open problems below refer to papers related to the open problem. Those papers do not always state the problem explicitly.

1. Improve on the known 3.6-approximation factor for circular label placement (3.6 times the optimal size that permits all labels to be placed) [DMM02].
2. Provide an approximation algorithm for the maximum independent set in

rectangle intersection graphs with an approximation factor better than $\Theta(\log n)$ [AvKS98].

3. Give an approximation algorithm for maximum independent set in square intersection graphs with an approximation factor less than 4.
4. Given a simple polygon P , compute the minimum perimeter shape with the same area but whose boundary remains within a given distance $\epsilon > 0$ of the original boundary of P . A solution is useful for cartographic generalization, in particular for the simplification and aggregation operators.
5. Given two simple polygons P and Q , compute two subpolygons $P' \subseteq P$ and $Q' \subseteq Q$ of maximum summed area such that, for any $p \in P'$ and $q \in Q'$, the distance between p and q is at least $\epsilon > 0$.
6. Given a simple polygonal line with n vertices, what is the complexity of computing the minimum-vertex polygonal line simplification that keeps the two endpoints, uses a subset of the intermediate vertices, has error below a given $\epsilon > 0$, and guarantees no self-intersection [AV00, dBvKS98]? (See also [Section 51.3](#).)
7. Can the output of the Douglas-Peucker line-simplification method be generated using a different algorithm that runs in linear time [HS98]?
8. Does an efficient data structure exist for natural neighbor interpolation queries in a point set S of n points with values? It is easy to develop a linear-size data structure with $O(\log n + k)$ query time, where k is the number of Voronoi neighbors of the query point amidst the points of the set S . However, k can be linear in n in the worst case.
9. Can the polyhedral terrain simplification algorithm of [Section 58.4.1](#) be implemented to run in $o(n^2)$ time? Implementations exist where $O(n \log n)$ time is typical for realistic inputs, but no implementation guarantees a running time less than quadratic [Hel90, Fjä91, HG95].
10. Develop elevation grid-to-TIN conversion algorithms that approximately preserve the slope of the terrain, rather than the elevation. Correct slope values are more important in practice than correct elevation values [GD02].
11. Let T_1 and T_2 be two polyhedral terrains covering the same region. Develop an algorithm that constructs a new polyhedral terrain T_3 which represents the multiplication of the corresponding elevation values of T_1 and T_2 within a given error ϵ . For certain variables that are scalar functions of location, models exist that express the value as the product of other variables that are scalar functions of location [Mit91].
12. How efficiently can the visibility index of all grid cells of an $n \times n$ grid of elevation values be computed?
13. Develop approximation algorithms for antenna placement on terrains, where the objective is placing as few antennas as possible for a given antenna height, while each point on the terrain has visibility to the top of some antenna. Furthermore, develop approximation algorithms when the antenna height is not fixed but should be kept small [Fra02, BMM⁺02].

14. Given a simple polygon and a positive real ϵ , what is the smallest (or largest) area of the polygon if each vertex can be moved over a distance at most ϵ ? A similar problem can be stated to give upper and lower bounds on the volume of a subsurface reservoir of oil, based on imprecise measurements of depth at various points. Several other problems arise due to measurement imprecision.

There are many open problems on improved algorithms for specific generalization operators, cartograms, flow maps, and other special-purpose maps, where “improved” refers to the visual output of the algorithm.

Following up on the previous point, it is important to study which geometric measures are most relevant to quantify visual aspects like sinuosity, density, similarity, and so on and so forth.

There are also many open problems concerning an appropriate (geometric) definition of physical geographic objects like mountains, valleys, and meanders. Such definitions lead to new algorithmic problems whose solutions will allow the automated characterization of the objects from data sets.

58.7 SOURCES AND RELATED MATERIAL

BOOKS

- [Jon97, LGMR01]: Two general GIS books.
- [BM98]: A GIS book that emphasizes spatial analysis for physical geography.
- [Wor95]: A GIS book with a spatial database focus.
- [RMM⁺95]: A book on cartography and, to lesser extent, GIS.
- [Den99]: A book on cartography that also contains several automated methods.

OTHER

Other surveys: computational geometry and GIS [dMP00], spatial data structures [NW97], algorithms for generalization [Wei97a], algorithms for DEMs [vK97], visualization of TINs [dB97].

Journals: International Journal of Geographical Information Science (IJGIS), GeoInformatica, Cartography & GIS, Cartographica.

Conference proceedings: International Symposium on Spatial Data Handling (SDH), Auto-Carto, International Cartographic Conference (ICC), GIScience, Conference on Spatial Information Theory (COSIT), Symposium on Spatial Databases (SSD).

RELATED CHAPTERS

- [Chapter 23: Voronoi diagrams and Delaunay triangulations](#)
- [Chapter 49: Computer graphics](#)
- [Chapter 51: Pattern recognition](#)
- [Chapter 54: Surface simplification and 3D geometry compression](#)

REFERENCES

- [AD97] P.K. Agarwal and P.K. Desikan. An efficient algorithm for terrain simplification. In *Proc. 8th ACM-SIAM Sympos. Discrete Algorithms*, pages 139–147, 1997.
- [AdB⁺02] P.K. Agarwal, M. de Berg, J. Gudmundsson, M. Hammar, and H.J. Haverkort. Box-trees and R-trees with near-optimal query time. *Discrete Comput. Geom.*, 28:291–312, 2002.
- [ALMS98] L. Aleksandrov, M. Lanthier, A. Maheshwari, and J.-R. Sack. An ϵ -approximation algorithm for weighted shortest paths on polyhedral surfaces. In *Proc. 6th Scand. Workshop Algorithm Theory*, volume 1432 of *Lecture Notes Comput. Sci.*, pages 11–22. Springer-Verlag, Berlin, 1998.
- [AM00] S. Avelar and M. Müller. Generating topologically correct schematic maps. In *Proc. 9th Internat. Sympos. Spatial Data Handling*, pages 4a.28–4a.35, 2000.
- [AS98] P.K. Agarwal and S. Suri. Surface approximation and geometric partitions. *SIAM J. Comput.*, 27:1016–1035, 1998.
- [AV00] P.K. Agarwal and K.R. Varadarajan. Efficient algorithms for approximating polygonal chains. *Discrete Comput. Geom.*, 23:273–291, 2000.
- [AvKS98] P.K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectangles. *Comput. Geom. Theory Appl.*, 11:209–218, 1998.
- [BB00] M. Bader and M. Barrault. Improving Snakes for linear feature displacement in cartographic generalization. In *Proc. GeoComputation*, 2000. <http://www.geocomputation.org/2000/GC034/Gc034.htm>
- [BCG⁺96] G. Barequet, B. Chazelle, L.J. Guibas, J.B.S Mitchell, and A. Tal. BOXTREE: A hierarchical representation for surfaces in 3D. *Comput. Graph. Forum*, 15:C387–C396, C484, 1996.
- [BFH01] B. Bell, S.K. Feiner, and T. Höllerer. View management for virtual and augmented reality. In *ACM Sympos. User Interface Software Tech.*, pages 101–110, 2001.
- [BKS93] T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient processing of spatial joins using R-trees. In *Proc. ACM SIGMOD*, pages 237–246, 1993.
- [BKSS90] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. ACM SIGMOD Conf. Management Data*, pages 322–331, 1990.
- [BM91] B.P. Buttenfield and R.B. McMaster, editors. *Map Generalization: Making Rules for Knowledge Representation*. Longman, London, 1991.
- [BM97] D. Burghardt and S. Meier. Cartographic displacement using the Snakes concept. In W. Foerstner and L. Pluemer, editors, *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps*. Birkhäuser-Verlag, Basel, 1997.
- [BM98] P.A. Burrough and R.A. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, New York, 1998.
- [BMM⁺02] B. Ben-Moshe, J.S.B. Mitchell, M.J. Katz, and Y. Nir. Visibility preserving terrain simplification—An experimental study. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 303–311, 2002.
- [BO79] J.L. Bentley and T.A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.

- [BW97] M. Bader and R. Weibel. Detecting and resolving size and proximity conflicts in the generalization of polygonal maps. In *Proc. 18th Internat. Cartographic Conf.*, pages 1525–1532, 1997.
- [BW98] U. Brandes and D. Wagner. Using graph layout to visualize train interconnection data. In *Proc. Graph Drawing*, volume 1547, *Lecture Notes Comput. Sci.*, pages 44–56. Springer-Verlag, Berlin, 1998.
- [CC96] W.S. Chan and F. Chin. Approximation of polygonal curves with minimum number of line segments or minimum error. *Internat. J. Comput. Geom. Appl.*, 6:59–77, 1996.
- [CdBv⁺01] S. Cabello, M. de Berg, S. van Dijk, M. van Kreveld, and T. Strijk. Schematization of road networks. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 33–39, 2001.
- [CEGS94] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Algorithms for bichromatic line segment problems and polyhedral terrains. *Algorithmica*, 11:116–132, 1994.
- [Cha94] T.M. Chan. A simple trapezoid sweep algorithm for reporting red/blue segment intersections. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 263–268, 1994.
- [Chr97] N.R. Chrisman. *Exploring Geographic Information Systems*. John Wiley, New York, 1997.
- [CMS95] J. Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Trans. Graphics*, 14:203–232, 1995.
- [CPG99] W. Cartwright, M. Peterson, and G. Gartner, editors. *Multimedia Cartography*. Springer-Verlag, Berlin, 1999.
- [dB97] M. de Berg. Visualization of TINs. In M. van Kreveld, J. Nievergelt, T. Roos, and P. Widmayer, editors, *Algorithmic Foundations of Geographic Information Systems*, volume 1340 of *Lecture Notes Comput. Sci.*, pages 79–97. Springer-Verlag, Berlin, 1997.
- [dB^B+02] M. de Berg, J. Bose, O. Cheong, and P. Morin. On simplifying dot maps. In *Proc. 18th Europ. Workshop Comput. Geom.*, pages 96–100, 2002.
- [dB^B+96] M. de Berg, P. Bose, K. Dobrindt, M. van Kreveld, M.H. Overmars, M. de Groot, T. Roos, J. Snoeyink, and S. Yu. The complexity of rivers in triangulated terrains. In *Proc. 8th Canad. Conf. Comput. Geom.*, pages 325–330, 1996.
- [dBvKS98] M. de Berg, M. van Kreveld, and S. Schirra. Topologically correct subdivision simplification using the bandwidth criterion. *Cartography and GIS*, 25:243–257, 1998.
- [DCN85] J.A. Dougenik, N.R. Chrisman, and D.R. Niemeyer. An algorithm to construct continuous area cartograms. *The Professional Geographer*, 37:75–81, 1985.
- [Den99] B.D. Dent. *Cartography*. WCB/McGraw-Hill, Boston, 5th edition, 1999.
- [DMM02] S. Doddi, M.V. Marathe, and B.M.E. Moret. Point set labeling with specified positions. *Internat. J. Comput. Geom. Appl.*, 12:29–66, 2002.
- [dMP00] L. de Floriani, P. Magillo, and E. Puppo. Applications of computational geometry to geographic information systems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 333–388. Elsevier North-Holland, Amsterdam, 2000.
- [DP73] D.H. Douglas and T.K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Canadian Cartographer*, 10:112–122, 1973.
- [DP89] L. de Floriani and E. Puppo. A survey of constrained Delaunay triangulation algorithms for surface representation. In G.G. Pineri, editor, *Issues on Machine Vision*, pages 95–104. Springer-Verlag, New York, 1989.

- [EM01] R. Estkowsky and J.S.B. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 40–49, 2001.
- [EW97] H. Edelsbrunner and E. Waupotitsch. A combinatorial approach to cartograms. *Comput. Geom. Theory Appl.*, 7:343–360, 1997.
- [FH95] U. Finke and K. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 119–126, 1995.
- [Fjä91] P.-O. Fjällström. Polyhedral approximation of bivariate functions. In *Proc. 3rd Canad. Conf. Comput. Geom.*, pages 187–190, 1991.
- [Fra02] W.R. Franklin. Siting observers on terrain. In *Proc. 10th Internat. Sympos. Advances in Spatial Data Handling*, pages 109–120. Springer-Verlag, Berlin, 2002.
- [FW91] M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 281–288, 1991.
- [GD02] C.M. Gold and M. Dakowicz. Terrain modelling based on contours and slopes. In *Advances in Spatial Data Handling (Proc. 10th Internat. Sympos.)*, pages 95–107. Springer-Verlag, Berlin, 2002.
- [GHMS93] L.J. Guibas, J. Hershberger, J.S.B. Mitchell, and J. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *Internat. J. Comput. Geom. Appl.*, 3:383–415, 1993.
- [GHvK02] J. Gudmundsson, M. Hammar, and M. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom. Theory Appl.*, 23:85–98, 2002.
- [Gut84] A. Guttmann. R-Trees: a dynamic indexing structure for spatial searching. In *Proc. ACM-SIGMOD Internat. Conf. Management Data*, pages 47–57, 1984.
- [Har99] L. Harrie. The constraint method for solving spatial conflicts in cartographic generalisation. *Cartography and GIS*, 26:55–69, 1999.
- [Hel90] M. Heller. Triangulation algorithms for adaptive terrain modeling. In *Proc. 4th Internat. Sympos. Spatial Data Handling*, pages 163–174, 1990.
- [HG95] P.S. Heckbert and M. Garland. Fast polygonal approximation of terrains and height fields. Report CMU-CS-95-181, Carnegie Mellon Univ., 1995.
- [Høj98] P. Højholt. Solving local and global space conflicts in map generalization using a finite element method adapted from structural mechanics. In *Proc. 8th Internat. Sympos. Spatial Data Handling*, pages 679–689, 1998.
- [HS98] J. Hershberger and J. Snoeyink. Cartographic line simplification and polygon CSG formulae in $O(n \log^* n)$ time. *Comput. Geom. Theory Appl.*, 11:175–185, 1998.
- [Imh75] E. Imhof. Positioning names on maps. *The American Cartographer*, 2:128–144, 1975.
- [Jaa98] O. Jaakkola. Multi-scale categorical data bases with automatic generalization transformations based on map algebra. *Cartography and GIS*, 25:195–207, 1998.
- [Jon97] C. Jones. *Geographical Information Systems and Computer Cartography*. Addison-Wesley Longman, Harlow, 1997.
- [KB01] M.-J. Kraak and A. Brown, editors. *Web Cartography: Developments and Prospects*. Taylor & Francis, London, 2001.
- [KBS91] H.-P. Kriegel, T. Brinkhoff, and R. Schneider. The combination of spatial access methods and computational geometry in geographic database systems. In *Proc. Advances Spatial Databases, Lecture Notes Comput. Sci.*, volume 525, pages 5–21. Springer-Verlag, Berlin, 1991.
- [KF94] I. Kamel and C. Faloutsos. Hilbert R-tree: An improved R-tree using fractals. In *Proc. 20th VLDB Conf.*, pages 500–510, 1994.

- [KH98] C.J. Kocmoud and D.H. House. A constrained-based approach to constructing continuous cartograms. In *Proc. 8th Internat. Sympos. Spatial Data Handling*, pages 236–246, 1998.
- [KOS92] M.J. Katz, M.H. Overmars, and M. Sharir. Efficient hidden surface removal for objects with small union size. *Comput. Geom. Theory Appl.*, 2:223–234, 1992.
- [LGMR01] P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind. *Geographic Information Systems and Science*. Wiley, Chichester, 2001.
- [LJ01] M. Lonergan and C.B. Jones. An iterative displacement method for conflict resolution in map generalization. *Algorithmica*, 21:287–301, 2001.
- [LLE97] S.T. Leutenegger, M.A. Lopez, and J. Edington. STR: A simple and efficient algorithm for R-tree packing. In *Proc. 13th IEEE Internat. Conf. Data Eng.*, pages 497–506, 1997.
- [LMS01] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating weighted shortest paths on polyhedral surfaces. *Algorithmica*, 30:527–562, 2001.
- [LP86] G.E. Langran and T.K. Poiker. Integration of name selection and name placement. In *Proc. Auto-Carto 8*, pages 50–64, 1986.
- [LS01] J.J. Little and P. Shi. Structural lines, TINs and DEMs. *Algorithmica*, 21:243–263, 2001.
- [MG99] W.A. Mackaness and E. Glover. The application of dynamic generalization to virtual map design. In *Proc. 19th Internat. Cartographic Conf.*, 1999. CD-ROM.
- [Mit91] C. Mitchell. *Terrain Evaluation*. Longman, Harlow, 2nd edition, 1991.
- [MLW95] J.-C. Müller, J.-P. Lagrange, and R. Weibel, editors. *GIS and Generalization – Methodology and Practice*, volume 1 of *GISDATA*. Taylor & Francis, London, 1995.
- [MM97] G.A. Mackechnie and W.A. Mackaness. Detection and simplification of road junctions in automated map generalization. In *ACSM/ASPRS Annu. Conv. & Expos., Tech. Papers Volume 1: Surveying & Cartography*, pages 72–82, Bethesda, 1997. ASPRS/ACSM.
- [MP01] W.A. Mackaness and R. Purves. Automated displacement for large numbers of discrete map objects. *Algorithmica*, 21:302–311, 2001.
- [MS99] M. McAllister and J. Snoeyink. Extracting consistent watersheds from digital river and elevation data. In *Proc. ASPRS/ACSM Annu. Conf. (Electronic)*, 1999.
- [Ney99] G. Neyer. Line simplification with restricted orientations. In *Proc. Workshop Algorithms Data Struct., Lecture Notes Comput. Sci.*, volume 1663, pages 13–24. Springer-Verlag, Berlin, 1999.
- [NW97] J. Nievergelt and P. Widmayer. Spatial data structures: concepts and design choices. In M. van Kreveld, J. Nievergelt, T. Roos, and P. Widmayer, editors, *Algorithmic Foundations of Geographic Information Systems*, volume 1340 of *Lecture Notes Comput. Sci.*, pages 153–197. Springer-Verlag, Berlin, 1997.
- [NW00] G. Neyer and F. Wagner. Labeling downtown. In *Proc. Italian Conf. Algorithms Complexity*, volume 1767 of *Lecture Notes Comput. Sci.*, pages 113–125. Springer-Verlag, Berlin, 2000.
- [Per66] J. Perkal. On the length of empirical curves. Discussion paper 10, Ann Arbor Michigan Inter-University Community of Mathematical Geographers, 1966.
- [PPH99] I. Petzold, L. Plümer, and M. Heber. Label placement for dynamically generated screen maps. In *Proc. 19th Internat. Cartographic Conf.*, pages 893–903, 1999.

- [PS94] L. Palazzi and J. Snoeyink. Counting and reporting red/blue segment intersections. *CVGIP: Graph. Models Image Process.*, 56:304–311, 1994.
- [PS97] E. Puppo and R. Scopigno. Simplification, LOD and multiresolution principles and applications. In *Eurographics 97*, volume 16, Tutorial Notes. Blackwell, Oxford, 1997.
- [RMM⁺95] A.H. Robinson, J. Morrison, P.C. Muehrcke, A.J. Kimerling, and S.C. Guptill. *Elements of Cartography*. John Wiley & Sons, New York, 6th edition, 1995.
- [SBM95] M. Sambridge, J. Braun, and H. McQueen. Geophysical parameterization and interpolation of irregular data using natural neighbours. *Geophys. J. Internat.*, 122:837–857, 1995.
- [Sch98] B. Schneider. Geomorphologically sound reconstruction of digital terrain surfaces from contours. In *Proc. 8th Internat. SympoS. Spatial Data Handling*, pages 657–667, 1998.
- [Sib81] R. Sibson. A brief description of natural neighbour interpolation. In Vic Barnett, editor, *Interpreting Multivariate Data*, pages 21–36. John Wiley & Sons, Chichester, 1981.
- [SMK95] C. Silva, J.S.B. Mitchell, and A.E. Kaufman. Automatic generation of triangular irregular networks using greedy cuts. In *Visualization 95*, pages 201–208, IEEE Computer Society Press, San Jose, 1995.
- [Str01] T. Strijk. *Geometric Algorithms for Cartographic Label Placement*. Ph.D. thesis, Utrecht Univ., Dept. of Comput. Sci., 2001.
- [TG00] D. Thibault and C.M. Gold. Terrain reconstruction from contours by skeleton construction. *GeoInformatica*, 4:349–373, 2000.
- [Tob86] W.R. Tobler. Pseudo-cartograms. *The American Cartographer*, 13:43–50, 1986.
- [TR95] R.C. Thomson and D.E. Richardson. A graph theory approach to road network generalization. In *Proc. 17th Internat. Cartographic Conf.*, pages 1871–1880, 1995.
- [vdPJ99] Customisable line generalisation using Delaunay triangulation. In *Proc. 19th Internat. Cartographic Conf.*, 1999. CD-ROM.
- [vDTdB02] S. van Dijk, D. Thierens, and M. de Berg. On the design of genetic algorithms for geographical applications. *GeoInformatica*, 6:381–413, 2002.
- [vK97] M. van Kreveld. Digital elevation models and TIN algorithms. In M. van Kreveld, J. Nievergelt, T. Roos, and P. Widmayer, editors, *Algorithmic Foundations of Geographic Information Systems*, volume 1340, *Lecture Notes Comput. Sci.*, pages 37–78. Springer-Verlag, Berlin, 1997.
- [vK01] M. van Kreveld. Smooth generalization for continuous zooming. In *Proc. 20th Internat. Cartographic Conf.*, volume 3, pages 2180–2185, 2001.
- [vKSW99] M. van Kreveld, T. Strijk, and A. Wolff. Point labeling with sliding labels. *Comput. Geom. Theory Appl.*, 13:21–47, 1999.
- [vKvOS97] M. van Kreveld, R. van Oostrum, and J. Snoeyink. Efficient settlement selection for interactive display. In *Proc. Auto-Carto 13: ACSM/ASPRS Annu. Conven. Tech. Papers*, pages 287–296, 1997.
- [vO94] P. van Oosterom. An R-tree based map-overlay algorithm. In *Proc. EGIS 94*, pages 318–327, 1994.
- [vO95] P. van Oosterom. The GAP-tree, an approach to ‘on-the-fly’ map generalization of an area partitioning. In J.-C. Müller, J.-P. Lagrange, and R. Weibel, editors, *GIS and Generalization – Methodology and Practice*, number 1 in GISDATA. Taylor & Francis, London, 1995.

- [Wat92] D.F. Watson. *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon, Oxford, 1992.
- [Wei97a] R. Weibel. Generalization of spatial data: principles and selected algorithms. In *Algorithmic Foundations of Geographic Information Systems, Lecture Notes Comput. Sci.*, volume 1340, pages 99–152. Springer-Verlag, Berlin, 1997.
- [Wei97b] R. Weibel. A typology of constraints to line simplification. In M.J. Kraak and M. Molenaar, editors, *Proc. 7th Internat. Sympos. Spatial Data Handling*, pages 533–546, Taylor & Francis, London, 1997.
- [WJ98] R. Weibel and C.B. Jones, editors. *Special Issue on Automated Map Generalization, GeoInformatica*, volume 2. 1998.
- [WK⁺00] A. Wolff, L. Knipping, M. van Kreveld, T. Strijk, and P.K. Agarwal. A simple and efficient algorithm for high-quality line labeling. In P.M. Atkinson and D.J. Martin, editors, *Innovations in GIS VII: GeoComputation*, chapter 11, pages 147–159. Taylor & Francis, London, 2000.
- [Wor95] M.F. Worboys. *GIS: A Computing Perspective*. Taylor & Francis, London, 1995.
- [WW97] F. Wagner and A. Wolff. A practical map labeling algorithm. *Comput. Geom. Theory Appl.*, 7:387–404, 1997.
- [WWKS01] F. Wagner, A. Wolff, V. Kapoor, and T. Strijk. Three rules suffice for good label placement. *Algorithmica*, 30:334–349, 2001.
- [Zhu97] B. Zhu. Computing the shortest watchtower of a polyhedral terrain in $O(n \log n)$ time. *Comput. Geom. Theory Appl.*, 8:181–193, 1997.

59 GEOMETRIC APPLICATIONS OF THE GRASSMANN-CAYLEY ALGEBRA

Neil L. White

INTRODUCTION

Grassmann-Cayley algebra is first and foremost a means of translating synthetic projective geometric statements into invariant algebraic statements in the bracket ring, which is the ring of projective invariants. A general philosophical principle of invariant theory, sometimes referred to as *Gram's theorem*, says that any projectively invariant geometric statement has an equivalent expression in the bracket ring; thus we are providing here the practical means to carry this out. We give an introduction to the basic concepts, and illustrate the method with several examples from projective geometry, rigidity theory, and robotics.

59.1 BASIC CONCEPTS

Let P be a $(d-1)$ -dimensional projective space over the field F , and V the canonically associated d -dimensional vector space over F . Let S be a finite set of n points in P and, for each point, fix a homogeneous coordinate vector in V . We assume that S spans V , hence also that $n \geq d$. Initially, we choose all of the coordinates to be distinct, algebraically independent indeterminates in F , although we can always specialize to the actual coordinates we want in applications. For $p_i \in S$, let the coordinate vector be $(x_{1,i}, \dots, x_{d,i})$.

GLOSSARY

Bracket: A $d \times d$ determinant of the homogeneous coordinate vectors of d points in S . Brackets are relative projective invariants, meaning that under projective transformations their value changes only in a very predictable way (in fact, under a basis change of determinant 1, they are literally invariant). Hence brackets may also be thought of as coordinate-free symbolic expressions. The bracket of u_1, \dots, u_d is denoted by $[u_1, \dots, u_d]$.

Bracket ring: The ring B generated by the set of all brackets of d -tuples of points in S , where $n = |S| \geq d$. It is a subring of the ring $F[x_{1,1}, x_{1,2}, \dots, x_{d,n}]$ of polynomials in the coordinates of points in S .

Straightening algorithm: A normal form algorithm in the bracket ring.

Join of points: An exterior product of k points, $k \leq d$, computed in the exterior algebra of V . We denote such a product by $a_1 \vee a_2 \vee \dots \vee a_k$, or simply $a_1 a_2 \cdots a_k$, rather than $a_1 \wedge a_2 \wedge \dots \wedge a_k$, which is commonly used in exterior algebra. A

concrete version of this operation is to compute the Plücker coordinate vector of (the subspace spanned by) the k points, that is, the vector whose components are all $k \times k$ minors (in some prespecified order) of the $d \times k$ matrix whose columns are the homogeneous coordinates of the k points.

Extensor of step k , or decomposable k -tensor: A join of k points. Extensors of step k span a vector space $V^{(k)}$ of dimension $\binom{d}{k}$. (Note that not every element of $V^{(k)}$ is an extensor.)

Antisymmetric tensor: Any element of the direct sum $\Lambda V = \bigoplus_k V^{(k)}$.

Copoint: Any antisymmetric tensor of step $d-1$. A copoint is always an extensor.

Join: The exterior product operation on ΛV . The join of two tensors can always be reduced by distributivity to a linear combination of joins of points.

Integral: $E = u_1 u_2 \cdots u_d$, for any vectors u_1, u_2, \dots, u_d such that $[u_1, u_2, \dots, u_d] = 1$. Every extensor of step d is a scalar multiple of the integral E .

Meet: If $A = a_1 a_2 \cdots a_j$ and $B = b_1 b_2 \cdots b_k$, with $j + k \geq d$, then

$$\begin{aligned} A \wedge B &= \sum_{\sigma} \text{sgn}(\sigma) [a_{\sigma(1)}, \dots, a_{\sigma(d-k)}, b_1, \dots, b_k] a_{\sigma(d-k+1)} \cdots a_{\sigma(j)} \\ &\equiv [\overset{\bullet}{a}_1, \dots, \overset{\bullet}{a}_{d-k}, b_1, \dots, b_k] \overset{\bullet}{a}_{d-k+1} \cdots \overset{\bullet}{a}_j . \end{aligned}$$

The sum is taken over all permutations σ of $\{1, 2, \dots, j\}$ such that $\sigma(1) < \sigma(2) < \dots < \sigma(d-k)$ and $\sigma(d-k+1) < \sigma(d-k+2) < \dots < \sigma(j)$. Each such permutation is called a *shuffle* of the $(d-k, j-(d-k))$ split of A , and the dots represent such a signed sum over all the shuffles of the dotted symbols.

Grassmann-Cayley algebra: The vector space $\Lambda(V)$ together with the operations \vee and \wedge .

PROPERTIES OF GRASSMANN-CAYLEY ALGEBRA

- (i) $A \vee B = (-1)^{jk} B \vee A$ and $A \wedge B = (-1)^{(d-k)(d-j)} B \wedge A$, if A and B are extensors of steps j and k .
- (ii) \vee and \wedge are associative and distributive over addition and scalar multiplication.
- (iii) $A \vee B = (A \wedge B) \vee E$ if $\text{step}(A) + \text{step}(B) = d$.
- (iv) A meet of two extensors is again an extensor.
- (v) The meet is dual to the join, where duality exchanges points and copoints.
- (vi) **Alternative Law:** Let a_1, a_2, \dots, a_k be points and $\gamma_1, \gamma_2, \dots, \gamma_s$ copoints. Then if $k \geq s$,

$$(a_1 a_2 \cdots a_k) \wedge (\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_s) = [\overset{\bullet}{a}_1, \gamma_1] [\overset{\bullet}{a}_2, \gamma_2] \cdots [\overset{\bullet}{a}_s, \gamma_s] \overset{\bullet}{a}_{s+1} \vee \cdots \vee \overset{\bullet}{a}_k .$$

Here the dots refer to all shuffles over the $(1, 1, \dots, 1, k-s)$ split of $a_1 \cdots a_k$, that is, a signed sum over all permutations of the a 's such that the last $k-s$ of them are in increasing order.

59.2 GEOMETRY \leftrightarrow G.-C. ALGEBRA \rightarrow BRACKET ALGEBRA

If X is a projective subspace of dimension $k - 1$, pick a basis a_1, a_2, \dots, a_k and let $A = a_1 a_2 \cdots a_k$ be an extensor. We call $X = \overline{A}$ the **support** of A.

- (i) If $A \neq 0$ is an extensor, then A determines \overline{A} uniquely.
 - (ii) If $\overline{A} \cap \overline{B} \neq \emptyset$, then $\overline{A \vee B} = \overline{A} + \overline{B}$.
 - (iii) If $\overline{A} \cup \overline{B}$ spans V , then $\overline{A \wedge B} = \overline{A} \cap \overline{B}$.
-

TABLE 59.2.1 Examples of geometric conditions and corresponding Grassmann-Cayley algebra statements.

GEOMETRIC CONDITION	DIM	G.-C. ALGEBRA STATEMENT	BRACKET STATEMENT
Point \overline{a} is on the line \overline{bc} (or \overline{b} is on \overline{ac} , etc.)	2	$a \wedge bc = 0$	$[abc] = 0$
Lines \overline{ab} and \overline{cd} intersect	3	$ab \wedge cd = 0$	$[abcd] = 0$
Lines \overline{ab} , \overline{cd} , \overline{ef} concur	2	$ab \wedge cd \wedge ef = 0$	$\overset{\bullet}{[acd]} \overset{\bullet}{[bef]} = 0$
Planes \overline{abc} , \overline{def} , and \overline{ghi} have a line in common	3	$abc \wedge def \wedge ghi = 0$	$\overset{\bullet}{[abcdef]} \overset{\bullet}{[bghi]} \overset{\bullet}{[cxyz]} = 0 \forall x, y, z$
The intersections of \overline{ab} with \overline{cd} and of \overline{ef} with \overline{gh} are collinear with \overline{i}	2	$(ab \wedge cd) \vee (ef \wedge gh) \vee i = 0$	$\overset{\bullet}{[acd]} \overset{\diamond}{[egh]} \overset{\bullet}{[bf]} \overset{\diamond}{[i]} = 0$

The geometric conditions in Table 59.2.1 should be interpreted projectively. For example, the concurrency of three lines includes as a special case that the three lines are mutually parallel, if one prefers to interpret the conditions in affine space. Degenerate cases are always included, so that the concurrency of three lines includes as a special case the equality of two or even all three of the lines, for example.

Most of the interesting geometric conditions translate into Grassmann-Cayley conditions of step 0 (or, equivalently, step d), and therefore expand into bracket conditions directly. When the Grassmann-Cayley condition is not of step 0, as in the example in Table 59.2.1 of three planes in three-space containing a common line, then the Grassmann-Cayley condition may be joined with an appropriate number of universally quantified points to get a conjunction of bracket conditions. The joined points may also be required to come from a specified basis to make this a conjunction of a finite number of bracket conditions.

In this fashion, any incidence relation in projective geometry may be translated into a conjunction of Grassmann-Cayley statements, and, conversely, Grassmann-Cayley statements may be translated back to projective geometry just as easily, provided they involve only join and meet, not addition.

Many identities in the Grassmann-Cayley algebra yield algebraic, coordinate-free proofs of important geometric theorems. These proofs typically take the form “the left-hand side of the identity is 0 if and only if the right-hand side of the identity is 0,” and the resulting equivalent Grassmann-Cayley conditions translate to interesting geometric conditions as above.

TABLE 59.2.2 Examples of Grassmann-Cayley identities and corresponding geometric theorems, in dimension 2.

GEOMETRIC THEOREM	G.-C. ALGEBRA IDENTITY
Desargues's theorem: Derived points $ab \wedge a'b'$, $ac \wedge a'c'$, and $bc \wedge b'c'$ are collinear if and only if abc or $a'b'c'$ are collinear or aa' , bb' , and cc' concur.	$(ab \wedge a'b') \vee (ac \wedge a'c') \vee (bc \wedge b'c') = [abc][a'b'c'](aa' \wedge bb' \wedge cc')$
Pappus's theorem and Pascal's theorem: If abc and $a'b'c'$ are both collinear sets, then $(bc' \wedge b'c)$, $(ca' \wedge c'a)$, and $(ab' \wedge a'b)$ are collinear.	$[ab'c'][a'bc'][a'b'c][abc] - [abc'][ab'c][a'bc]\underline{[a'b'c']} = (bc' \wedge b'c) \vee (ca' \wedge c'a) \vee (ab' \wedge a'b)$
Pappus's theorem (alternate version): If $aa'x$, $bb'x$, $cc'x$, $ab'y$, $bc'y$, and $ca'y$ are collinear, then ac' , ba' , cb' concur.	$aa' \wedge bb' \wedge cc' + ab' \wedge bc' \wedge ca' + ac' \wedge ba' \wedge cb' = 0$
Fano's theorem: If no three of a, b, c, d are collinear, then $(ab \wedge cd)$, $(bc \wedge ad)$, and $(ca \wedge bd)$ are collinear if and only if $\text{char } F = 2$.	$(ab \wedge cd) \vee (bc \wedge ad) \vee (ca \wedge bd) = 2[abc][abd][acd][bcd]$

The identities in Table 59.2.2 are proved by expanding both sides, using the rules for join and meet, and then verifying the equality of the resulting expressions by using the straightening algorithm of bracket algebra (see [Stu93]).

The right-hand side of the identity for the first version of Pappus's theorem is also the Grassmann-Cayley form of the geometric construction used in Pascal's theorem, and hence is 0 if and only if the six points lie on a common conic (Pappus's theorem being the degenerate case of Pascal's theorem in which the conic consists of two lines). Hence the left-hand side of the same identity is the bracket expression that is 0 if and only if the six points lie on a common conic. In particular, if abc and $a'b'c'$ are both collinear, we see immediately from the underlined brackets that the left-hand side is 0.

Numerous other projective geometry incidence theorems may be proved using the Grassmann-Cayley algebra. We illustrate this with an example modified from [RS76]. Other examples may be found in the same reference.

THEOREM 59.2.1

In 3-space, if triangles abc and $a'b'c'$ are in perspective from the point d , then the lines $a'bc \wedge ab'c'$, $b'ca \wedge bc'a'$, $c'ab \wedge ca'b'$, and $a'b'c' \wedge abc$ are all coplanar.

Proof. We prove the general case, where a, b, c, d, a', b', c' are all distinct, triangles abc and $a'b'c'$ are nondegenerate, and d is in neither the plane abc nor the plane $a'b'c'$. Then, since a, a', d are collinear, we may write $\alpha a' = \beta a + d$ for nonzero scalars α and β . Since we are using homogeneous coordinates for points, a , and similarly a' , may be replaced by nonzero scalar multiples of themselves without changing the geometry. Thus, without loss of generality, we may write $a' = a + d$. Similarly, $b' = b + d$ and $c' = c + d$. Now

$$\begin{aligned} L_1 &:= a'bc \wedge ab'c' = [a'ab'c']bc - [bab'c']a'c + [cab'c']a'b \\ &= [dabc]bc + [badc]ca + [cabd]ab + [badc]cd + [cabd]db \\ &= [abcd](-bc - ac + ab + cd - bd). \end{aligned}$$

Similarly,

$$L_2 := b'ca \wedge bc'a' = [abcd](ac + ab + bc + ad - cd),$$

$$L_3 := c'ab \wedge ca'b' = [abcd](-ab + bc - ac + bd - ad),$$

$$L_4 := a'b'c' \wedge abc = [abcd](bc - ac + ab).$$

Now we check that any two of these lines intersect. For example,

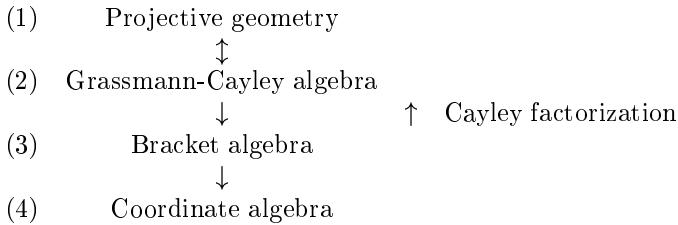
$$L_1 \wedge L_2 = [abcd]^2(-bc - ac + ab + cd - bd) \wedge (ac + ab + bc + ad - cd) = 0.$$

However, this shows only that either all four lines are coplanar or all four lines concur. To prove the former, it suffices to check that the intersection of $\overline{L_1}$ and $\overline{L_4}$ is distinct from that of $\overline{L_2}$ and $\overline{L_4}$. Notice that $L_1 \wedge L_4$ does not tell us the point of intersection, because $\overline{L_1}$ and $\overline{L_4}$ do not jointly span V , by our previous computation. But if we choose a generic vector x representing a point in general position, it follows from $\overline{L_1} \neq \overline{L_4}$, which must hold in our general case, that $(L_1 \vee x) \wedge L_4$ is nonzero and does represent the desired point of intersection. Then we compute

$$\begin{aligned} (L_1 \vee x) \wedge L_4 &= [abcd]^2(-bcx - acx + abx + cdx - bdx) \wedge (bc - ac + ab) \\ &= [abcd]^2(2[abcx] - [bedx] - [acdx])(c - b) \\ &= \alpha(c - b) \end{aligned}$$

for some nonzero scalar α . Similarly, $(L_2 \vee x) \wedge L_4 = \beta(c - a)$ for some nonzero scalar β . By the nondegeneracy of the triangle abc , these two points of intersection are distinct. \square

59.3 CAYLEY FACTORIZATION: BRACKET ALGEBRA \rightarrow GEOMETRY



$(1) \leftrightarrow (2) \rightarrow (3)$ in the chart above is explained in Section 59.2 above, with $(2) \rightarrow (1)$ being straightforward only in the case of a Grassmann-Cayley expression involving only joins and meets. $(3) \rightarrow (4)$ is the trivial expansion of a determinant into a polynomial in its d^2 entries. $(4) \rightarrow (3)$ is possible only for invariant polynomials (under the special linear group); see [Stu93] for an algorithm.

PHILOSOPHY OF INVARIANT THEORY: It is best for many purposes to avoid level (4), and to work instead with the symbolic coordinate-free expressions on levels (2) and (3).

Cayley factorization, (3)→(2), refers to the translation of a bracket polynomial into an equivalent Grassmann-Cayley expression involving only joins and meets. The input polynomial must be homogeneous (i.e., each point must occur the same number of times in the brackets of each bracket monomial of the polynomial), and Cayley factorization is not always possible. No practical algorithm is known in general, but an algorithm [Whi91] is known that finds such a factorization—or else announces its impossibility—in the multilinear case (each point occurs exactly once in each monomial). This algorithm is practical up to about 20 points.

MULTILINEAR CAYLEY FACTORIZATION

The multilinear Cayley factorization (MCF) algorithm is too complex to present here in detail; instead, we give an example and indicate roughly how the algorithm proceeds on the example.

Let

$$\begin{aligned} P = & -[acj][deh][bfg] - [cdj][aeh][bfg] - [cdj][abe][fgh] \\ & + [acj][bdf][egh] - [acj][bdg][efh] + [acj][bdh][efg]. \end{aligned}$$

Note that P is multilinear in the 9 points. The MCF algorithm now looks for sets of points x, y, \dots, z such that the extensor $xy \dots z$ could be part of a Cayley factorization of P . For this choice of P , it turns out that no such set larger than a pair of elements occurs. An example of such a pair is a, d ; in fact, if d is replaced by a in P , leaving two a 's in each term of P , although in different brackets, the resulting bracket polynomial is equal to 0, as can be verified using the straightening algorithm. The MCF algorithm, using the straightening algorithm as a subroutine, finds that $(a, d), (b, h), (c, j), (f, g)$ are all the pairs with this property.

The algorithm now looks for combinations of these extensors that could appear as a meet in a Cayley factorization of P . (For details, see [Whi91].) It finds in our example that $ad \wedge cj$ is such a combination. As soon as a single such combination is found, an algebraic substitution involving a new variable, $z = ad \wedge cj$, is performed, and a new bracket polynomial of smaller degree involving this new variable is derived; the algorithm then begins anew on this polynomial. If no such combination is found, the input bracket polynomial is then known to have no Cayley factorization. In our example, this derived polynomial turns out to be $P = [ze][gbh] - [zeg][fbh]$, which of necessity is still multilinear. The MCF algorithm proceeds to find (and we can directly see by consulting [Table 59.2.1](#)) that $P = ze \wedge fg \wedge bh$. Thus, our final Cayley factorization is output as

$$P = ((ad \wedge cj) \vee e) \wedge fg \wedge bh.$$

It is significant that this algorithm requires no backtracking. For example, once $ad \wedge cj$ is found as a possible meet in a Cayley factorization of P , it is known that if P has a Cayley factorization at all, then it must also have one using the factor $ad \wedge cj$; hence we are justified in factoring it out, i.e., substituting a new variable for it. Other Cayley factorizations may be possible, for example,

$$P = ((fg \wedge bh) \vee (ad \wedge cj)) \wedge e.$$

Note that these two factorizations have the same geometric meaning.

59.4 APPLICATIONS

59.4.1 ROBOTICS

GLOSSARY

Robot arm: A set of rigid bodies, or links, connected in series by joints that allow relative movement of the successive links, as described below. The first link is regarded as fixed in position, or tied to the ground, while the last link, called the *end-effector*, is the one that grasps objects or performs tasks.

Revolute joint: A joint between two successive links of a robot arm that allows only a rotation between them. In simpler terms, a revolute joint is a hinge connecting two links.

Prismatic joint: A joint between two successive links of a robot arm that allows only a translational movement between the two links.

Screw joint: A joint between two successive links of a robot arm that allows only a screw movement between the two links.

TABLE 59.4.1 Modeling instantaneous robotics.

ROBOTICS CONCEPT	GRASSMANN-CAYLEY EQUIVALENT
Revolute joint on axis \overline{ab}	$\alpha(a \vee b)$, a 2-extensor
Rotation about line \overline{ab}	$\beta(a \vee b)$
Motion of point p in rotation about line \overline{ab}	$\beta(a \vee b) \vee p$
Screw joint	indecomposable 2-tensor
Prismatic joint	2-extensor at infinity
Motion space of the end-effector, where j_1, j_2, \dots, j_k are joints in series	span of the extensors $\langle j_1, j_2, \dots, j_k \rangle$

We are considering here only the instantaneous kinematics or statics of robot arms, that is, positions and motions at a given instant in time. A robot arm has a *critical configuration* if the joint extensors become linearly dependent. If the arm has six joints in three-space, a critical configuration means a loss of full mobility. If the arm has a larger number of joints, criticality is defined as any six of the joint extensors becoming linearly dependent. This can mean severe problems with the driving program in real-life robots, even when the motion space retains full dimensionality.

In one sense, criticality is trivial to determine, since we need only compute a determinant function, called the *superbracket*, on the six-dimensional space $\Lambda^2(V)$. However, if we want to know all the critical configurations of a given robot arm, this becomes a nontrivial question, that of determining all of the zeroes of the superbracket. To answer it, we need to express the superbracket in terms of ordinary brackets. This has been done in [MW91], where the superbracket of the six 2-extensors $a_1a_2, b_1b_2, \dots, f_1f_2$ is given by

$$\begin{aligned}
[[a_1 a_2, b_1 b_2, c_1 c_2, d_1 d_2, e_1 e_2, f_1 f_2]] &= \\
&- [a_1 a_2 b_1 b_2] [c_1 c_2 \overset{\bullet}{d}_1 \overset{\diamond}{e}_1] [\overset{\bullet}{d}_2 \overset{\diamond}{e}_2 f_1 f_2] \\
&+ [a_1 a_2 \overset{\bullet}{b}_1 \overset{\diamond}{c}_1] [\overset{\bullet}{b}_2 \overset{\diamond}{c}_2 d_1 d_2] [e_1 e_2 f_1 f_2] \\
&- [a_1 a_2 \overset{\bullet}{b}_1 \overset{\diamond}{c}_1] [\overset{\bullet}{b}_2 d_1 d_2 \overset{\triangleleft}{e}_1] [\overset{\diamond}{c}_2 \overset{\triangleleft}{e}_2 f_1 f_2] \\
&+ [a_1 a_2 \overset{\bullet}{b}_1 d_1] [\overset{\bullet}{b}_2 c_1 c_2 \overset{\triangleleft}{e}_1] [\overset{\diamond}{d}_2 \overset{\triangleleft}{e}_2 f_1 f_2].
\end{aligned}$$

(Here the dots, diamonds, and triangles have the same meaning as the dots in Section 59.1.)

Consider the particular example of the six-revolute-joint robot arm illustrated in Figure 59.4.1, whose first two joints lie on intersecting lines, whose third and fourth joints are parallel, and whose last two joints also lie on intersecting lines. The larger cylinders in the figure represent the revolute joints. To express the superbracket, we must choose two points on each joint axis. We may choose $b_1 = a_2$, $d_1 = c_2$ (where this point is at infinity), and $f_1 = e_2$, as shown by the black dots. The thin cylinders represent the links; for example, the first link, between a_2 and b_2 , is connected to the ground (not shown) by joint $a_1 a_2$, and can therefore only rotate around the axis $\overline{a_1 a_2}$.

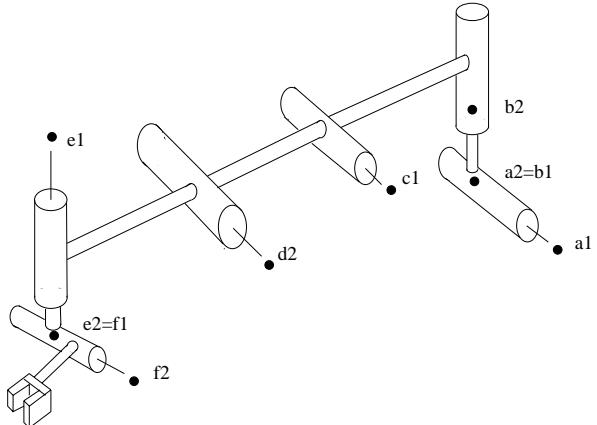


FIGURE 59.4.1
Six-revolute-joint robot arm.

Plugging in and deleting terms with a repeated point inside a bracket, we get

$$- [a_1 a_2 b_1 \overset{\bullet}{c}_1] [a_2 c_2 d_2 e_2] [\overset{\bullet}{c}_2 e_1 e_2 f_2] \quad (59.4.1)$$

$$+ [a_1 a_2 b_1 \overset{\bullet}{c}_2] [a_2 c_1 c_2 e_2] [\overset{\bullet}{d}_2 e_1 e_2 f_2] \quad (59.4.2)$$

$$= [\overset{\bullet}{c}_1 a_1 a_2 b_2] [\overset{\bullet}{d}_2 a_2 c_2 e_2] [\overset{\bullet}{c}_2 e_1 e_2 f_2], \quad (59.4.3)$$

where each of (59.4.1) and (59.4.2) has two terms because of the dotting, and the same four terms constitute (59.4.3), since two of the six terms generated by the dotting are zero because of the repetition of c_2 in the second bracket.

Finally, we recognize (59.4.3) as the bracket expansion of

$$(c_1 d_2 c_2) \wedge (a_1 a_2 b_2) \wedge (a_2 c_2 e_2) \wedge (e_1 e_2 f_2).$$

We then recognize that the geometric conditions for criticality are any positions that make this Grassmann-Cayley expression 0, namely

- (i) one or more of the planes $\overline{c_1c_2d_2}, \overline{a_1a_2b_2}, \overline{a_2c_2e_2}, \overline{e_1e_2f_2}$ is degenerate, or
- (ii) the four planes have nonempty intersection.

Notice that in an actual robot arm of the type we are considering, none of the degeneracies in (i) can actually occur.

See [Section 48.1](#) for more information.

59.4.2 BAR FRAMEWORKS

Consider a generically $(d-1)$ -isostatic graph G (see [Section 60.1](#) of this Handbook), that is, a graph for which almost all realizations in $(d-1)$ -space as a bar framework are minimally first-order rigid. Since first-order rigidity is a projective invariant (see [Theorem 60.1.23](#)), we would like to know the projective geometric conditions that characterize all of its nonrigid (first-order flexible) realizations. By Gram's theorem, these conditions must be expressible in terms of bracket conditions, and [WW83] shows that the first-order flexible realizations are characterized by the zeroes of a single bracket polynomial C_G , called the *pure condition* (see [Theorem 60.1.25](#)). Furthermore, [WW83] gives an algorithm to construct the pure condition C_G directly from the graph G . Then we require Cayley factorization to recover the geometric incidence condition, if it is not already known. Consider the following examples, illustrated in Figure 59.4.2.

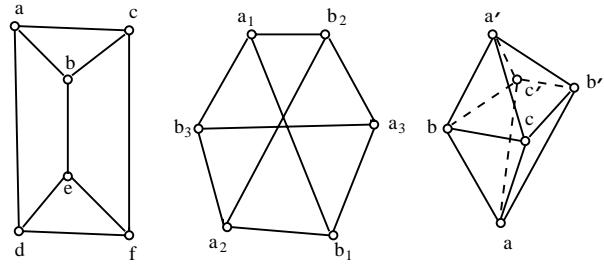


FIGURE 59.4.2

Three examples of bar frameworks.

- (i) The graph G is the edge skeleton of a triangular prism, realized in the plane. We have $C_G = [abc][def]([abe][dfc] - [dbe][afc])$, and we may recognize the factor in parentheses as the third example in [Table 59.2.1](#). Thus $C_G = 0$, and the framework is first-order flexible, if and only if one of the triangles \overline{abc} or \overline{def} is degenerate, or the three lines \overline{ad} , \overline{be} , \overline{cf} are concurrent, or one or more of these lines is degenerate.
- (ii) The graph G is $K_{3,3}$, a complete bipartite graph, realized in the plane. Then $C_G = [a_1a_2a_3][a_1b_2b_3][b_1a_2b_3][b_1b_2a_3] - [b_1b_2b_3][b_1a_2a_3][a_1b_2a_3][a_1a_2b_3]$, and this is the second example in [Table 59.2.2](#). Thus $C_G = 0$, and the framework is first-order flexible, if and only if the six points lie on a common conic or, equivalently by Pascal's theorem, the three points $\overline{a_1b_2 \wedge a_2b_1}$, $\overline{a_1b_3 \wedge a_3b_1}$, $\overline{a_2b_3 \wedge a_3b_2}$ are collinear.
- (iii) The graph G is the edge skeleton of an octahedron, realized in Euclidean 3-space. Then $C_G = [abc'a'][bca'b'][cab'c'] + [abc'b'][bca'c'][cab'a']$, and this can

be recognized directly as the expansion of the Grassmann-Cayley expression $abc \wedge a'b'c' \wedge a'b'c \wedge ab'c'$. Thus $C_G = 0$, and the framework is first-order flexible, if and only if the four alternating octahedral face planes \overline{abc} , $\overline{a'b'c}$, $\overline{a'b'c}$, and $\overline{ab'c'}$ concur, or any one or more of these planes is degenerate. This, in turn, is equivalent to the same condition on the other four face planes, \overline{abc} , $\overline{ab'c}$, $\overline{a'b'c}$, $\overline{a'b'c'}$.

59.4.3 BAR-AND-BODY FRAMEWORKS

A ***bar-and-body framework*** consists of a finite number of $(d-1)$ -dimensional rigid bodies, free to move in Euclidean $(d-1)$ -space, and connected by rigid bars, with the connections at the ends of each bar allowing free rotation of the bar relative to the rigid body; i.e., the connections are “universal joints.” Each rigid body may be replaced by a first-order rigid bar framework in such a way that the result is one large bar framework, thus in one sense reducing the study of bar-and-body frameworks to that of bar frameworks. Nevertheless, the combinatorics of bar-and-body frameworks is quite different from that of bar frameworks, since the original rigid bodies are not allowed to become first-order flexible in any realization, contrary to the case with bar frameworks.

A generically isostatic bar-and-body framework has a pure condition, just as a bar framework has, whose zeroes are precisely the special positions in which the framework has a first-order flex. However, this pure condition is a bracket polynomial in the *bars* of the framework, as opposed to a bracket polynomial in the vertices, as was the case with bar frameworks. An algorithm to directly compute the pure condition for a bar-and-body framework, somewhat similar to that for bar frameworks, is given in [WW87]. We illustrate with the example in Figure 59.4.3, consisting of three rigid bodies and six bars in the plane. We may interpret the word “plane” here as “real projective plane.”

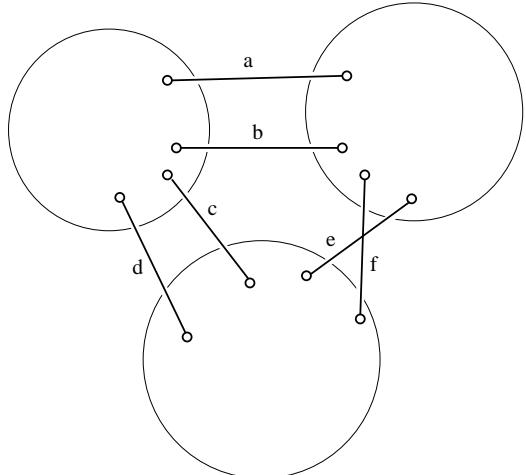


FIGURE 59.4.3
A *bar-and-body framework*.

Hence $V = \mathbb{R}^3$, and we let $W = \Lambda^2(V) \cong V^* \cong \mathbb{R}^3$. We think of the endpoints of the bars as elements of V , and hence the lines determined by the bars are two-extensors of these points, or elements of W . The algorithm produces the pure condition $[abc][def] - [abd][cef]$. This bracket polynomial may be Cayley factored

as $ab \wedge cd \wedge ef$, as seen in [Table 59.2.1](#). Now we switch to thinking of a, b, \dots, f as 2-extensors in V rather than elements of W , and recall that there is a duality between V and W , hence between $\Lambda(V)$ and $\Lambda(W)$. Thus, the framework has a first-order flex if and only if $(a \wedge b) \vee (c \wedge d) \vee (e \wedge f) = 0$ in $\Lambda(V)$. Hence the desired geometric condition for the existence of a first-order flex is that the three points $\overline{a \wedge b}$, $\overline{c \wedge d}$, and $\overline{e \wedge f}$ are collinear. Now $\overline{a \wedge b}$ is just the center of relative (instantaneous) motion for the two bodies connected by those two bars: think of fixing one of the bodies and then rotating the other body about this center; the lengths of the two bars are instantaneously preserved. The geometric result we have obtained is just a restatement of the classical theorem of Arnhold-Kempe that in any flex of three rigid bodies, the centers of relative motion of the three pairs of bodies must be collinear.

59.4.4 AUTOMATED GEOMETRIC THEOREM-PROVING

J. Richter-Gebert [RG95] uses Grassmann-Cayley algebra to derive bracket conditions for projective geometric incidences in order to produce coordinate-free automatic proofs of theorems in projective geometry. By introducing two circular points at infinity, the same can be done for theorems in Euclidean geometry [CRG95].

Richter-Gebert's technique is to reduce each hypothesis to a *binomial* equation, that is, an equation with a single product of brackets on each side. For example, as we have seen, the concurrence of three lines $\overline{ab}, \overline{cd}, \overline{ef}$ may be rewritten as $[acd][bef] = [bcd][aef]$. Similarly, the collinearity of three points $\overline{a}, \overline{b}, \overline{c}$ may be expressed as $[abd][bce] = [abe][bcd]$, avoiding the much more obvious expression $[abc] = 0$ since it is not of the required form. If all binomial equations are now multiplied together, and provided they were appropriately chosen in the first place, common factors may be canceled (which involves nondegeneracy assumptions, so that the common factors are nonzero), resulting in the desired conclusion. A surprising array of theorems may be cast in this format, and this approach has been successfully implemented.

More recent work along similar lines, extending it especially to conic geometry, is by H. Li and Y. Wu [LW03a, LW03b].

59.4.5 COMPUTER VISION

Much of computer vision study involves projective geometry, and hence is very amenable to the techniques of the Grassmann-Cayley algebra. One reference that explicitly applies these techniques to a system of up to three pinhole cameras is Faugeras and Papadopoulo [FP98].

59.5 SOURCES AND RELATED MATERIAL

SURVEYS

[DRS74] and [BBR85]: These two papers survey the properties of the Grassmann-Cayley algebra (called the “double algebra” in [BBR85]).

[Whi95]: A more elementary survey than the two above.

[Whi94]: Emphasizes the concrete approach via Plücker coordinates, and gives more detail on the connections to robotics.

RELATED CHAPTERS

[Chapter 9: Geometry and topology of polygonal linkages](#)

[Chapter 48: Robotics](#)

[Chapter 60: Rigidity and scene analysis](#)

REFERENCES

- [BBR85] M. Barnabei, A. Brini, and G.-C. Rota. On the exterior calculus of invariant theory. *J. Algebra*, 96:120–160, 1985.
- [CRG95] H. Crapo and J. Richter-Gebert. Automatic proving of geometric theorems. In N. White, editor, *Invariant Methods in Discrete and Computational Geometry*, pages 167–196. Kluwer, Dordrecht, 1995.
- [DRS74] P. Doubilet, G.-C. Rota, and J. Stein. On the foundations of combinatorial theory: IX, combinatorial methods in invariant theory. *Stud. Appl. Math.*, 53:185–216, 1974.
- [FP98] O. Faugeras and T. Papadopoulo. Grassmann-Cayley algebra for modelling systems of cameras and the algebraic equations of the manifold of trifocal tensors. *Philos. Trans. Roy. Soc. London, Ser. A*, 356:1123–1152, 1998.
- [LW03a] H. Li and Y. Wu. Automated short proof generation for projective geometric theorems with Cayley and bracket algebras, I. Incidence geometry. *J. Symbolic Comput.*, 36:717–762, 2003.
- [LW03b] H. Li and Y. Wu. Automated short proof generation for projective geometric theorems with Cayley and bracket algebras, II. Conic geometry. *J. Symbolic Comput.*, 36:763–809, 2003.
- [MW91] T. McMillan and N. White. The dotted straightening algorithm. *J. Symbolic Comput.*, 11:471–482, 1991.
- [RG95] J. Richter-Gebert. Mechanical theorem proving in projective geometry. *Ann. Math. Artif. Intell.*, 13:139–172, 1995.
- [RS76] G.-C. Rota and J. Stein. Applications of Cayley algebras. In *Colloquio Internazionale sulle Teorie Combinatorie*, pages 71–97. Accademia Nazionale dei Lincei, 1976.
- [Stu93] B. Sturmfels. *Algorithms in Invariant Theory*. Springer-Verlag, New York, 1993.
- [Whi91] N. White. Multilinear Cayley factorization. *J. Symbolic Comput.*, 11:421–438, 1991.
- [Whi94] N. White. Grassmann-Cayley algebra and robotics. *J. Intell. Robot. Syst.*, 11:91–107, 1994.
- [Whi95] N. White. A tutorial on Grassmann-Cayley algebra. In N. White, editor, *Invariant Methods in Discrete and Computational Geometry*, pages 93–106. Kluwer, Dordrecht, 1995.
- [WW83] N. White and W. Whiteley. The algebraic geometry of stresses in frameworks. *SIAM J. Algebraic Discrete Methods*, 4:481–511, 1983.
- [WW87] N. White and W. Whiteley. The algebraic geometry of motions in bar-and-body frameworks. *SIAM J. Algebraic Discrete Methods*, 8:1–32, 1987.

60 RIGIDITY AND SCENE ANALYSIS

Walter Whiteley

INTRODUCTION

Rigidity and flexibility of frameworks (motions preserving lengths of bars) and scene analysis (liftings from plane polyhedral pictures to spatial polyhedra) are two core examples of a general class of geometric problems:

- (a) Given a discrete configuration of points, lines, planes, ... in Euclidean space, and a set of geometric constraints (fixed lengths for rigidity, fixed incidences, and fixed projections of points for scene analysis), what is the set of solutions and what is its local form: discrete? k -dimensional?
- (b) Given a structure satisfying the constraints, is it unique, or at least locally unique, up to trivial changes, such as congruences for rigidity, or vertical scale for liftings?
- (c) How does this answer depend on the combinatorics of the structure and how does it depend on the specific geometry of the initial data or object?

The rigidity of frameworks examines points constrained by fixed distances between pairs, using vocabulary and linear techniques drawn from structural engineering: bars and joints, first-order rigidity and first-order flexes, and static rigidity and static self-stresses (Section 60.1). Scene analysis and the dual concept of parallel drawings are described in Section 60.2. Finally, reciprocal diagrams form a fundamental geometric connection between liftings of polyhedral pictures and self-stresses in frameworks (Section 60.3).

These core problems have a wide range of applications across many areas of applied geometry. The methods used and the results obtained for these problems serve as a model for what might be hoped for other sets of constraints (plane first-order results) and as a warning of the complexity that does arise (higher dimensions and broader forms of rigidity). The subject has a rich history, stretching back into at least the middle of the 19th century, in structural and mechanical engineering. Other independent rediscoveries and connections have arisen in crystallography and scene analysis. Some other geometric problems with related mathematical and algorithmic patterns are mentioned in Sections 60.1.5, 60.2.3, and 60.3. For more general geometric reconstruction problems, see [Chapter 29](#).

60.1 RIGIDITY OF BAR FRAMEWORKS

Given a set of points in space, with certain distances to be preserved, what other configurations have the same distances? If we make small changes in the distances, will there be a small (linear scale) change in the position? What is the structure, locally and globally, of the algebraic variety of these “realizations”?

We begin with the simplest linear theory: first-order rigidity, and the equivalent dual static rigidity, which are the linearized (and therefore linear algebra) version of rigidity. Generic rigidity refers to first-order rigidity of “almost all” geometric positions of the underlying combinatorial structure. After the initial results presenting first-order rigidity (Section 60.1.1), the study divides into the combinatorics of generic rigidity, using graphs (Section 60.1.2); the geometry of special positions in first-order rigidity, using projective geometry (Section 60.1.3); more general concepts of rigidity (Section 60.1.4); and extensions to tensegrity frameworks, using geometry and minima of energy functions for rigidity (Section 60.1.5).

60.1.1 FIRST-ORDER RIGIDITY

GLOSSARY

Configuration of points in d -space: An assignment $p = (p_1, \dots, p_v)$ of points $p_i \in \mathbb{R}^d$ to an index set V , where $v = |V|$.

Congruent configurations: Two configurations p and q in d -space, on the same set V , related by an isometry T of \mathbb{R}^d (with $T(p_i) = q_i$ for all $i \in V$).

Bar framework in d -space $G(p)$ (or **framework**): A graph $G = (V; E)$ (no loops or multiple edges) and a configuration p in d -space for the vertices V (Figure 60.1.1A).

Bar: An edge $\{i, j\} \in E$ for a framework $G(p)$.

First-order flex or **infinitesimal motion**: For a bar framework $G(p)$, an assignment of velocities $p' : V \rightarrow \mathbb{R}^d$, such that for each edge $\{i, j\} \in E$: $(p_i - p_j) \cdot (p'_i - p'_j) = 0$ (Figure 60.1.1.C,D, where the arrows represent nonzero velocities).

Trivial first-order flex: A first-order flex p' that is the derivative of a flex of congruent frameworks (Figure 60.1.1C). (There is a fixed skew-symmetric matrix S (a rotation) and a fixed vector t (a translation) such that, for all vertices $i \in V$, $p'_i = p_i S + t$.)

First-order flexible framework: A framework $G(p)$ with a nontrivial first-order flex (Figure 60.1.1D).

First-order rigid framework: A bar framework $G(p)$ for which every first-order flex is trivial (Figures 60.1.1A, 60.1.2A).

Rigidity matrix: For a framework $G(p)$ in d -space, $R_G(p)$ is the $|E| \times d|V|$ matrix for the system of equations: $(p_i - p_j) \cdot (p'_i - p'_j) = 0$ in the unknown velocities p'_i . The first-order flex equations are expressed as

$$R_G(p)p'^T = \begin{bmatrix} \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \cdots & (p_i - p_j) & \cdots & (p_j - p_i) & \cdots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \end{bmatrix} \times p'^T = 0^T.$$

Self-stress: For a framework $G(p)$, a row dependence ω for the rigidity matrix: $\omega R_G(p) = 0$. Equivalently, an assignment of scalars ω_{ij} to the edges such that

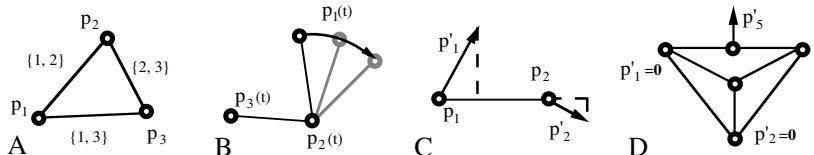
at each vertex i , $\sum_{\{j \mid \{i,j\} \in E\}} \omega_{ij}(p_i - p_j) = 0$ (placing these “internal forces” $\omega_{ij}(p_i - p_j)$ in equilibrium at vertex i). $\omega_{ij} < 0$ is tension, $\omega_{ij} > 0$ is compression.

Independent framework: A bar framework $G(p)$ for which the rigidity matrix has independent rows. Equivalently, there is only the zero self-stress.

Isostatic framework: A framework $G(p)$ that is first-order rigid and independent.

Generically rigid graph in d -space: A graph G for which the frameworks $G(p)$ are first-order rigid on an open dense subset of configurations p in d -space (Figures 60.1.1A, 60.1.2A).

FIGURE 60.1.1



Generic d -circuit: A graph G such that with the deletion of any edge e , $G - e$ is generically rigid in d -space.

BASIC CONNECTIONS

Because the constraints $|p_i - p_j| = |q_i - q_j|$ are algebraic in the coordinates of the points (after squaring), we can work with the Jacobian matrix formed by the partial derivatives of these equations—the rigidity matrix of the framework.

The dimension of the space of trivial first-order motions of a framework in d -space is $\binom{d+1}{2}$ provided $|V| \geq d$ (the velocities generated by d translations and by $\binom{d}{2}$ rotations form a basis).

THEOREM 60.1.1 *First-order Rank*

A framework $G(p)$ with $|V| \geq d$ is first-order rigid if and only if the rigidity matrix $R_G(p)$ has rank $d|V| - \binom{d+1}{2}$.

A framework $G(p)$ with few vertices, $|V| \leq d$, is isostatic if and only if the rigidity matrix $R_G(p)$ has rank $\binom{|V|}{2}$ (if and only if G is the complete graph on V and the points p_i do not lie in an affine space of dimension $|V| - 2$).

First-order rigidity is linear algebra, with first-order rigid frameworks, self-stresses, and isostatic frameworks playing the roles of spanning sets, linear dependence, and bases of the row space for the rigidity matrix of the complete graph on the configuration p .

There is a dual theory of static rigidity for bar frameworks. Where first-order rigidity focuses on the kernel of the rigidity matrix (first-order flexes) and on the column space and column rank, static rigidity focuses on the cokernel of the rigidity matrix (the self-stresses) and on the row space of the rigidity matrix (the resolvable static loads). Methods from both approaches are widely used [CW82, Whi84, Whi96], although in this chapter we present the results primarily in the vocabulary of first-order rigidity.

THEOREM 60.1.2 Isostatic Frameworks

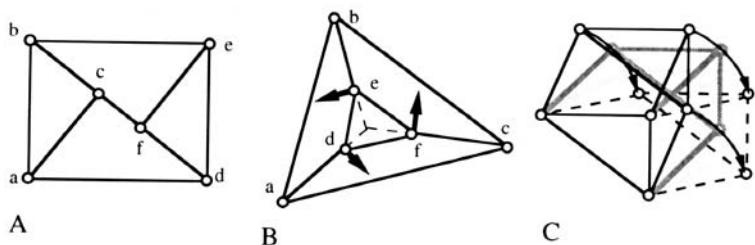
For a framework $G(p)$ in d -space, with $|V| \geq d$, the following are equivalent:

- (a) $G(p)$ is isostatic (first-order rigid and independent);
- (b) $G(p)$ is first-order rigid with $|E| = d|V| - \binom{d+1}{2}$;
- (c) $G(p)$ is independent with $|E| = d|V| - \binom{d+1}{2}$;
- (d) $G(p)$ is first-order rigid, and removing any one bar (but no vertices) leaves a first-order flexible framework.

First-order rigidity of a framework $G(p)$ is a robust property: a small change in the configuration p preserves this rigidity. Independence implies that the distances are robust: any small change in these distances can be realized by a nearby configuration. On the other hand, self-stresses mean that one of the distances is algebraically dependent on the others: many small changes in the distances will have no realizations, or no nearby realizations.

Figure 60.1.2 illustrates a single graph with plane configurations that produce: (A) a first-order rigid framework; (B) a first-order flexible, but rigid, framework, and (C) a flexible framework (see Section 60.1.4). The graph itself is generically 2-rigid.

FIGURE 60.1.2



THEOREM 60.1.3 Generic Rigidity Theorem

For a graph G and a fixed dimension d the following are equivalent:

- (a) G is generically rigid in d -space;
- (b) for each configuration $p \in \mathbb{R}^{dv}$ using algebraically independent numbers over the rationals as coordinates, the framework $G(p)$ is first-order rigid;
- (c) $G(p)$ is first-order rigid for some configuration $p \in \mathbb{R}^{dv}$.

60.1.2 COMBINATORICS FOR GENERIC RIGIDITY

The major goal in generic rigidity is a combinatorial characterization of graphs that are generically rigid in d -space. The companion problem is to find efficient combinatorial algorithms to test graphs for generic rigidity. For the plane (and the line), this is solved. Beyond the plane the results are essentially incomplete, but some significant partial results are available.

GLOSSARY

Generically d -independent: A graph G for which some (equivalently, almost all) configurations p produce independent frameworks in d -space.

Generically d -isostatic graph: A graph G for which some (equivalently, almost all) configurations p produce isostatic frameworks in d -space.

Generic d -circuit: A graph G that is dependent for all configurations p in d -space but for all edges $\{i, j\} \in E$, $G - \{i, j\}$ is generically independent in d -space.

Complete bipartite graph: A graph $K_{m,n} = (A \cup B, A \times B)$, where A and B are disjoint sets of cardinality $|A| = m$ and $|B| = n$.

Triangulated d -pseudomanifold: A finite set of d -simplices (complete graphs on $d + 1$ points) with the property that each d subset (facet) occurs in exactly two simplices, any two simplices are connected by a path of simplices and shared facets, and any two simplices sharing a vertex are connected through other simplices at this vertex. (For example, the triangles, edges, and vertices of a closed triangulated 2-surface without boundary, such as a sphere or torus, form a 2-pseudomanifold.) Cf. Section 18.3.

Henneberg d -construction for a graph G : A sequence $(V_d, E_d), \dots, (V_n, E_n)$ of graphs, such that:

- (i) For each index $d < j \leq n$, (V_j, E_j) is obtained from (V_{j-1}, E_{j-1}) by

vertex addition: attaching a new vertex by d edges (Figure 60.1.4A for $d = 2$), or

edge splitting: replacing an edge from (V_{j-1}, E_{j-1}) with a new vertex joined to its ends and to $d - 1$ other vertices (Figure 60.1.4B for $d = 2$); and

- (ii) (V_d, E_d) is the complete graph on d vertices, and $(V_n, E_n) = G$ (Figure 60.1.6A).

Proper 3Tree2 partition: A partition of the edges of a graph into three trees, such that each vertex is attached to exactly two of these trees and no nontrivial subtrees of distinct trees T_i have the same support (i.e., the same vertices) (Figure 60.1.6B).

Proper 2Tree partition: A partition of the edges of a graph into two spanning trees, such that no nontrivial subtrees of distinct trees T_i have the same support (i.e., the same vertices) (Figure 60.1.6C).

d -connected graph: A graph G such that removing any $d - 1$ vertices (and all incident edges) leaves a connected graph. (Equivalently, a graph such that any two vertices can be connected by at least d paths that are vertex-disjoint except for their endpoints.)

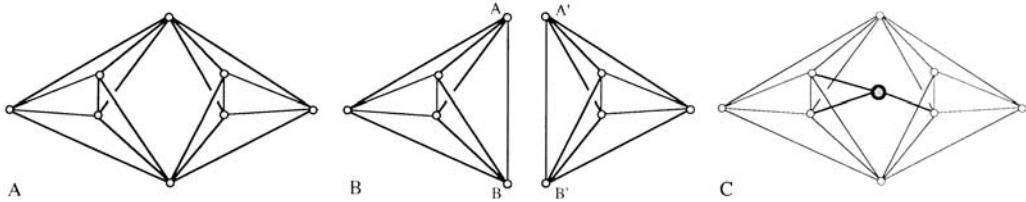
BASIC PROPERTIES IN ALL DIMENSIONS

THEOREM 60.1.4 Necessary Counts and Connectivity Theorem

If a graph G is generically d -isostatic, then, if $V \geq d$, $|E| \leq d|V| - \binom{d+1}{2}$ and for every subgraph on $|V'| \geq d$ vertices with edges E' in $V' \times V'$, $|E'| \leq d|V'| - \binom{d+1}{2}$.

If $G = (V, E)$ is a generically d -isostatic graph with $|V| > d$, then (V, E) is a d -connected graph.

FIGURE 60.1.3



For dimensions 1 and 2, the first count alone is sufficient for generic rigidity (see below). For dimensions $d > 2$, these two conditions are not enough to characterize the generically d -isostatic graphs. Figure 60.1.3A shows a generically flexible counterexample for the sufficiency of the counts in dimension 3. This example is generated by a “circuit exchange” on two over-counted graphs (Figure 60.1.3B). Figure 60.1.3C adds 3-connectivity, but preserves the flexibility and the counts.

THEOREM 60.1.5 Bipartite Graphs

A complete bipartite graph $K_{m,n}$, with $m > 1$, is generically rigid in dimension d if and only if $m + n \geq \binom{d+2}{2}$ and $m, n > d$.

INDUCTIVE CONSTRUCTIONS FOR ISOSTATIC GRAPHS

Inductive constructions for graphs that preserve generic rigidity are used both to prove theorems for general classes of frameworks and to analyze particular graphs.

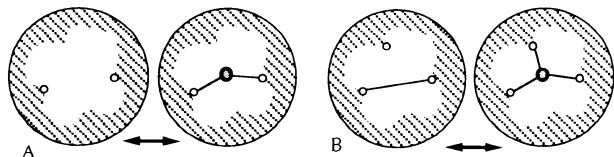
THEOREM 60.1.6 Vertex Addition Theorem

Let $G = (V, E)$ be a graph with a vertex i of valence d ; let $H = (U, F)$ denote the subgraph obtained by deleting i and the edges incident with it. Then G is generically d -isostatic if and only if H is generically d -isostatic (Figure 60.1.4A for $d = 2$).

THEOREM 60.1.7 Edge Split Theorem

Let $G = (V, E)$ be a graph with a vertex i of valence $d+1$, let S be the set of vertices adjacent to i , and let $H = (U, F)$ be the subgraph obtained by deleting i and its $d+1$ incident edges. Then G is generically d -isostatic if and only if there is a pair j, k of vertices of V such that the edge $\{j, k\}$ is not in F and the graph $H' = (U, F \cup \{j, k\})$ is generically d -isostatic (Figure 60.1.4B for $d = 2$).

FIGURE 60.1.4



THEOREM 60.1.8 Construction Theorem

If a graph G is obtained by a Henneberg d -construction, then G is generically d -isostatic (Figure 60.1.6A for $d = 2$).

THEOREM 60.1.9 Gluing Theorem

If $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are generically d -rigid graphs sharing at least d vertices, then $G = (V_1 \cup V_2, E_1 \cup E_2)$ is generically d -rigid.

THEOREM 60.1.10 Vertex Splitting Theorem

If the graph G' is a vertex split of a generically d -isostatic graph G on d edges (Figure 60.1.5A for $d = 3$) or a vertex split on $d - 1$ edges (Figure 60.1.5B for $d = 3$), then G' is generically d -isostatic.

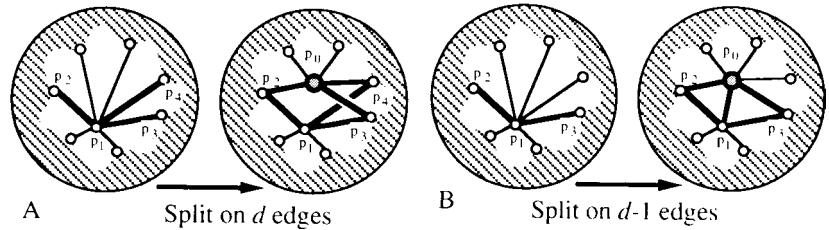


FIGURE 60.1.5

PLANE ISOSTATIC GRAPHS

Many plane results are expressed in terms of trees in the graph, building on a simpler correspondence between rigidity on the line and the connectivity of the graph.

THEOREM 60.1.11 Line Rigidity

For graph G and configuration p on the line with $p_i \neq p_j$ for all $\{i, j\} \in E$, the following are equivalent:

- (a) $G(p)$ is minimal among rigid frameworks on the line with these vertices;
- (b) $G(p)$ is isostatic on the line;
- (c) G is a spanning tree on the vertices;
- (d) $|E| = |V| - 1$ and for every nonempty subset E' with vertices V' , $|E'| \leq |V'| - 1$.

THEOREM 60.1.12 Plane Isostatic Graphs Theorem

For a graph G with $|V| \geq 2$, the following are equivalent:

- (a) G is generically isostatic in the plane;
- (b) $|E| = 2|V| - 3$, and for every subgraph (V', E') with $|V'| \geq 2$ vertices, $|E'| \leq 2|V'| - 3$ (Laman's theorem);
- (c) there is a Henneberg 2-construction for G (Henneberg's theorem);
- (d) E has a proper 3Tree2 partition (Crapo's theorem);
- (e) for each $\{i, j\} \in E$, the multigraph obtained by doubling the edge $\{i, j\}$ is the union of two spanning trees (Recski's theorem).

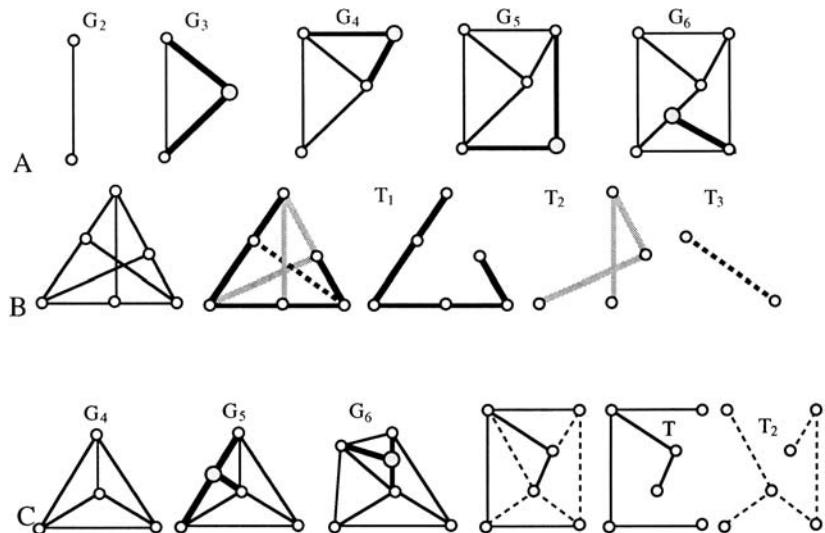


FIGURE 60.1.6

Figure 60.1.6A shows the Henneberg plane construction for the isostatic graph of [Figure 60.1.2](#). Figure 60.1.6B shows a proper 3Tree2 partition of the isostatic complete bipartite graph $K_{3,3}$. With an added edge, joining T_2 to T_3 , this partition creates several of the pairs of spanning trees predicted by Recski's theorem.

THEOREM 60.1.13 *Plane 2-Circuits Theorem*

For a graph G with $|V| \geq 2$, the following are equivalent:

- (a) G is a generic 2-circuit;
- (b) $|E| = 2|V| - 2$, and for every proper subset E' on vertices V' , $|E'| \leq 2|V'| - 3$;
- (c) there is a construction for G from K_4 , using only edge splitting and gluing; (Berg and Jordan's theorem);
- (d) E has a proper 2Tree partition.

Figure 60.1.6C shows the construction for a 2-circuit, and an associated 2Tree partition. For 2-circuits with planar graphs, the planar dual is also a 2-circuit. The inductive techniques given above, and others, form dual pairs of constructions for these planar 2-circuits [BCW02].

THEOREM 60.1.14 *Sufficient Connectivity*

If a graph G is 6-connected, then G is generically rigid in the plane.

There are 5-connected graphs that are not generically rigid in the plane.

ALGORITHMS FOR GENERIC 2-RIGIDITY

Each of the combinatorial characterizations has an associated algorithm for verifying whether a graph is generically 2-isostatic:

- (i) Counts: This can be checked by an $O(|V|^2)$ algorithm based on bipartite matchings or network flows on an associated graph [Sug86].
- (ii) 2-construction: Existence of a 2-construction can be checked by an $O(2^{|V|})$ algorithm, but a proposed 2-construction can be verified in $O(|V|)$ time.
- (iii) 3Tree2 covering: Existence can be checked by an $O(|V|^2)$ matroid partition algorithm [Cra].
- (iv) Double tree partition: All required double-tree partitions can be found by a matroidal algorithm of order $O(|V|^3)$.

GENERICALLY RIGID GRAPHS IN HIGHER DIMENSIONS

Most of the results are covered by the initial summary for all dimensions d . Special results apply to the graphs of triangulated polytopes, as well as more general surfaces.

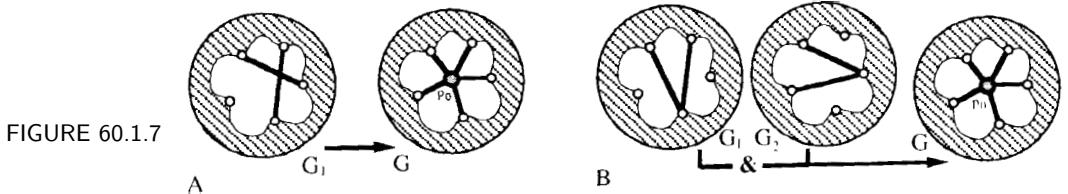
THEOREM 60.1.15 *Triangulated Pseudomanifolds Theorem*

For $d \geq 2$, the graph of a triangulated d -pseudomanifold is generically $(d+1)$ -rigid.

In particular, the graph of any closed triangulated 2-surface without boundary is generically rigid in 3-space (Fogelsanger's theorem), and the graph of any triangulated sphere is generically 3-isostatic (Gluck's theorem). Beyond the triangulated spheres in 3-space, most of these graphs are not isostatic, but are dependent.

OPEN PROBLEMS

There is no combinatorial characterization of generically 3-isostatic graphs. There are several related conjectures, due to Dress, Graver, and Tay and Whiteley, that may be correct but are unproven. We offer one of these.



CONJECTURE 60.1.16 *3-D Replacement Conjecture*

The X -replacement in Figure 60.1.7A takes a graph G_1 that is generically rigid in 3-space to a graph G that is generically rigid in 3-space.

The double V -replacement in Figure 60.1.7B takes two graphs G_1, G_2 that are generically rigid in 3-space to a graph G that is generically rigid in 3-space.

Every 3-isostatic graph is generated by an “extended Henneberg 3-construction,” which adds these two moves to the simpler edge splitting and vertex addition. What is unproven is that *only* 3-isostatic graphs are generated in this way.

The plane analogue of X-replacement is true for plane generic rigidity (without adding the fifth bar) [BCW02], and the 4-space analogue is false for some graphs (with two extra bars added in this analogue). If these conjectured steps prove correct in 3-space, then we would have inductive techniques to generate the graphs of all isostatic frameworks in 3-space, but the algorithm would be exponential.

For 4-space, there is no conjecture that has held up against the known counter-examples based on generically 4-flexible complete bipartite graphs such as $K_{7,7}$.

CONJECTURE 60.1.17 Sufficient Connectivity Conjecture

If a graph G is 12-connected, then G is generically rigid in 3-space.

A graph can be checked for generic 3-rigidity by a “brute force” $O(2^{2^{|V|}})$ algorithm. Assign the points independent variables as coordinates, form the rigidity matrix, then check the rank by symbolic computation. On the other hand, if numerical coordinates are chosen for the points “at random,” then the rank of this numerical matrix ($O(|E|^3)$) will be the generic value, with probability 1. This problem has a randomized polynomial-time algorithm, but there is no known deterministic algorithm that runs in polynomial, or even exponential, time.

60.1.3 GEOMETRY OF FIRST-ORDER RIGIDITY

GLOSSARY

Special position of a graph G in d -space: Any configuration $p \in \mathbb{R}^{dv}$ such that the rigidity matrix $R_G(p)$, or any submatrix, has rank smaller than the maximum rank (the rank at a configuration with algebraically independent coordinates).

Projective transform of a d -configuration p : A d -configuration q on the same vertices, such that there is an invertible matrix T of size $d+1 \times d+1$ making $T(p_i, 1) = \lambda_i(q_i, 1)$ (where $(p_i, 1)$ is the vector p_i extended with an additional 1 — the affine coordinates of p_i).

Affine spanning set for d -space: A configuration p of points such that every point $q_0 \in \mathbb{R}^d$ can be expressed as an affine combination of the p_i : $q_0 = \sum_i \lambda_i p_i$, with $\sum_i \lambda_i = 1$. (Equivalently, the affine coordinates $(p_i, 1)$ span the vector space \mathbb{R}^{d+1} .)

Cone graph: The graph $G * u$ obtained from $G = (V, E)$ by adding a new vertex u and the $|V|$ edges (u, i) for all vertices $i \in V$.

Cone projection from p_0 : For a $(d+1)$ -configuration p on V , a configuration $q = \Pi_0(p)$ in d -space (placed as a hyperplane in $(d+1)$ -space) on the vertices $V \setminus 0$, such that $p_i \neq p_0$ is on the line $q_i p_0$ for all $i \neq 0$.

BASIC RESULTS

THEOREM 60.1.18 First-order Flex Test

If the points of a configuration p on the vertices V affinely span d -space, then a

first-order motion p' is nontrivial if and only if there is some pair h, k (not a bar) such that: $(p_h - p_k) \cdot (p'_h - p'_k) \neq 0$.

THEOREM 60.1.19 Projective Invariance

If a framework $G(p)$ is first-order rigid (isostatic, independent) and $q = T(p)$ is a projective transform of p , then $G(q)$ is first-order rigid (isostatic, independent, respectively).

The following result provides an alternate proof of projective invariance as well as a corresponding generic result for cones.

THEOREM 60.1.20 Coning Theorem

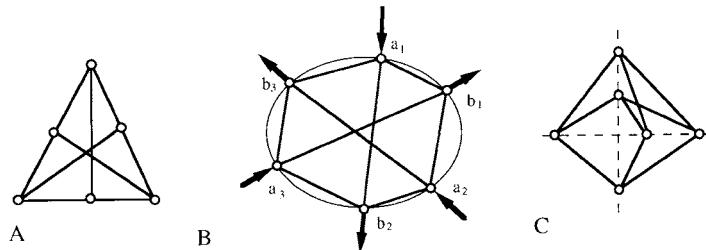
A framework $G(\Pi_0 p)$ is first-order rigid (isostatic, independent) in d -space if and only if the cone $(G * u)(p)$ is first-order rigid (isostatic, independent, respectively) in $(d+1)$ -space.

The special positions of a graph in d -space are rare, since they form a proper algebraic variety (essentially generated by minors of the rigidity matrix with variables for the coordinates of points). For a generically isostatic graph, this set of special positions can be described by the zeros of a single polynomial [WW83].

THEOREM 60.1.21 Pure Condition

For any graph G that is generically isostatic in d -space, there is a homogeneous polynomial $C_G(x_{1,1}, \dots, x_{1,d}, \dots, x_{|V|,1}, \dots, x_{|V|,d})$ such that $G(p)$ is first-order flexible if and only if $C_G(p_1, \dots, p_{|V|}) = 0$. C_G is of degree $(\text{val}_i + 1 - d)$ in the variables $(x_{i,1}, \dots, x_{i,d})$ for each vertex i of valence val_i in the graph.

FIGURE 60.1.8



Since Grassmann algebra (Chapter 59) is the appropriate language for these projective properties, these pure conditions C_G are polynomials in the Grassmann algebra. Section 59.4 contains several examples of these polynomial conditions.

THEOREM 60.1.22 Quadratics for Bipartite Graphs

For a complete bipartite graph $K_{m,n}$ and $d > 1$, the framework $K_{m,n}(p)$, with $p(A)$ and $p(B)$ each affinely spanning d -space, is first-order flexible if and only if all the points $p(A \cup B)$ lie on a quadric surface of d -space (Figure 60.1.8).

The following classical result describes an important open set of configurations that are not special for triangulated spheres.

THEOREM 60.1.23 Extended Cauchy Theorem

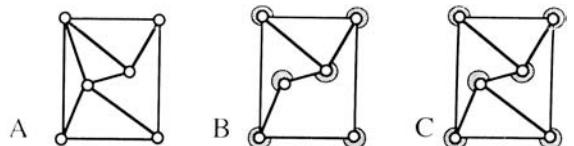
If $G(p)$ consists of the vertices and edges of a convex simplicial d -polytope, then

$G(p)$ is first-order rigid in d -space.

If $G(p)$ consists of the vertices and edges of a strictly convex polyhedron in 3-space, then $G(p)$ is independent.

We recall that Steinitz's theorem guarantees that every 3-connected planar graph has a realization as the edges of a strictly convex polyhedron in 3-space, which gives Gluck's theorem. There are numerous example of nonconvex simplicial polytopes that are not first-order rigid. Connelly [Con78] gives a nonconvex (but not self-intersecting) triangulated sphere (with nine vertices) that is flexible (see the definition below). For many graphs, such as a triangulated torus (Theorem 60.1.15), we do not have even one specific configuration that gives a first-order rigid framework, only the guarantee that “almost all” configurations will work.

FIGURE 60.1.9



Recent papers [Str03, HOR⁺02] suggest that pseudotriangulations play a role for planar graphs in plane rigidity analogous to the role of convex polyhedra for planar graphs in 3-space. **Pseudotriangulations** were defined in Chapter 5, as plane-embedded graphs with a convex polygonal boundary, all interior regions being polygons with exactly three interior angles that are $< \pi$ (Figure 60.1.9A,C). A plane-embedded graph is **pointed** if at each vertex there is an angle that is embedded as $> \pi$ (Figure 60.1.9B,C). The following are some of these recent results.

THEOREM 60.1.24 Counts on Pseudotriangulations

For a general position configuration p , the following properties are equivalent:

- (a) $G(p)$ is a pointed pseudotriangulation;
- (b) $G(p)$ is a pseudotriangulation with $|E| = 2|V| - 3$;
- (c) $G(p)$ is a noncrossing pointed graph with $|E| = 2|V| - 3$;
- (d) $G(p)$ is a noncrossing pointed graph and is maximal with this property, with the given vertices.

THEOREM 60.1.25 Rigidity of Pseudotriangulations

A pseudotriangulation $G(p)$, realized as a bar framework, is first-order rigid. A pointed noncrossing graph $G(p)$ is an independent bar framework.

A planar graph G is generically 2-isostatic if and only if it has a realization as a pointed pseudotriangulation.

There are further significant consequences of the underlying projective geometry of first-order rigidity [CW82]. The concepts of first-order rigidity and first-order flexibility, as well as the dual statics, can be expressed in any of the Cayley-Klein metrics that are extracted from the shared underlying projective space. This family includes the spherical metric, the hyperbolic metric, and others. It is possible to

express first-order rigidity in entirely projective terms that are essentially independent of the metric. In this way, the points “at infinity” in the Euclidean space can be fully integrated into first-order rigidity. However, in some metrics such as the hyperbolic metric, there is a singular set (the sphere at infinity, also known as the *absolute*) on which rigidity equations have distinct properties. This transfer goes back to Pogorelov and has been reworked in [SW02].

THEOREM 60.1.26 *Transfer of Metrics*

For a given graph G and a fixed point p in projective space of dimension d , the framework $G(p)$ is first-order rigid in Euclidean space if and only if $G(p)$ is first-order rigid in any alternate Cayley-Klein metric, with p not containing points on the absolute.

The most extreme projective transformation is a polarity, in which points and hyperplanes (e.g., planes in 3-space) switch roles. For Euclidean 3-space, there are translations of first-order rigidity results to these dual “sheet” structures [Whi87]. For other metrics, the duality in three dimensions changes distance constraints on pairs of points into angle constraints on pairs of planes [SW02].

OTHER RELATED STRUCTURES

A number of related structures have also been investigated for first-order rigidity. One, which appears in engineering, robotics, and chemistry, is the “body-and-hinge framework.” Rigid bodies, indexed by V , are connected in pairs along hinges (lines in 3-space), indexed by edges of a graph. The bodies each move, preserving the contacts at the hinges. Such hinged frameworks could be modeled as bar-and-joint frameworks, with each hinge replaced by a pair of joints and each body replaced by a first-order rigid framework on the joints of its hinges (and other joints if desired); cf. Sections 48.1 and 59.4. Unlike the unsolved problems for generic rigidity of frameworks in 3-space, the generic behavior of body-and-hinge structures has been completely solved. We state two sample results and a related conjecture.

THEOREM 60.1.27 *Tay’s Theorem*

For a graph G the following are equivalent:

- (a) *for some hinge assignment of lines $h_{i,j}$ in 3-space to the edges $\{i,j\}$ of G , the body-and-hinge framework $G(h)$ is first-order rigid;*
- (b) *for almost all hinge assignments h , the body-and-hinge framework $G(h)$ is first-order rigid;*
- (c) *if each edge of the graph is replaced by five copies, the resulting multigraph contains six edge-disjoint spanning trees.*

Tay’s theorem extends directly to all dimensions d (finding $\binom{d+1}{2}$ edge-disjoint spanning trees inside $\binom{d+1}{2} - 1$ copies of the graph).

THEOREM 60.1.28 *Spherical Flexes and Stresses*

Given an abstract spherical structure (see Section 60.3) $S = (V, F; \underline{E})$, and an assignment of distinct points $p_i \in \mathbb{R}^3$ to the vertices, the following two conditions are equivalent:

- (a) the bar framework $G(p)$ on $G = (V, E)$ has a nontrivial self-stress;
- (b) the body-and-hinge framework on the dual graph $G^* = (F, E^*)$ with hinge lines $p_i p_j$ for each edge $\{i, j\}$ of G is first-order flexible.

A second “model” treats the atoms of a molecule as the bodies, and the lines of the bond lines as hinges. Such structures are geometrically singular since the lines of all bonds of an atom are concurrent in the center of the atom. This model, and the equivalent bar frameworks, are central to applications of rigidity to protein structures with thousands of atoms [Whi99].

CONJECTURE 60.1.29 Molecular Conjecture

If a graph G is realized as the atoms (points) and bonds (lines) of a molecular structure, then the molecular structure is generically rigid if, and only if, when each edge of the graph G is replaced by five copies, the resulting multigraph contains six edge-disjoint spanning trees.

This conjecture is embedded in the FIRST algorithm for protein flexibility [JRKT01]. In polar form, the conjecture states that if each body is realized with all hinges of each body coplanar (plate structures), the generic rigidity is still measured by the existence of six spanning trees.

60.1.4 RIGID AND FLEXIBLE FRAMEWORKS

GLOSSARY

Bar equivalence: Two frameworks $G(p)$ and $G(q)$ such that all bars have the same length in both configurations: $|p_i - p_j| = |q_i - q_j|$ for all bars $\{i, j\} \in E$.

Analytic flex: An analytic function $p(t) : [0, 1] \rightarrow \mathbb{R}^{vd}$ such that $G(p(0))$ is bar-equivalent to $G(p(t))$ for all t (i.e., all bars have constant length).

Flexible framework: A bar framework $G(p)$ in \mathbb{R}^d with an analytic flex $p(t)$ such that $p(0) = p$ but p is not congruent to $p(t)$ for all $0 < t$ ([Figure 60.1.1B](#)).

Rigid framework: A bar framework $G(p)$ in d -space that is not flexible ([Figure 60.1.1A,D](#)).

BASIC CONNECTIONS

Because the constraints $|p_i - p_j| = |q_i - q_j|$ are algebraic in the coordinates of the points (after squaring), many alternate definitions of a “rigid framework” are equivalent. These connections depend on results such as the curve selection theorem of algebraic geometry or the inverse function theorem.

THEOREM 60.1.30 Alternate Rigidity Definitions

For a bar framework $G(p)$ the following conditions are equivalent:

- (a) the framework is rigid;

- (b) for every continuous path, or **continuous flex** of $G(p)$, $p(t) \in \mathbb{R}^{vd}$, $0 \leq t < 1$ and $p(0) = p$, such that $G(p(t))$ is bar-equivalent to $G(p)$ for all t , $p(t)$ is congruent to p for all t ;
- (c) there is an $\epsilon > 0$ such that if $G(p)$ and $G(q)$ are bar-equivalent and $|p - q| < \epsilon$, then p is congruent to q .

Essentially, the first derivative of a nontrivial analytic flex is a nontrivial first-order flex: $D_t((p_i(t) - p_j(t))^2 = c_{ij})|_{t=0} \Rightarrow 2(p_i - p_j) \cdot (p'_i - p'_j) = 0$. (If this first derivative is trivial, then the earliest nontrivial derivative is a first-order motion.) This result is related to general forms of the inverse function theorem.

THEOREM 60.1.31 First-order Rigid to Rigid

If a bar framework $G(p)$ is first-order rigid, then $G(p)$ is rigid.

Some first-order flexes are not the derivatives of analytic flexes (Figures 60.1.1D and 60.1.2B). However, a nontrivial first-order flex for a framework does guarantee a pair of nearby noncongruent, bar-equivalent frameworks.

THEOREM 60.1.32 Averaging Theorem

If the points of a configuration p affinely span d -space, then the assignment p' is a nontrivial first-order flex of $G(p)$ if and only if the frameworks $G(p + p')$ and $G(p - p')$ are bar-equivalent and not congruent.

Rigidity and first-order rigidity are equivalent in some situations.

THEOREM 60.1.33 Rigid to First-order Rigid

If bar framework $G(p)$ is independent, then $G(p)$ is first-order rigid if and only if $G(p)$ is rigid.

The recent solution of the Carpenter’s Rule problem on straightening plane-embedded polygonal paths and convexifying plane-embedded polygons [CDR03, Str03] uses independence of appropriate bar frameworks, and resulting paths. The independence is proven using Maxwell’s theorem (see Section 60.3). See Chapter 9 for more connections. The following is one form of this connection [RSS03].

THEOREM 60.1.34 Expansive Motions

If one edge of the boundary polygon of a pointed pseudotriangulation $G(p)$ is removed, and its two vertices are spread apart in a motion, then the resulting path (unique up to congruences) is expansive—all pairs of joints are either moving apart or remaining at a constant distance.

Whereas first-order rigidity is projectively invariant, rigidity itself is not projectively invariant—or even affinely invariant. It is a purely Euclidean property.

THEOREM 60.1.35 Generic Rigidity Theorem II

For a graph G and a fixed dimension d the following are equivalent:

- (a) G is generically rigid in d -space;
- (b) for all $q \in U \subset \mathbb{R}^{dv}$, U some nonempty open set, $G(q)$ is rigid;
- (c) for all $q \in W \subset \mathbb{R}^{dv}$, W some open dense set, $G(q)$ is first-order rigid.

60.1.5 TENSEGRITY FRAMEWORKS

In a tensegrity framework, we replace some (or all) of the equalities for bars with inequalities for the distances—corresponding to *cables* (the distance can shrink but not expand) and *struts* (the distance can expand but not shrink). The study of these inequalities introduces techniques from linear programming.

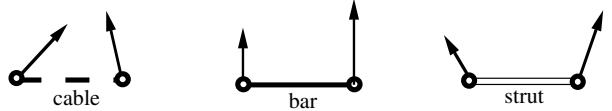
GLOSSARY

Signed graph: A graph with a partition of the edges into three classes, written $G_{\pm} = (V; E_-, E_0, E_+)$.

Tensegrity framework $G_{\pm}(p)$ in \mathbb{R}^d : A signed graph $G_{\pm} = (V; E_-, E_0, E_+)$ and a configuration p on V .

Cables, bars, struts: For a tensegrity framework $G_{\pm}(p)$, the members of E_- , of E_0 , and of E_+ , respectively. In figures, cables are indicated by dashed lines, struts by double thin lines, and bars by single thick lines (see Figure 60.1.10).

FIGURE 60.1.10



$G_{\pm}(p)$ **dominates** $G_{\pm}(q)$: For each edge, the appropriate condition holds:

$$\begin{aligned} |p_i - p_j| &\geq |q_i - q_j| && \text{when } \{i, j\} \in E_- \\ |p_i - p_j| &= |q_i - q_j| && \text{when } \{i, j\} \in E_0 \\ |p_i - p_j| &\leq |q_i - q_j| && \text{when } \{i, j\} \in E_+. \end{aligned}$$

Rigid tensegrity framework $G_{\pm}(p)$: For every analytic path $p(t)$ in \mathbb{R}^{vd} , $0 \leq t < 1$, if $p(0) = p$ and $G(p)$ dominates $G(p(t))$ for all t , then p is congruent to $p(t)$ for all t .

First-order flex of a tensegrity framework G_{\pm} : An assignment $p' : V \rightarrow \mathbb{R}^d$ of velocities to the vertices such that, for each edge $\{i, j\} \in E$ (Figure 60.1.10),

$$\begin{aligned} (p_j - p_i) \cdot (p'_j - p'_i) &\leq 0 && \text{for cables } \{i, j\} \in E_- \\ (p_j - p_i) \cdot (p'_j - p'_i) &= 0 && \text{for bars } \{i, j\} \in E_0 \\ (p_j - p_i) \cdot (p'_j - p'_i) &\geq 0 && \text{for struts } \{i, j\} \in E_+. \end{aligned}$$

Trivial first-order flex: A first-order flex p' of a tensegrity framework $G_{\pm}(p)$ such that $p'_i = Sp_i + t$ for all vertices i , with a fixed skew-symmetric matrix S and vector t .

First-order rigid: A tensegrity framework $G_{\pm}(p)$ is first-order rigid if every first-order flex is trivial, and **first-order flexible** otherwise.

Proper self-stress on a tensegrity framework $G_{\pm}(p)$: An assignment ω of scalars to the edges of G such that:

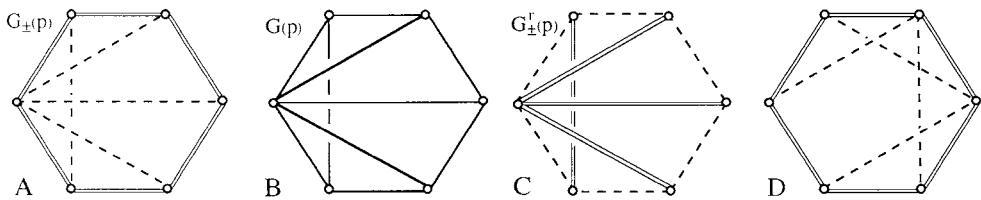
- (a) $\omega_{ij} \geq 0$ for cables $\{i, j\} \in E_-$;

- (b) $\omega_{ij} \leq 0$ for struts $\{i, j\} \in E_+$; and
- (c) for each vertex i , $\sum_{\{j \mid \{i,j\} \in E\}} \omega_{ij}(p_j - p_i) = 0$.

Strict self-stress: A proper self-stress ω with the inequalities in (a) and (b) strict.

Underlying bar framework: For a tensegrity framework $G_{\pm}(p)$, the bar framework $G(p)$ on the unsigned graph $G = (V, E)$, where $E = E_- \cup E_0 \cup E_+$ (Figure 60.1.11A,B).

FIGURE 60.1.11



BASIC RESULTS

The equivalent definitions of “rigidity” and the basic connections between rigidity and first-order rigidity all transfer directly to tensegrity frameworks [RW81].

THEOREM 60.1.36 First-order Stress Test

A tensegrity framework $G_{\pm}(p)$ is first-order rigid if and only if the underlying bar framework $G(p)$ is first-order rigid and there is a strict self-stress on $G_{\pm}(p)$ (Figure 60.1.11A,B).

This connection to self-stresses means that any first-order rigid tensegrity framework with at least one cable or strut has $|E| > d|V| - \binom{d+1}{2}$ edges.

THEOREM 60.1.37 Reversal Theorem

A tensegrity framework $G_{\pm}(p)$ is first-order rigid if and only if the reversed framework $G_{\pm}^r(p)$ is first-order rigid, where the graph G_{\pm}^r interchanges cables and struts (Figure 60.1.11A,C).

There is no single “generic” behavior for a signed graph G_{\pm} . If some configuration produces a first-order rigid framework for a graph G_{\pm} , then the set of all such configurations is open but not dense. The algebraic variety of “special positions” of the underlying unsigned graph divides the configuration space into open subsets, in some of which all configurations are rigid, and in others, none are. The required sign pattern for a self-stress can change as you cross such a boundary [WW83].

The first-order rigidity of a tensegrity framework is projectively invariant, with the proviso that a cable (strut) $\{i, j\}$ is switched to a strut (cable) whenever $\lambda_i \lambda_j < 0$ for the projective transformation.

THEOREM 60.1.38 Stress Existence

If a tensegrity framework $G_{\pm}(p)$ with at least one cable or strut is rigid, then there is a nonzero proper self-stress.

A number of results relate minima of quadratic energy functions to the rigidity of tensegrity frameworks. These energy results are not invariant under projective transformations, but such rigidity is preserved under “small” affine transformations. This is one result, drawn from results on second-order rigidity [CW96].

THEOREM 60.1.39 Rigidity Stress Test

A tensegrity framework $G_{\pm}(p)$ is rigid if, for each nontrivial first-order motion p' of $G_{\pm}(p)$, there is a proper self-stress $\omega^{p'}$ making $\sum_{ij} \omega_{ij}^{p'} (p'_i - p'_j) \cdot (p'_i - p'_j) > 0$.

A special result for modified frameworks—with some vertices fixed or pinned—further illustrates the role of tensegrity frameworks. A **spiderweb** is a partitioned graph $G_- = (V_0, V_1, E_-)$, with pinned vertices V_0 , with $E_- \setminus V_1 \times [V_0 \cup V_1]$ and a configuration p for $V_0 \cup V_1$. A **spiderweb self-stress** for $G_-(p)$ is an assignment ω of nonnegative scalars to E_- such that for each unpinned vertex $i \in V_1$, $\sum_{\{j \mid \{i,j\} \in E_-\}} \omega_{ij} (p_j - p_i) = 0$. A **spiderweb flex** for $G_-(p)$ is a flex $p(t)$ of the induced tensegrity framework on the spiderweb, with all pinned vertices fixed ($p_k(t) = p_k$) (Figure 60.1.12).

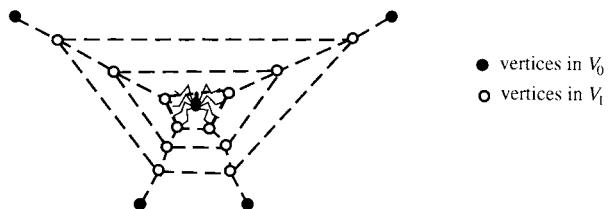


FIGURE 60.1.12

THEOREM 60.1.40 Spiderweb Rigidity

Any spiderweb $G_-(p)$ in d -space with a spiderweb self-stress, positive on all cables, is rigid in d -space.

All critical points of functions of squared edge lengths correspond to proper self-stresses of a tensegrity framework, with members E_- for positive coefficients and E_+ for negative coefficients in the energy function. As a corollary, graph drawing programs (Chapter 52) that use minima (or critical points) of such energy functions will generate polyhedral pictures for planar graphs.

In the spiderweb energies, there is a global minimum of energy. This means that the configuration is globally rigid—no other realizations have the same edge lengths. In general, global rigidity has a distinct theory with some specific overlaps to the theory presented here.

Related to sphere packings (Chapter 61) are “reversed spiderwebs”: tensegrity frameworks with vertices at the centers of the spheres (fixed joints for external pressures or constraints) and struts when two spheres contact. Such strut frameworks are rigid (corresponding to locally maximal density of the packing) if and only if they are first-order rigid (again with vertices in V_0 fixed) [Con88].

60.2 SCENE ANALYSIS

The problem of reconstructing spatial objects (polyhedra or polyhedral surfaces) from a single plane picture is basic to several applications. This section summarizes the combinatorial results for “generic pictures” (Section 60.2.1). Section 60.2.2 presents a polar “parallel configurations” interpretation of the same abstract mathematics and Section 60.2.3 presents connections to other fields of discrete geometry.

60.2.1 COMBINATORICS OF PLANE POLYHEDRAL PICTURES

GLOSSARY

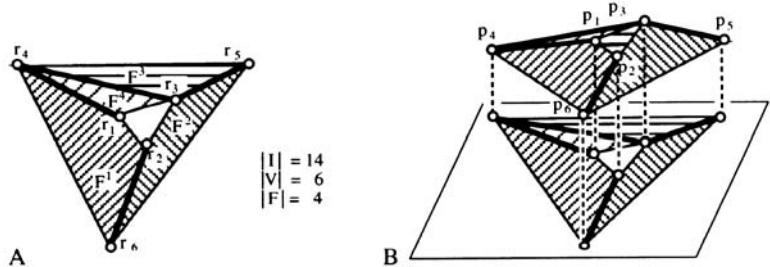
Polyhedral incidence structure S : An abstract set of *vertices* V , an abstract set of *faces* F and a set of *incidences* $I \subset V \times F$.

d -scene for an incidence structure $S = (V, F; I)$: A pair of location maps, $p : V \rightarrow \mathbb{R}^d$, $p_i = (x_i, \dots, z_i, w_i)$ and $P : F \rightarrow \mathbb{R}^d$, $P^j = (A^j, \dots, C^j, D^j)$, such that, for each $(i, j) \in I$: $A^j x_i + \dots + C^j z_i + w_i + D^j = 0$. (We assume that no hyperplane is vertical, i.e., is parallel to the vector $(0, 0, \dots, 0, 1)$.)

$(d-1)$ -picture of an incidence structure S : A location map $r : V \rightarrow \mathbb{R}^{d-1}$, $r_i = (x_i, \dots, z_i)$ (Figure 60.2.1A).

Lifting of a $(d-1)$ -picture $S(r)$: A d -scene $S(p, P)$ with vertical projection $\Pi(p) = r$ (Figure 60.2.1B). (I.e., if $p_i = (x_i, \dots, z_i, w_i)$, then $r_i = (x_i, \dots, z_i) = \Pi(p_i)$).

FIGURE 60.2.1



Lifting matrix for a picture $S(r)$: The $|I| \times (|V| + d|F|)$ coefficient matrix $M_S(r)$ of the system of equations for liftings of a picture $S(r)$: for each $(i, j) \in I$, $A^j x_i + \dots + C^j z_i + w_i + D^j = 0$, where the variables are ordered:

$$\dots, w_i, \dots ; \dots, A^j, \dots, C^j, D^j, \dots$$

Sharp picture: A $(d-1)$ -picture $S(r)$ that has a lifting $S(p, P)$ with a distinct hyperplane for each face (Figure 60.2.1A,B).

BASIC RESULTS

Since the incidence equations are linear, there is no distinction between “continuous liftings” and “first-order liftings.” Since the rank of the lifting matrix is determined

by a polynomial process on the entries, “generic properties” of pictures have several characterizations.

THEOREM 60.2.1 *Generic Pictures*

For a structure S and a dimension d , the following are equivalent:

- (a) the structure is generically sharp in d -space (an open dense subset of configurations r produce sharp d -pictures);
 - (b) $S(r)$ is sharp for a configuration r with algebraically independent coordinates.

The generic properties of a structure are robust: all small changes in such a sharp picture are also sharp pictures and small changes in the points of a sharp picture require only small changes in the sharp lifting. Even special positions of such structures will always have nontrivial liftings, although these may not be sharp. However, up to numerical round-off, all pictures “are generic.” Other structures that are not generically sharp (Figure 60.2.2A) may have sharp pictures in special positions (Figure 60.2.2B), but a small change in the position of even one point can destroy this sharpness.

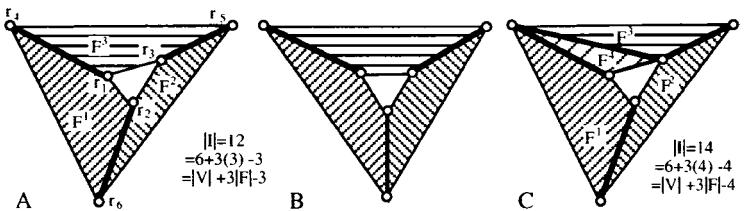


FIGURE 60.2.2

The incidence equations allow certain “trivial” changes to a lifted scene that will preserve the picture—generated by adding a single plane H^0 to all existing planes: $P_*^j = H^0 + P^j$; and by changes in vertical scale in the scene: $w_i^* = \lambda w_i$. This space of *lifting equivalences* has dimension $d+1$, provided the points of the scene do not lie in a single hyperplane.

THEOREM 60.2.2 *Picture Theorem*

A generic picture of an incidence structure $S = (V, F; I)$ with at least two faces has a sharp lifting, unique up to lifting equivalence, if and only if $|I| = |V| + d|F| - (d+1)$ and, for all subsets I' of incidences on at least two faces, $|I'| \leq |V'| + d|F'| - (d+1)$ (Figure 60.2.1A,C).

A generic picture of an incidence structure $S = (V, F; I)$ has independent rows in the lifting matrix if and only if for all nonempty subsets I' of incidences, $|I'| \leq |V'| + d|F'| - d$ (Figure 60.2.2A).

ALGORITHMS

Any part of a structure with $|I'| = |V'| + d|F'| - d$ independent incidences will be forced to be coplanar over a picture with algebraically independent coordinates for the points. If the structure is not generically sharp, then an effective, robust lifting algorithm consists of selecting a subset of vertices for which the incidences

are sharp, then “correcting” the position of the other vertices based on calculations in the resulting scene. This requires effective algorithms for selecting such a set of incidences. Sugihara and Imai have implemented $O(|I|^2)$ algorithms for finding generically sharp (independent) structures using modified bipartite matching on the incidence structure [Sug86].

60.2.2 PARALLEL DRAWINGS

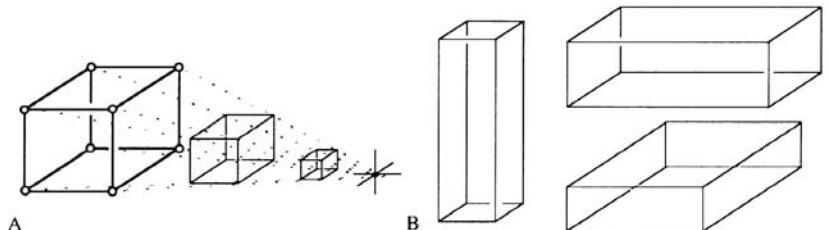
The mathematical structure defined for polyhedral pictures has another, dual interpretation: the polar of a “point constrained by one projection” is a “hyperplane constrained by an assigned normal.” Two configurations sharing the prescribed normals are “parallel drawings” of one another. These geometric patterns, used by engineering draftsmen in the nineteenth century, have reappeared in a number of branches of discrete geometry. This dual interpretation also establishes a basic connection between the geometry and combinatorics of scene analysis and the geometry and combinatorics of first-order rigidity of frameworks.

GLOSSARY

Parallel d -scenes for an incidence structure: Two d -scenes $S(p, P), S(q, Q)$ such that for each face j , $P^j \parallel Q^j$ (that is, the first $d - 1$ coordinates are equal) (Figure 60.2.3). (For convenience, not necessity, we stick with the “nonvertical” scenes of the previous section.)

Nontrivially parallel d -scene for a d -scene $S(p, P)$: A parallel d -scene $S(q, Q)$, such that the configuration q is not a translation or dilatation of the configuration p (Figure 60.2.3B for $d = 2$).

FIGURE 60.2.3



Directions for the faces: An assignment of d -vectors $D^j = (A^j, \dots, C^j)$ to $j \in F$.

d -scene realizing directions D : A d -scene $S(p, P)$ such that for each face $j \in F$, the first $d - 1$ coordinates of P^j and D^j coincide.

Parallel drawing matrix for directions D in d -space: The $|I| \times (|V| + d|F|)$ matrix $M_S(D)$ for the system of equations for each incidence $(i, j) \in I$: $A^j x_i + B^j y_i + \dots + C^j z_i + w_i + D^j = 0$, where the variables are ordered:

$$\dots, D^j, \dots ; \dots, x_i, y_i, \dots, z_i, w_i, \dots$$

BASIC RESULTS

All results for polyhedral pictures dualize to parallel drawings. Again, for parallel drawings there is no distinction between continuous changes and first-order changes. The trivially parallel drawings, generated by d translations and one dilatation towards a point, form a vector space of dimension $d + 1$, provided there are at least two distinct points (Figure 60.2.3A). (A trivially parallel drawing may even have all points coincident, though the faces will still have assigned directions (Figure 60.2.3A).)

THEOREM 60.2.3 *Parallel Drawing Theorem*

For generic selections of the directions D in d -space for the faces, a structure $S = (V, F; I)$ has a realization $S(p, P)$ with all points p distinct if and only if, for every nonempty set I' of incidences involving at least two points $V(I')$ and faces $F(I')$, $|I'| \leq d|V(I')| + |F(I')| - (d + 1)$ (Figure 60.2.3A).

In particular, a configuration p, P with distinct points realizing generic directions for the incidence structure is unique, up to translation and dilatation, if and only if $|I| = d|V| + |F| - (d + 1)$ and $|I'| \leq d|V'| + |F'| - (d + 1)$.

Of course other nontrivially parallel drawings will also occur if the rank is smaller than $d|V'| + |F'| - (d + 1)$ (Figure 60.2.3 B, with a generic rank 1 less than required for $d = 2$, and a geometric rank, as drawn, 2 less than required).

Figure 60.2.3 may also be interpreted as the parallel drawings of a “cube in 3-space.” For spherical polyhedra, there is an isomorphism between the nontrivially parallel drawings in 3-space (the parallel drawings modulo the trivial drawings) and the nontrivially parallel drawings in a plane projection [CW94]. Only the dimension (4 vs. 3) of the trivially parallel drawings will change with the projection.

60.2.3 CONNECTIONS TO OTHER FIELDS

FIRST-ORDER RIGIDITY

For any plane framework, if we turn the vectors of a first-order motion 90° (say clockwise), they become the vectors joining p to a parallel drawing q of the framework (Figure 60.2.4A,B). The converse is also true.

THEOREM 60.2.4

A plane framework $G(p)$ has a nontrivial first-order flex if and only if the configuration $G(p)$ has a nontrivially parallel drawing $G(q)$ (Figure 60.2.4C,D).

Because of this connection, combinatorial and geometric results for plane first-order rigidity and for plane parallel drawings have numerous deep connections. For example, Laman’s theorem (Theorem 60.1.12b) is a corollary of the parallel drawing theorem, for $d = 2$. In higher dimensions, the connection is one-way: a nontrivially parallel drawing of a “framework” (the “direction of an edge” is represented by $d - 1$ facets through the two points) induces one (or more) nontrivial first-order motions of the corresponding bar framework. The theory of parallel drawing in higher

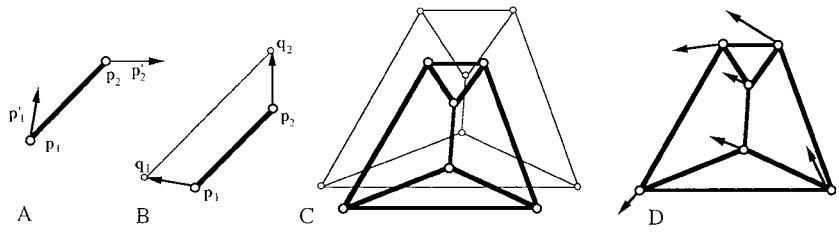


FIGURE 60.2.4

dimensions is more complete and has simpler algorithms than the theory of first-order rigidity in higher dimensions, generalizing almost all results for plane first-order rigidity and plane parallel drawings, including combinatorial characterizations using counts, tree partitions, and inductive constructions of maximimal independent sets.

MINKOWSKI DECOMPOSABILITY

By a theorem of Shephard, a polytope is decomposable as the Minkowski sum of two simpler polyhedra if and only if the faces and vertices of the polytope (or the edges and vertices of the polytope) have a nontrivially parallel drawing. Many characterizations of Minkowski indecomposable polytopes can be deduced directly from results for parallel d -scenes (or equivalently, for polyhedral pictures of the polar polytope).

ANGLES IN CAD

In plane computer-aided design, many different patterns of constraints (lengths, angles, incidences of points and lines, etc.) are used to design or describe configurations of points and lines, up to congruence or local congruence. With distances between points, the geometry becomes that of first-order rigidity. If angles and incidences are added, even the problems of “generic rigidity” of constraints are unsolved (and perhaps not solvable in polynomial time). However, special designs, mixing lengths, distances of points to lines, and trees of angles have been solved, using direct extensions of the techniques and results for plane frameworks and plane parallel drawings [SW99].

There is another connection between angles of intersections and rigidity. A recent manuscript [SW02] describes a correspondence between the first-order theory of circles of variable radius and intersection angles as constraints and distance constraints between points in Euclidean (and hyperbolic) 3-space, as well as spheres and angles in 3-space and points and distances in 4-space. As a result, the full complexity of distance constraints in 4-space is embedded inside general dimensioning in 3-space CAD. In general, geometric systems of constraints do not yield simple combinatorial counting algorithms of the type found for plane first-order rigidity.

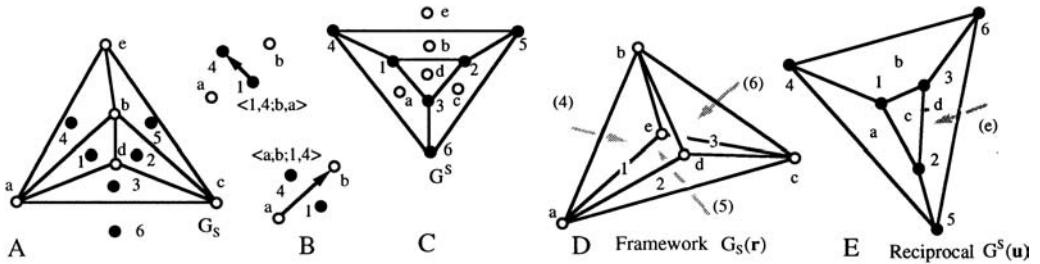
60.3 RECIPROCAL DIAGRAMS

The reciprocal diagram is a single geometric construction that has appeared, independently over a 140-year span, in areas such as “graphical statics” (drafting techniques for resolving forces), scene analysis, and computational geometry.

GLOSSARY

Abstract spherical polyhedron $S = (V, F; \underline{E})$: For a 2-connected planar graph $G_S = (V, E_S)$, drawn without self-intersection on a sphere (or in the plane), we record the vertices as V and the regions as faces F , and rewrite the directed edges \underline{E} as ordered 4-tuples $\underline{e} = \langle h, i; j, k \rangle$, where the edge from vertex h to vertex i has face j on the right and face k on the left. (The reversed edge $-\underline{e} = \langle i, h; k, j \rangle$ runs from i to h , with k on the right.)

FIGURE 60.3.1



Dual abstract spherical polyhedron: The abstract spherical polyhedron S^* formed by switching the roles of V and F , and switching the pairs of indices in each ordered edge $\underline{e} = \langle h, i; j, k \rangle$ into $\underline{e}^* = \langle j, k; i, h \rangle$. (Also the abstract spherical polyhedron formed by the dual planar graph $G^S = (F, E^S)$ of the original planar graph (Figure 60.3.1A,C).)

Proper spatial spherical polyhedron: An assignment of points $p_i = (x_i, y_i, z_i)$ to the vertices and planes $P^j = (A^j, B^j, D^j)$ to the faces of an abstract spherical polyhedron $(V, F; \underline{E})$, such that if vertex i and face j share an edge, then the point lies on the plane: $A^j x_i + B^j y_i + z_i + D^j = 0$; and at each edge the two vertices are distinct points and the two faces have distinct planes.

Projection of a proper spatial polyhedron $S(p, P)$: The plane framework $G_S(r)$, where r is the vertical projection of the points p (i.e., $r_i = \Pi p_i = (x_i, y_i)$) (Figure 60.3.2).

Gradient diagram of a proper spatial polyhedron $S(p, P)$: The plane framework $G^S(s)$, where $s_j = (A^j, B^j)$ is (minus) the gradient of the plane P^j (Figure 60.3.2).

Reciprocal diagrams: For an abstract spherical polyhedron S , two frameworks $G_S(r)$ and $G^S(s)$ on the graph and the dual graph of the polyhedron, such that for each directed edge $\langle h, i; j, k \rangle \in \underline{E}$, $(r_h - r_i) \cdot (s_j - s_k) = 0$ (Figure 60.3.1D,E).

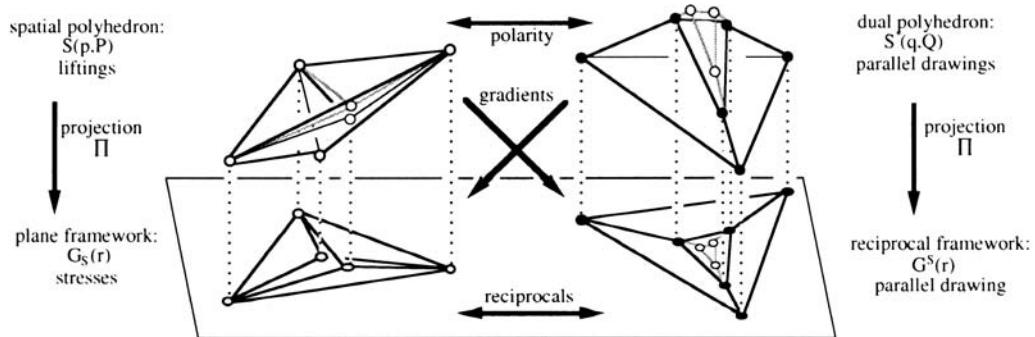
BASIC RESULTS

Reciprocal diagrams have deep connections to both of our previous topics:

- Given a spatial scene on a spherical structure, with no faces vertical, the vertical projection and the gradient diagram are reciprocal diagrams. (This follows because the difference of the gradients at an edge is a vector perpendicular to the vertical plane through the edge.)
- Given a pair of reciprocal diagrams on $S = (V, F; E)$, then for each edge $e = \langle h, i; j, k \rangle$ the scalars ω_{ij} defined by $\omega(r_h - r_i) = (s_j - s_k)^\perp$ (where $^\perp$ means rotate by 90° clockwise) form a self-stress on the framework $G_S(r)$. (This follows because the closed polygon of a face in $G^S(s)$ is, after $^\perp$, the vector sum for the “vertex equilibrium” in the self-stress condition.)

These facts can be extended to other oriented polyhedra and their projections. The real surprise is that, for spherical polyhedra, the converses hold and all these concepts are equivalent (an observation dating back to Clerk Maxwell and the drafting techniques of graphical statics).

FIGURE 60.3.2



THEOREM 60.3.1 *Maxwell's Theorem*

For an abstract spherical polyhedron $(V, F; E)$, the following are equivalent:

- The framework $G_S(r)$, with the vertices of each edge distinct, has a self-stress nonzero on all edges;
- $G_S(r)$ has a reciprocal framework $G^S(s)$ with the vertices of each edge distinct;
- $G_S(r)$ is the vertical projection of a proper spatial polyhedron $S(p, P)$;
- $G_S(r)$ is the gradient diagram of a proper spatial polyhedron $S^*(q, Q)$.

There are other refinements of this theorem, that connect the space of self-stresses of $G_S(r)$ with the space of parallel drawings (and first-order flexes) of $G^S(s)$, the space of polyhedra $S(p, P)$ with the same projection, and the space of

parallel drawings of $S^*(q, Q)$ [CW94] (Figure 60.3.2). A second refinement connects the local convexity of the edge of the polyhedron with the sign of the self-stress.

THEOREM 60.3.2 Convex Self-stress

The vertical projection of a strictly convex polyhedron, with no faces vertical, produces a plane framework with a self-stress that is < 0 on the boundary edges (the edges bounding the infinite region of the plane) and > 0 on all edges interior to this boundary polygon.

A plane Delaunay triangulation also has a basic “reciprocal” relationship to the plane Voronoi diagram: the edges joining vertices at the centers of the regions are perpendicular to edges of the polygon of the Voronoi regions surrounding the vertex. This pair of reciprocals is directly related to the projection of a spatial convex polyhedral cap, as are generalized Voronoi diagrams. See Section 23.1.

This pattern of “reciprocal constructions” and the connection to liftings to polytopes in the next dimension generalizes to higher dimensions [CW94]. For example, for Voronoi diagrams and Delaunay simplicial complexes, the edges of one are perpendicular to facets of the other, in all dimensions. Moreover, for appropriate sphere-like homology, the existence of a reciprocal corresponds to the existence of nontrivial liftings [CW94, ERR01, Ryb99]. Such geometric structures are also related to k -rigidity and to combinatorial proofs of the g -theorem in polyhedral combinatorics [TW00]. Finally, [BGH02] makes a related connection between parallel drawings and group actions on complex manifolds.

60.4 SOURCES AND RELATED MATERIALS

SURVEYS AND BASIC SOURCES

All results not given an explicit reference can be traced through these surveys:

[CW96]: A presentation of basic results for concepts of rigidity between first-order rigidity and rigidity for tensegrity frameworks.

[CW]: A thorough introduction to a number of topics on the rigidity of frameworks, in manuscript form only.

[GSS93]: A monograph devoted to combinatorial results for the graphs of generically rigid frameworks, with an extensive bibliography on many aspects of rigidity.

[Ros00]: A recent thesis that explores in depth both topics of this chapter and their connections.

[Sug86]: A monograph on the reconstruction of spatial polyhedral objects from plane pictures.

[Whi93]: A survey of results relating first-order rigidity to matroid theory and related matroids for scene analysis, and to multivariate splines.

[Whi96]: An expository article presenting matroidal aspects of first-order rigidity, scene analysis, and multivariate splines.

RELATED CHAPTERS

- Chapter 9: Geometry and topology of polygonal linkages
 - Chapter 18: Face numbers of polytopes and complexes
 - Chapter 23: Voronoi diagrams and Delaunay triangulations
 - Chapter 29: Geometric reconstruction problems
 - Chapter 48: Robotics
 - Chapter 52: Graph drawing
 - Chapter 59: Geometric applications of the Grassmann-Cayley algebra
 - Chapter 61: Sphere packing and coding theory
-

REFERENCES

- [BCW02] L. Berenbrink, L. Chavez, and W. Whiteley. Inductive constructions for 2-rigidity: bases and circuits via tree partitions. Manuscript, York University, Toronto, 2002.
- [BJ03] A. Berg and T. Jordán. A proof of Connelly's conjecture on 3-connected circuits of the rigidity matroid. *J. Combinatorial Theory Ser. B.*, 88:77–97, 2003.
- [BGH02] E. Bolker, V. Guillemin, and T. Holmes. How is a graph like a manifold? Preprint, MIT, Cambridge, 2002.
- [Con78] R. Connelly. A flexible sphere. *Math. Intelligencer*, 1:130–131, 1978.
- [Con82] R. Connelly. Rigidity and energy. *Invent. Math.*, 66:11–33, 1982.
- [Con88] R. Connelly. Rigidity and sphere packing I, II. *Structural Topology*, 14:43–60, 1988 and 16:59–75, 1990.
- [CW96] R. Connelly and W. Whiteley. Second-order rigidity and pre-stress stability for tensegrity frameworks. *SIAM J. Discrete Math.*, 9:453–492, 1996.
- [CDR03] R. Connelly, E. Demaine, and G. Rote. Straightening polygonal arcs and convexifying polygons. *Discrete Comput. Geom.*, 30:205–239, 2003.
- [Cra] H. Crapo. On the generic rigidity of structures in the plane. *Adv. in Appl. Math.*, to appear.
- [CW82] H. Crapo and W. Whiteley. Statics of frameworks and motions of panel structures: a projective geometric introduction. *Structural Topology* 6:43–82, 1982.
- [CW94] H. Crapo and W. Whiteley. Spaces of stresses, projections and parallel drawings for spherical polyhedra. *Contrib. Alg. Geom.*, 35:259–281, 1994.
- [CW] H. Crapo and W. Whiteley, editors. *The Geometry of Rigid Structures*. Draft manuscript chapters, York University, Toronto.
- [ERR01] R. Erdahl, K. Rybníkov, and S. Ryškov. On traces of d -stresses in skeletons of lower dimensions of homology d -manifolds. *European J. Combin.*, 22:801–820, 2001.
- [GSS93] J. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*. Number 2 of *AMS Monographs*. Amer. Math. Soc., Providence, 1993.
- [HOR⁺02] R. Haas, D. Ogdan, G. Rote, F. Santos, B. Servatius, H. Servatius, D. Souvaine, I. Streinu, and W. Whiteley. Planar minimally rigid graphs have pseudo-triangular embeddings. Manuscript, 2002.
- [JRKT01] D. Jacobs, A.J. Rader, L. Kuhn, and M. Thorpe. Protein flexibility predictions using graph theory. *Proteins*, 44:150–165, 2001.

- [Ros00] L. Ros. *A Kinematic-Geometric Approach to Spatial Interpretation of Line Drawings*. PhD thesis. Technical University of Catalonia, 2000. Available at <http://www-iri.upc.es/people/ros>.
- [RSS03] G. Rote, F. Santos, and I. Streinu. Expansive motions and the polytope of pointed pseudo-triangulations. In B. Aronov, S. Basu, J. Pach, and M. Sharir, editors, *Discrete and Computational Geometry—The Goodman-Pollack Festschrift, Algorithms Combin.*, pages 699–736. Springer-Verlag, Berlin, 2003.
- [RW81] B. Roth and W. Whiteley. Tensegrity frameworks. *Trans. Amer. Math. Soc.*, 265:419–446, 1981.
- [Ryb99] K. Rybníkov. Lifting and stresses of cell complexes. *Discrete Comput. Geom.*, 21:481–517, 1999.
- [SW02] F. Saliola and W. Whiteley. Rigidity of frameworks: Euclidean, spherical, hyperbolic, and projective. Preprint, York Univ., Toronto, 2002.
- [SW99] B. Servatius and W. Whiteley. Constraining plane configurations in CAD: combinatorics of directions and lengths. *SIAM J. Discrete Math.*, 12:136–153, 1999.
- [Str03] I. Streinu. Combinatorial roadmaps in configuration spaces of simple planar polygons. In S. Basu and L. Gonzalez-Vega, editors, *Algorithmic and Quantitative Real Algebraic Geometry*, volume 60 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 181–205. Amer. Math. Soc., Providence, 2003.
- [Sug86] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, Cambridge, 1986.
- [TW00] T-S. Tay and W. Whiteley. A homological approach to skeletal rigidity. *Adv. in Appl. Math.*, 25:102–151, 2000.
- [WW83] N. White and W. Whiteley. Algebraic geometry of stresses in frameworks. *SIAM J. Alg. Disc. Meth.*, 4:53–70, 1983.
- [Whi84] W. Whiteley. Infinitesimally rigid polyhedra I: statics of frameworks. *Trans. Amer. Math. Soc.*, 285:431–465, 1984.
- [Whi87] W. Whiteley. Rigidity and polarity I: statics of sheetworks. *Geom. Dedicata*, 22:329–362, 1987.
- [Whi93] W. Whiteley. Matroids and rigidity. In Neil White, editor, *Matroid Applications*, pages 1–53. Cambridge Univ. Press, 1993.
- [Whi94] W. Whiteley. How to describe or design a polyhedron. *J. Intell. Robotic Syst.*, 11:135–160, 1994.
- [Whi96] W. Whiteley. Some matroids from discrete applied geometry. In J. Bonin, J. Oxley, and B. Servatius, editors, *Matroid Theory*, volume 197 of *Contemp. Math.*, pages 171–311. Amer. Math. Soc., Providence, 1996.
- [Whi99] W. Whiteley. Rigidity of molecular structures: generic and geometric analysis. In P.M. Duxbury and M.F. Thorpe, editors, *Rigidity Theory and Applications*, Kluwer/Plenum, New York, 1999.

61 SPHERE PACKING AND CODING THEORY

G.A. Kabatiansky and J.A. Rush

INTRODUCTION

Consider a metric space equipped with some measure (natural in all examples) in which all balls of the same radius have the same “volume” (measure). A set of metric balls of the same radius is called a sphere packing if the intersection of any two balls has measure zero. In the case where the space has finite measure the density of a sphere packing is defined as the ratio of the measure of the union of the balls to the measure of the whole space. In the other case one can consider some natural “large subspace” of the space, such as a ball of large radius, and define the density of a sphere packing as the limit of the ratio of the corresponding volumes. The most famous instance of this problem is sphere packing in Euclidean n -dimensional space, where one asks how densely it is possible to fill \mathbb{R}^n with nonoverlapping balls of a fixed (and by homogeneity, irrelevant) radius.

In posing a code-theoretic analogue to the previous question, one specifies a finite alphabet A of q elements and a metric $d(x, y)$ on the set A^n of q -ary n -tuples; often d is the Hamming metric, and A^n is then called the Hamming space. One then asks for the size $A_q(n, d)$ of a maximal subset (code) of A^n for which any two points are at distance at least d apart. For $d = 2t + 1$, in particular, this is equivalent to finding the largest sphere packing (of radius t) in the Hamming space.

One frequently requires the centers of a sphere packing in \mathbb{R}^n to form a lattice. The analogous code-theoretic requirement is that the centers be not merely a subset but more stringently a subspace, i.e., that the codes be *linear*.

In Section 61.1 we consider sphere packing, and in Section 61.2 we consider sphere packing in connection with spherical codes. We look at error-correcting codes (including nonlinear codes and codes in metrics other than the Hamming) in Section 61.3, and at the construction of sphere packings, as well as packings of more general bodies, from error-correcting codes, in Section 61.4.

61.1 SPHERE PACKING AND QUADRATIC FORMS

SPHERE PACKING IN \mathbb{R}^n

The word “sphere,” as used in packing theory, usually denotes a solid ball. This is in contrast to the usage in the rest of mathematics, where “sphere” almost always refers to the outer surface alone. For historical reasons, the subject seems destined always to be called “sphere packing,” even though the terms “sphere” and “ball” are interchangeable within the sphere-packing literature.

GLOSSARY

The *ball of radius r* around the origin is

$$B^n(r) = \{x = (x_1, \dots, x_n) \in \mathbb{R}^n \mid x_1^2 + \dots + x_n^2 \leq r^2\}.$$

Its volume is $V_n r^n$, where

$$V_n = \int_{x \in B^n} dx_1 \cdots dx_n = \frac{2^n \Gamma(\frac{n+1}{2}) \pi^{(n-1)/2}}{\Gamma(1+n)} = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$$

is the volume of a unit ball $B^n = B^n(1)$.

Sphere packing: An arrangement of balls of the same radius, whose interiors are disjoint.

Lattice: The integral span of a basis of \mathbb{R}^n . Equivalently, a nonsingular linear transform of the points \mathbb{Z}^n with integer coordinates.

Lattice packing of spheres: The centers of the balls in the packing are all the points of a lattice.

Density of a sphere packing: Let \overline{P} be the union of balls in the packing P .

The density of P is

$$\delta(P) = \lim_{r \rightarrow \infty} \frac{\text{Vol}(\overline{P} \cap B^n(r))}{V_n r^n}.$$

Maximum packing density of the sphere: This is $\delta(n) = \sup \delta(P)$, where the supremum is over packings P of B^n .

Determinant of a lattice: The volume of the parallelepiped spanned by a basis for the lattice Λ , written $\det \Lambda$; it is independent of the basis. (Some authors call the determinant of a lattice the *square* of that volume. We refer to the squared volume as the *determinant of the quadratic form associated with the lattice*; see below.)

Density of a lattice packing of spheres: If the minimum distance between points of the lattice Λ is $2r$, then Λ provides a packing for balls of radius r , and its density is $\delta(\Lambda) = V_n r^n / \det \Lambda$.

Maximum lattice-packing density of the sphere: The quantity $\delta_L(n) = \sup \delta(\Lambda)$, the supremum being taken over lattice packings of B^n .

The **center density** $\delta^*(P)$ of a packing P is $\delta(P)/V_n$, the number of ball centers per unit volume of space when the minimum distance between the centers is normalized to 2. Analogously, $\delta^*(n) = \sup \delta^*(P) = \delta(n)/V_n$ and $\delta_L^*(n) = \delta_L(n)/V_n$.

The main problem in the theory of sphere packing is the determination of the quantities δ and δ_L in a given dimension n . Dense packing was Problem 18 of Hilbert's famous problem list [Hil01]. Some authors express results in terms of center density $\delta^*(n)$ instead, or in terms of $\log_2 \delta^*(n)$.

QUADRATIC FORMS IN n VARIABLES

GLOSSARY

Quadratic form associated with a lattice: If a lattice Λ is the integral span of the vectors $\mathbf{l}_1, \dots, \mathbf{l}_n$, which are the rows of the $n \times n$ matrix $L = (l_{ij})$, then this is the positive definite quadratic form

$$f_L(x_1, \dots, x_n) = \left(\sum_{i=1}^n x_i \mathbf{l}_i, \sum_{i=1}^n x_i \mathbf{l}_i \right) = \sum_{i,j=1}^n a_{ij} x_i x_j,$$

where $A = (a_{ij}) = LL^T$. (Here T means transpose.) The symmetric positive definite matrix A is called an ***inner product matrix*** for the lattice Λ , and $\det f = \det A$ is the determinant of the quadratic form associated with the lattice.

The ***arithmetic minimum*** of the positive definite quadratic form $f(x_1, \dots, x_n)$ is $M(f) =$ the smallest value taken on by f on $\mathbb{Z}^n \setminus \{O\}$.

Hermite's constant is $\gamma_n = \sup(M(f)/\sqrt[n]{\det f})$, the supremum taken as f varies over positive definite quadratic forms in n variables.

If $M(f)/\sqrt[n]{\det f} = \gamma_n$, then f is called ***absolutely extreme***.

Hermite's constant is related to the maximum center lattice-packing density of a sphere by

$$\left(\frac{\sqrt{\gamma_n}}{2} \right)^n = \delta_L^*(n).$$

Thus, the geometric problem of finding the densest lattice packing of a sphere is equivalent to the number-theoretic problem of maximizing the arithmetic minimum of a positive definite quadratic form of fixed determinant. This well-known equivalence is often unstated; papers on arithmetic minima frequently don't mention sphere packing, and vice versa. The historical trend is toward stating results in terms of sphere packing.

LAMINATED LATTICES

Define Λ_0 as the trivial lattice consisting of one point. For $n = 1, 2, 3, \dots$, we understand a ***laminated lattice*** Λ_n to be any n -dimensional lattice with these three properties: First, its minimum distance is 2. Second, some Λ_{n-1} is a sublattice. And third, Λ_n has minimal determinant among lattices satisfying the first two conditions.

Notice that it is not apparent from the definition how many laminated lattices there are. It turns out that there are two Λ_{11} 's, three Λ_{12} 's, and three Λ_{13} 's. For all the other values of n in $0 \leq n \leq 24$, there is exactly one Λ_n . There are exactly 23 different Λ_{25} 's, and for $n \geq 26$ there are probably a great many.

The ***Leech lattice***, Λ_{24} , has a profound influence on all smaller dimensions, and indeed all of the closest lattice packings of spheres that have been found to date are sections of the Leech lattice. These dense lattices are the lattices Λ_n for $1 \leq n \leq 24$ excepting $n = 11, 12, 13$, for which there are denser cross sections of Λ_{24} , called K_{11} , K_{12} , and K_{13} in [CS99].

It seems reasonable to consider the dimensions up to 24 separately.

DIMENSIONS UP TO 24

The values of $\delta_L(n)$ are known (i.e., proved) in dimensions $n \leq 8$, and conjectured with modest conviction in dimensions $9 \leq n \leq 24$. It is a theorem due to Thue [Thu10] that $\delta_L(2) = \delta(2) = \pi/\sqrt{12}$. This is the density of the usual hexagonal packing of circles in the plane, shown in [Figure 61.1.1](#).

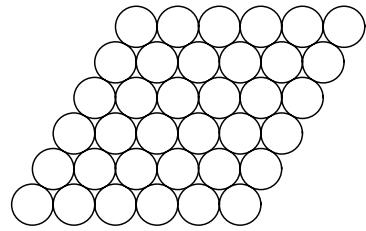


FIGURE 61.1.1

Closest packing of circles in the plane.

Gauss proved that $\delta_L(3) = \pi/\sqrt{18} = .7404\dots$. This is the density of the so-called **face-centered cubic lattice**, shown in Figure 61.1.2, which is generated by three equal vectors, each of which makes an angle of $\pi/3$ with the other two. For almost four centuries the Kepler Conjecture that $\delta(3) = \delta_L(3) = \pi/\sqrt{18}$ remained open.¹ The Rogers bound gives $\delta(3) \leq .7796\dots$, which was improved by Lindsey [Lin86] $\delta(B^3) \leq .7784$ and then by Muder [Mud93], who found that $\delta(B^3) \leq .773055\dots$. Finally, the Kepler Conjecture was proved by Hales in 1998 (see his nicely written overview [Hal00])

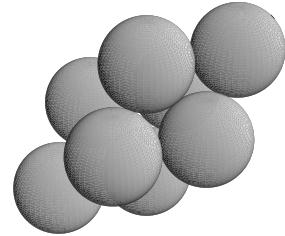


FIGURE 61.1.2

The densest lattice packing of spheres in three dimensions, which is also the densest packing.

It is interesting to see when the best known value of $\delta(n)$ is bigger than $\delta_L(n)$ for $n \leq 24$ (see Table 61.1.1.). The recent progress in constructing nonlattice packings denser than lattice ones started from Vardy's 20-dimensional packing [Var95]. Immediately afterward, Conway and Sloane [CS96] found that Vardy's nonlattice packing had analogues in dimension 22 (and dimensions 44 through 47), which also set new density records.

TABLE 61.1.1 Comparison of known values of δ^* and δ_L^* .

DIMENSION n	$\delta_L^*(n)$	$\delta^*(n)$	δ/δ_L
10	$\frac{1}{16\sqrt{3}}$	$\frac{5}{128}$	1.08523
11 and 13	$\frac{1}{18\sqrt{3}}$	$\frac{9}{256}$	1.09696
18	$\frac{1}{8\sqrt{3}}$	$\frac{3^9}{4^9}$	1.04040
20	$\frac{1}{8}$	$\frac{7^{10}}{2^{31}}$	1.05230
22	$\frac{1}{2\sqrt{3}}$	$\frac{11^{11}}{2^{23}3^{10}\sqrt{3}}$	1.15198

¹Generating the famous joke that “all physicists KNOW and all mathematicians BELIEVE that...”

The densest lattice packings known in dimensions up to $n = 10$ can be obtained from their predecessors merely by adjoining a new basis vector of the same length as all the others. Unfortunately the 11-dimensional dense lattice is not obtainable in this way; one must start from scratch.

Let $A = (a_{ij})$ be the following ten-by-ten matrix:

$$\begin{pmatrix} 4 & 2 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 4 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 4 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 4 & 2 & 2 & 0 & 0 & 0 & 1 \\ 0 & 2 & 2 & 2 & 4 & 2 & 0 & 0 & 0 & 1 \\ 2 & 2 & 2 & 2 & 2 & 4 & 2 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 4 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 & 4 & 1 \\ 0 & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 1 & 4 \end{pmatrix}.$$

Our ability to build by adding fixed-length basis vectors through dimension ten is reflected algebraically in a property of this quadratic form due to Chaundy [Cha46]:

$$\begin{aligned} f(x_1, \dots, x_{10}) &= \sum_{i,j=1}^{10} a_{ij} x_i x_j \\ &= 2(x_2 + x_3 + x_4 + x_5 + \frac{1}{2}x_6)^2 + 2(x_1 + x_2 + \frac{1}{2}x_6)^2 + 2(x_1 + x_3 + \frac{1}{2}x_6)^2 \\ &\quad + 2(x_4 + \frac{1}{2}x_6 + \frac{1}{2}x_{10})^2 + 2(x_5 + \frac{1}{2}x_6 + \frac{1}{2}x_{10})^2 + 2(x_7 + \frac{1}{2}x_6 + \frac{1}{2}x_{10})^2 \\ &\quad + 2(x_8 + \frac{1}{2}x_6 + \frac{1}{2}x_{10})^2 + 2(x_7 + x_8 + x_9 + \frac{1}{2}x_6)^2 + 2(x_9 + \frac{1}{2}x_{10})^2 + \frac{3}{2}x_{10}^2. \end{aligned}$$

The property is that

$$g(x_1, \dots, x_i) = f(x_1, \dots, x_i, 0, \dots, 0)$$

is an absolutely extreme quadratic form in i variables for $1 \leq i \leq 8$, and very probably is one for $i = 9, 10$ as well. Thus each new inner product matrix can be obtained from the previous one by adding a new column to the right, and its transpose to the bottom, of the previous matrix.

Although it is not possible to get past $n = 10$ in the manner described above, it is nonetheless possible, as stated in the previous section, to obtain all the best lattice packings known up to $n = 24$ by taking intersections of the 24-dimensional Leech lattice Λ_{24} (to be constructed in Section 61.4) with certain subspaces. Moreover, the highest known center densities attained for lattice packings of B^n are symmetric about $n = 12$. Let us write $\xi(x)$ for the reciprocal of the presumably optimal center density for a lattice in dimensions $12 \pm x$ for $0 \leq x \leq 12$. Its values are summarized in [Table 61.1.2](#).

DIMENSIONS UP TO 2048

The best packings known in these dimensions are still fairly good, but less likely to be optimal than those of lower dimension. (See [Table 61.1.3](#).) The success in these dimensions is due to the residual influences of combinatorial accidents such as the

TABLE 61.1.2 Reciprocal center densities of the densest known lattice packings of B^n in dimensions up to 24.

x	12	11	10	9	8	7	6	5	4	3	2	1	0
$\xi(x)$	1	2	$2\sqrt{3}$	$4\sqrt{2}$	8	$8\sqrt{2}$	$8\sqrt{3}$	16	16	$16\sqrt{2}$	$16\sqrt{3}$	$18\sqrt{3}$	27

TABLE 61.1.3 Base 2 logarithms of center densities of some lattice packings in moderately large dimensions, in comparison with upper and lower bounds.

DIMENSION n	LOWER BOUND	ATTAINED	UPPER BOUND	SOURCE
32	-8.22	1.359	5.52	Quebbemann
36	-7.10	1.504	8.63	Kschischang-Pasupathy
48	-2.05	14.039	15.27	Thompson
54	1.27	15.88	25.86	Elkies
60	5.04	17.435	27.85	Kschischang-Pasupathy
64	7.79	24.71	31.14	Elkies
80	20.40	40.14	49.90	Shioda
104	43.38	67.01	80.20	"
128	70.28	97.40	118.6	Elkies
256	257.76	294.80	357.0	"
512	759.21	797.12	957.4	"
1024	2016.6	2018.2	< 2418	"
2048	5041.87	4891	< 5827	"

existence of special algebraic curves, and the eight- and twenty-four-dimensional packings E_8 and Λ_{24} , the Leech lattice.

61.2 SPHERICAL CODES AND GENERAL BOUNDS ON SPHERE-PACKING DENSITY

PACKING IN THE UNIT SPHERE, OR SPHERICAL CODES

Consider the unit sphere S^n in $(n+1)$ -dimensional Euclidean space \mathbb{R}^{n+1} as a metric space with angular distance (the Riemann sphere) and its packing by metric balls of “radius” φ , i.e., by spherical caps of angular radius φ . The centers of any such packing form a so-called **spherical code** C with minimal angular distance 2φ (in brief, a 2φ -spherical code), since the angle between any two distinct points of C is at least 2φ . The **density** of a packing C with caps of angular radius φ is defined as $\sigma_n^{-1}|C|\sigma_n(\varphi)$, where

$$\sigma_n(\varphi) = (\sin \varphi)^n \sigma_{n-1} \int_0^1 \frac{x^{n-1} dx}{\sqrt{1 - x^2 \sin^2(\varphi)}}$$

is the measure of a cap of angular radius φ and $\sigma_n = 2\sigma_n(\pi/2) = (n+1)V_{n+1}$ is

the measure of the surface of S^n . Let $M(n, \varphi)$ be the largest possible cardinality of a φ -spherical code and

$$\delta(n, \varphi) = \frac{M(n, 2\varphi)\sigma_n(\varphi)}{\sigma_n}$$

be the largest possible packing density in S^n with caps of angular radius φ . There is a natural relationship between dense sphere packing in Euclidean space and packing in the sphere, namely,

$$\delta(n) = \lim_{\varphi \rightarrow 0} \delta(n, \varphi).$$

The value $\tau(n) = M(n-1, \pi/3)$ is known as the *kissing*, or *contact number*, and equals the maximal number of equal spheres in \mathbb{R}^n that touch one sphere of the same radius without overlapping. For 3-dimensional space it was the subject of discussion between Newton ($\tau(3) := 12$) and Gregory ($\tau(3) := 13$) at the end of the 17th century. Spherical codes also have many applications in communications as sets of signals for various modulation schemes.

For $\varphi \geq \pi/2$ the problem of spherical codes (or packing of caps in S^n) is solved completely, due to the following **Rankin upper bounds** [Ran55]:

$$M(n, \varphi) \leq -\frac{1 - \cos \varphi}{\cos \varphi}, \quad \text{if } \cos \varphi < 0$$

$$M(n, \varphi) \leq 2(n+1) \frac{1 - \cos \varphi}{1 - (n+1) \cos \varphi}, \quad \text{if } \cos \varphi < 1/(n+1),$$

which show that the regular simplex on $i+1$ vertices ($i = 1, \dots, n+1$) and the set of vertices of the octahedron $\{\pm \mathbf{e}_i\}$ are optimal spherical codes.

Hence, for $\varphi \geq \pi/2$ it is not possible to arrange more than a linear number of points (in fact, not more than $2(n+1)$) on S^n in such a way that the angle between any two points is at least φ . However, for any fixed $\varphi < \pi/2$ one can specify exponentially many points on S^n with the desired property. Indeed, any optimal φ -spherical code is at the same time a covering of S^n by caps of angular radius φ . Since the density of any covering is at least 1 it follows that [Sha59]

$$M(n, \varphi) \geq \sigma_n / \sigma_n(\varphi) > (\sin \varphi)^{-n} \quad \text{for } \varphi < \pi/2,$$

or, equivalently,

$$n^{-1} \log M(n, \varphi) \geq -\log \sin \varphi.$$

On the other hand, upper bounds of Rankin and Coxeter state that

$$n^{-1} \log M(n, \varphi) \leq -\log \sin(\varphi/2) - 0.5 + o(1).$$

The best known asymptotic bound (the KL bound [KL78]) was obtained by establishing a deep relationship between the size of a spherical code (and its combinatorial properties—see [Lev98]), on the one hand, and zonal spherical functions of $SO(n+1)$, on the other hand.

It follows from the theory of group representations that a continuous and invariant (under the action of the group $SO(n+1)$) kernel $F(x, y)$ is positive semidefinite (or nonnegative definite), i.e.,

$$F(x, y) = f((x, y)) = a_0 + \sum_{i=1}^N u_i(x)u_i(y), \quad a_0 > 0$$

if and only if

$$f(t) = \sum_{i=0} f_i C_i^m(t),$$

where $f_i \geq 0$ for all $i \geq 1$, $f_0 > 0$, $m = (n-1)/2$, and

$$C_i^m(t) = \sum_{j=0}^{i/2} (-1)^j \binom{i-j}{j} \binom{i-j+m-1}{i-j} (2t)^{i-2j}$$

are the *Gegenbauer*, or *ultraspherical polynomials*. Consider a polynomial $f(t)$ with the following properties:

- 1) $f_0 > 0$ and $f_i \geq 0$ for all $i \geq 1$;
- 2) $f(t) \leq 0$ for $-1 \leq t \leq \cos \varphi$.

Then for an arbitrary φ -spherical code C we have

$$S = \sum_{x,y \in C} f((x,y)) = f_0|C|^2 + \sum_i \sum_{x \in C} |u_i(x)|^2 \geq f_0|C|^2.$$

On the other hand,

$$S = \sum_{x=y \in C} f((x,y)) + \sum_{x \neq y \in C} f((x,y)) \leq |C|f(1).$$

Hence we obtain the following inequality, often called the *linear programming bound* (see [CS99]):

$$M(n, \varphi) \leq f(1)/f_0.$$

The optimal choice of polynomial $f(t)$ is an open problem. With the polynomial

$$f(t) = (C_{k+1}^m(t)C_k^m(s) - C_k^m(t)C_{k+1}^m(s))^2/(t-s)$$

it was shown in [KL78] that

$$M(n, \varphi) \geq 4 \binom{k+n-1}{k} (1 - \xi_{k+1}^{(m)})^{-1} \quad \text{if } \cos \varphi < \xi_k^{(m)},$$

where $\xi_k^{(m)}$ is the largest root of $C_k^m(t)$ on $(-1, +1)$, $m = (n-1)/2$. Then the asymptotic formula for $\xi_k^{(m)}$ [KL78] leads to the aforementioned **KL bound**:

$$n^{-1} \log M(n, \varphi) \leq \frac{1 + \sin \varphi}{2 \sin \varphi} \log(\frac{1 + \sin \varphi}{2 \sin \varphi}) - \frac{1 - \sin \varphi}{2 \sin \varphi} \log(\frac{1 - \sin \varphi}{2 \sin \varphi}) + o(1).$$

Since

$$(\sin(\varphi/2))^n M(n, \varphi) \leq (\sin(\theta/2))^n M(n, \theta) \quad \text{for } \varphi < \theta,$$

the KL bound can be improved to

$$n^{-1} \log M(n, \varphi) \leq -\log \sin(\varphi/2) - 0.599 + o(1)$$

for $\varphi \leq \varphi^*$, where $\varphi^* \approx 63$ is the root of $\cos \varphi (\ln(1 + \sin \varphi) - \ln(1 - \sin \varphi)) + (1 + \cos \varphi) \sin \varphi = 0$.

For the particular case of the kissing number ($\varphi = \pi/3$), the lower and upper bounds have the following form:

$$1 - \frac{\log_2 3}{2} + o(1) = 0.2075\dots + o(1) \leq n^{-1} \log_2 \tau(n) \leq 0.401 + o(1).$$

Let us note that it is unknown if the kissing number $\tau_L(n)$ for lattices can be exponentially large. The best known result is that $\tau_L(n) \geq 2^{\Omega(\log^2(n))}$. On the other hand, Alon [Alo97] has constructed, on the basis of error-correcting codes, a finite packing of balls in \mathbb{R}^n whose minimum kissing number is at least $2\sqrt{n}$. It follows from the recent result [ABV01] that the corresponding algebro-geometric codes form a finite packing of balls in which every ball touches the same number, $2^{\Omega(n)}$, of “neighbors,” and this construction can be easily extended to an infinite packing of balls with the same property.

A better choice of polynomials was found by Levenshtein [Lev79]. The corresponding polynomial of odd degree (a simpler case) has the following form:

$$f_{2k-1}^{(s)} = (t-s)\left(\sum_{i=0}^{k-1} r_i(P_k(s) - P_i(s))P_i(t)\right)^2,$$

where $P_i(t) = C_i^m(t)/C_i^m(1)$, $m = (n-1)/2$, and $r_i = \binom{i+n-1}{i} + \binom{i+n-2}{i-1}$. With these polynomials it was shown [Lev79] that

$$M(n, \varphi) \leq 2 \binom{k+n-1}{k-1} \quad \text{if } \cos \varphi < \xi_{k-1}^{(m+1)}.$$

Despite the fact that these bounds are asymptotically the same as the KL bound, they are always (also for k even) better than those coming from [KL78], and enable one to prove the optimality or asymptotic optimality of some known packings in the sphere. First of all, these bounds led [Lev83] to the first two infinite families of spherical codes that are asymptotically optimal (by cardinality for a given angle φ). Both families are constructed from known binary error-correcting codes (see Section 61.3) by embedding them into S^{n-1} via the mapping $X_i = (-1)^{x_i} n^{-1/2}$. The first family is based on the well-known Kerdock codes (see [MS78]) and yields the following parameters: $n = 2^{2l}$, $M = n^2$, $\cos \varphi = n^{-1/2}$. The second family, based on the Sidelnikov codes [Sid71], has the parameters $n = (2^{4l} - 1)/(2^l + 1)$, $M = 2^{4l} \approx n^{4/3}$, $\cos \varphi = n^{-2/3}$.

The most impressive results derived from these bounds are for the kissing numbers, where it was proved independently [Lev79, OS79] that $\tau_8 = 240$ (achieved on the lattice E_8) and $\tau_{24} = 196560$ (achieved on the Leech lattice Λ_{24}). There are also eleven other examples (see Chapter 2) of point arrangements on the unit sphere S^n (for rather small $n \leq 23$) whose optimality follows from the new bounds. It is worth mentioning that these bounds give an analytic proof that the maximal value of the minimal separation angle for 12 points on S^2 is $\arccos 1/\sqrt{5}$.

GENERAL BOUNDS ON SPHERE-PACKING DENSITY

Clearly $\delta(n) \geq \delta_L(n)$. It was shown elegantly by K.M. Ball [Bal93] that

$$\delta_L(n) \geq (n-1)2^{1-n}\zeta(n).$$

The right-hand side is $2^{-n(1+o(1))}$ for large n , and bounds of that form have been known for a long time [Min69]. Note that the simple observation that any maximal packing of spheres should be a covering by spheres of twice larger radius immediately leads to $\delta(n) \geq 2^{-n}$. Ball’s result was a refinement, for spheres, of the ***Minkowski-Hlawka bound*** [Hla43],

$$\delta_L(G) \geq 2^{1-n}\zeta(n),$$

which is applicable to all compact, convex, O -symmetric bodies G . In the other direction, Yaglom's inequality allows us to “transform” upper bounds on the size $M(n, \varphi)$ of spherical codes into an upper bound on $\delta(n)$. This states that

$$\delta(n) \leq (\sin \frac{\varphi}{2})^n M(n, \varphi) \quad \text{for } 0 < \varphi < \pi/2.$$

For $\varphi = \pi/3$ this inequality can be slightly strengthened: $\delta(n) \leq 2^{-n} \tau(n)$. Then an application of the improved KL bound, or, in particular, the upper bound for the kissing numbers, gives the **Kabatiansky-Levenshtein bound** [KL78]

$$\delta(n) \leq 2^{-(.599\dots)n(1+o(1))}.$$

For $n \leq 42$, the Kabatiansky-Levenshtein bound is not as good as the **Rogers bound** [Rog64], which is given by $\delta(n) \leq \eta_n$, where η_n is the fraction of a solid regular simplex of edge 2 in R^n that is covered by the $n + 1$ unit balls centered at its vertices. The quantity η_n is bounded above by

$$\frac{n^{\frac{n+3}{2}} \sqrt{\pi} 2^{u-n}}{e^{1+\frac{n}{2}} \Gamma(1 + \frac{n}{2})},$$

where $u \leq 21/(4n + 10)$. For large n , we have $\eta_n = (n/e)2^{-n/2}(1 + o(1))$.

61.3 ERROR-CORRECTING CODES

Known results on constructions of error-correcting codes mostly address the case where the size q of the alphabet A is a power of a prime; in this case A will be considered below as a finite field \mathbb{F}_q , unless otherwise specified.

GLOSSARY

A **q -ary n -dimensional Hamming space** \mathbb{H}_q^n is the set of n -tuples over a q -ary alphabet A with the **Hamming distance** $d(x, y)$ defined as the number of positions where x and y are distinct.

The **volume (cardinality) of the ball of radius t** around a point $a \in \mathbb{H}_q^n$ equals $V_n(t; q) = \sum_{j=0}^t \binom{n}{j} (q-1)^j$.

A **q -ary code of length n** is a subset of H_q^n . Elements of the code are called **codewords**. **Binary codes** have $q = 2$.

The **(minimum) Hamming distance** $d(C)$ of a code C is the minimum of $d(x, y)$ for $x \neq y \in C$. Hence, a code with Hamming distance $d(C) \geq 2t + 1$ is the same as a packing of balls of radius t in the Hamming space. In other words, a code C can correct t errors iff $d(C) \geq 2t + 1$. $A_q(n, d)$ denotes the maximum possible cardinality of a code C with $d(C) \geq d$.

A **q -ary linear $[n, k]$ -code** is a k -dimensional subspace of \mathbb{F}_q^n . An $[n, k]$ -linear code C can be conveniently described by the generator $k \times n$ -matrix G_C whose columns form a basis of C , or by a parity-check $(n - k) \times n$ -matrix H_C whose columns form a basis of the dual space C^\perp . Any s columns of H_C are linearly independent iff $d(C) \geq s + 1$. Hence, $d(C) \leq n - k + 1$ for any linear $[n, k]$ -code C (the **Singleton bound**).

The **Hamming code** has length $n = (q^m - 1)/(q - 1)$, where q is a prime power and $m = 2, 3, \dots$, and is defined by a parity-check $(n - m) \times n$ -matrix whose columns are all $(q^m - 1)/(q - 1)$ noncollinear vectors of \mathbb{F}_q^m (i.e., all points of $(m-1)$ -dimensional projective space over \mathbb{F}_q). A Hamming code has distance $d \geq 3$ (in fact, equal to 3) since any two columns of its parity-check matrix are linearly independent.

Density of a sphere packing in a Hamming space: Let C be a sphere packing in \mathbb{H}_q^n of radius t , i.e., a code with distance $d(C) \geq 2t + 1$. The density of C is $\delta(C) = |C|V_q(t, n)/q^n$. The maximum packing density in the Hamming space is $\delta_q(n, t) = A_q(n, 2t + 1)V_q(t, n)/q^n$.

The obvious inequality $\delta(C) \leq 1$ is known in coding theory as the **Hamming bound**. In contradistinction to Euclidean space, there are sphere packings with density 1 in a Hamming space. Such packings (codes) are called **perfect codes**. There are only two perfect codes with $t > 1$, namely the binary and ternary Golay codes (see below). For $t = 1$ and q a prime power, the known perfect codes include the infinite family of q -ary Hamming codes (see [MS78]).

It is an open question whether packings of radius $t = 1$ with density 1 in a Hamming space (i.e., perfect single-error correcting codes) exist in the case where q is not a prime power.

For x real we write $\binom{x}{j} = x(x - 1)(x - 2) \cdots (x - j + 1)/(j!)$ and define the

Krawtchouk polynomial

$$K_k^{(n)}(x) = \sum_{j=0}^k (-1)^j \binom{x}{j} \binom{n-x}{k-j} (q-1)^{k-j}.$$

The role of Krawtchouk polynomials in a Hamming space is analogous to that of Gegenbauer polynomials in Euclidean space

CYCLIC CODES

GLOSSARY

Let $\mathbb{F}_q[x]$ be the ring of polynomials in x with coefficients in \mathbb{F}_q and $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$ be its quotient ring modulo $x^n - 1$. It is convenient to identify a vector $\mathbf{a} = (a_0, \dots, a_{n-1})$ in \mathbb{F}_q^n with a polynomial $a(x) = a_0 + a_1x^1 + \cdots + a_{n-1}x^{n-1}$ in $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$. Then the polynomial $xa(x)$ corresponds to a cyclic shift of the vector \mathbf{a} . Let $g(x) \in \mathbb{F}_q[x]$ divide $x^n - 1$. The ideal $\langle g(x) \rangle$ in $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$ is called a **cyclic code** since all its vectors are invariant under the cyclic shifts. It has **length** n and **dimension** $k = n - \deg g$. The polynomial $g(x)$ is called the **generator polynomial** of the code.

Let $n = (q^m - 1)/(q - 1)$ and let β be a primitive n th root of unity in $GF(q^m)$.

Assume that m and $q - 1$ are relatively prime. Let $g(x) \in GF(q)[x]$ be the minimal polynomial of β . Then the ideal $\langle g(x) \rangle$ is equivalent to a *Hamming code* in the sense that one code can be obtained from the other by applying a fixed permutation to each codeword.

Let α be a primitive n th root of unity in some extension field \mathbb{F}_{q^m} of \mathbb{F}_q . Consider the generator polynomial $g(x)$ that has roots $\alpha^L, \alpha^{L+1}, \alpha^{L+2}, \dots, \alpha^{L+s-2}$ and is such that $g(x)$ is the LCM of the minimal polynomials of those powers of α . Then $\langle g(x) \rangle$ is called a **BCH code of designed distance s** because up to $\lfloor(s-1)/2\rfloor$ errors can be corrected efficiently (with complexity $\Omega(n^2)$) by the *Berlekamp-Massey algorithm* (see [MS78]). For $L = 1$ it is called a **narrow-sense BCH code**. If $n = q^m - 1$, so that α is a primitive element of \mathbb{F}_{q^m} , the code is called a **primitive BCH code**.

A primitive BCH code with $m = 1$, i.e., with $n = q - 1$, is called a **Reed-Solomon code**. These codes are optimal since they achieve the Singleton bound $d(C) \leq n - k + 1$.

Let p be a prime that we reserve for the size of the symbol field. We assume that the code length n is also prime and require that n divides $p^{(n-1)/2} - 1$ so that p is a quadratic residue mod n . (If $p = 2$ this implies that n is of the form $8j \pm 1$.) Let Q_+ be the set of quadratic residues (i.e., squares) mod n and let Q_- be the set of nonresidues. Let α be a primitive n th root of unity in some extension field of $GF(p)$, and let

$$q_+(x) = \prod_{j \in Q_+} (x - \alpha^j) \in GF(p)[x], \quad q_-(x) = \prod_{j \in Q_-} (x - \alpha^j) \in GF(p)[x].$$

Then $(x - 1)q_+(x)q_-(x) = x^n - 1$. Define four cyclic codes as follows:

$$C_1 = \langle q_+(x) \rangle, \quad C_2 = \langle (x - 1)q_+(x) \rangle, \quad C_3 = \langle q_-(x) \rangle, \quad C_4 = \langle (x - 1)q_-(x) \rangle.$$

These are called **quadratic residue (QR) codes**.

The **Golay codes** G_{23}, G_{11} are special quadratic residue codes over $GF(2)$ and $GF(3)$, respectively. Their error correction capacity is three errors for G_{23} and two errors for G_{11} . These codes are the only two perfect codes that are capable of correcting more than one error.

The parameters of these codes are found in [Table 61.3.1](#) below.

OTHER LINEAR CODES FOR THE HAMMING METRIC

GLOSSARY

The **extended Golay codes** G_{24}, G_{12} are obtained from the Golay codes by appending a digit, that is, an element of \mathbb{F}_q , to each codeword to make the sum of the n digits of each codeword equal to zero mod q . (Thus $n = 24$ and $q = 2$ for G_{24} , while $n = 12$ and $q = 3$ for G_{12} .)

The **binary Reed-Muller code of order r** , where $0 \leq r \leq m$, consists of the vectors that correspond to (and form the outputs of) all Boolean polynomials of degree at most r over \mathbb{F}_2 in the binary variables v_1, v_2, \dots, v_r .

A **Goppa code** is a linear code

$$C = \left\{ v = (v_1, \dots, v_n) \in F_q^n \mid \sum_{i=1}^n \frac{v_i}{z - P_i} \equiv 0 \pmod{G(z)} \right\}$$

where $v_i \in \mathbb{F}_q$, $P_i \in \mathbb{F}_{q^m}$, and $G(z)$ is a polynomial over \mathbb{F}_{q^m} for which $G(P_i) \neq 0$, all of these holding for $1 \leq i \leq n$.

Algebro-geometric codes are constructed as follows. Let X be a smooth, projective, algebraic curve over \mathbb{F}_q , absolutely irreducible over \mathbb{F}_q . Let $X(\mathbb{F}_q) = \{P_1, \dots, P_n\}$ be the set of \mathbb{F}_q points of X , so that $n = |X(\mathbb{F}_q)|$. Let g be the genus of X , that is, the genus of the compact Riemann surface associated with X in the sense of the Riemann-Roch theorem. Choose a divisor D of X whose associated vector space $L(D)$ has dimension k over \mathbb{F}_q . Our code $C \subset \mathbb{F}_q^n$ is the image of $L(D)$ under the evaluation map $\text{Ev} : L(D) \rightarrow \mathbb{F}_q^n$, $\text{Ev} : f \mapsto (f(P_1), \dots, f(P_n))$.

The parameters of these codes can be found in Table 61.3.1.

TABLE 61.3.1 Comparison of parameters for certain types of codes for the Hamming metric. Distances and dimensions are at least as large as stated.

TYPE	BLOCK LENGTH n	DIMENSION k	DISTANCE d
Hamming	$\frac{q^m - 1}{q - 1}$	$n - m$	3
Primitive BCH	$q^m - 1$	$n - \deg g(x)$	$d \geq s$
Reed-Solomon	$q - 1$	$n - s + 1$	$d = s$
QR codes C_1, C_3	prime	$(n+1)/2$	$d \geq \sqrt{n}$
QR codes C_2, C_4	prime	$(n-1)/2$	$d \geq \sqrt{n}$
Ternary Golay G_{11}	11	6	5
Binary Golay G_{23}	23	12	7
Ext. Golay G_{12}	12	6	6
Ext. Golay G_{24}	24	12	8
Reed-Muller	2^m	$1 + \binom{m}{1} + \dots + \binom{m}{r}$	2^{m-r}
Goppa	n	$n - m \deg G$	$1 + \deg G$
Algebro-geometric	$n = \text{no. of } GF(q)\text{-points on curve, so that } n \leq q + 1 + 2g\sqrt{q}$	$k = \dim L(D)$	$n + 1 - g - k$, where $g = \text{genus of curve}$

GENERAL BOUNDS ON CODE SIZE FOR THE HAMMING METRIC

There are many analogies between spherical codes and error-correcting codes, especially in the binary case, when the mapping $\lambda : \mathbb{H}_2^n \rightarrow S^{n-1}$ given by $\lambda(x_1, \dots, x_n) = (X_1, \dots, X_n)$ with $X_i = (-1)^{x_i} n^{-1/2}$ embeds the n -dimensional binary Hamming space into S^{n-1} and $(X, Y) = 1 - 2d(x, y)/n$.

Clearly any optimal code of Hamming distance d is at the same time a covering of the Hamming space with spheres of radius $d - 1$. This implies that

$$A_q(n, d) \geq q^n / V_n(d-1; q)$$

since the density of a covering is at least 1 (the **Gilbert bound**). This bound is also valid for linear codes, in which case it has a slightly stronger form (the **Varshamov bound**):

$$A_q^{(Lin)}(n, d) \geq q^{n - \lceil \log_q V_n(d-2; q) \rceil}.$$

It is common for asymptotic problems of coding theory to consider one of the two following asymptotic processes: $d = \text{const}$ and $d/n = \Delta = \text{const} > 0$. For the latter case it is convenient to consider the code rate $R(C) = n^{-1} \log_q |C|$. The **optimal**

code rate $R_q(\Delta)$ is defined as

$$R_q(\Delta) = \lim_{n \rightarrow \infty} n^{-1} \log_q A_q(n, \Delta n).$$

Then both bounds have the same form called, known as the **VG bound**:

$$R(\Delta) \geq 1 - H_q(\Delta) - \Delta \log_q(q-1) \quad \text{for } \Delta \leq (q-1)/q,$$

where $H_q(x) = -(x \log_q(x) + (1-x) \log_q(1-x))$ is the q -ary **entropy function**. One of the longest-standing open problems in coding theory is whether a family of cyclic codes of a fixed rate R can have Hamming distance that grows linearly with n , i.e., the relative distance $\Delta > 0$. S. Berman proved that for any family of cyclic codes C_i whose length n has only a fixed number of prime divisors in its factorization the Hamming distance is bounded above by some absolute constant that depends on the number of divisors.

Surprisingly, the VG bound is known not to be tight as $n \rightarrow \infty$ for a fixed $\Delta = d/n$ and for q an even power of a prime greater than or equal to 49, a result due to Tsfasman, Vlăduț, and Zink [TVZ82]. Namely, the parameters of AG codes asymptotically approach the **AG bound**:

$$R_q(\Delta) \geq 1 - \Delta - \frac{1}{\sqrt{q}-1}.$$

Note that these codes are polynomially constructible (the first construction with polynomial complexity was due to S. Vlăduț; by a recent result of [SAK⁺01] the construction complexity is $O((n \log_q n)^3)$) and even polynomially decodable. The VG bound only guarantees the existence of such codes, whereas their construction and decoding have complexity $\exp(\Omega(n))$.

The Hamming bound for the optimal code rate has the form

$$R_q(\Delta) \leq 1 - H_q\left(\frac{\Delta}{2}\right) - \log_q(q-1)\frac{\Delta}{2}.$$

The simple recursion $A_q(n, d) \leq q^{n-n'} A_q(n', d)$ leads to the **Singleton bound** (for arbitrary codes) $\log_q A_q(n, d) \leq n - d + 1$. Asymptotically we obtain

$$R(\Delta) \leq 1 - \Delta.$$

The **Plotkin bound**

$$A_q(n, d) \leq d(d - (1 - q^{-1})n)^{-1} \quad \text{if } d - (1 - q^{-1})n > 0$$

is an analog of the Rankin bound for spherical codes. It is attained on codes dual to the Hamming codes (which are mapped by the embedding λ to right simplexes in \mathbb{R}^n , $n = 2^m - 1$, $m = 2, 3, \dots$). The same recursion, $A_q(n, d) \leq q^{n-n'} A_q(n', d)$, leads to a more general form of the Plotkin bound: $A_q(n, d) \leq dq^{n-\lfloor q(d-1)/(q-1) \rfloor}$. Asymptotically this gives

$$R_q(\Delta) \leq 1 - \frac{q}{q-1}\Delta.$$

Note that the Plotkin bound implies (analogously to the Rankin bound) that for $d \geq n(q-1)/q$ it is not possible to find more than a linear (in n) number of points in H_q^n with the property that the Hamming distance between any two points is at

least d . On the other hand, the VG bound guarantees that for any fixed $\Delta < 1 - 1/q$ there exists a code of cardinality $\exp(\Omega(n))$ with Hamming distance $d \geq \Delta n$. Note that “critical points” $\varphi = \pi/2$ for S^{n-1} and $\Delta = 1/2$ for H_2^n are identified by the embedding λ .

Almost all known upper bounds for codes can be obtained by an application of the linear programming (LP) bound approach introduced by P. Delsarte [Del73], namely,

$$A_q(n, d) \leq f(1)/f_0$$

for any polynomial $f(x) = \sum_{i=0}^n f_i K_i^{(n)}(x)$ with the following properties:

- 1) $f_0 > 0$ and $f_i \geq 0$ for all $i \geq 1$;
- 2) $f(x) \leq 0$ for $x \geq d$.

The best asymptotic bound was obtained in [MRRW77]. For binary codes, in particular, it gives

$$R_2(\Delta) \leq H_2(1/2 - \sqrt{\Delta(1-\Delta)}) \quad \text{for } 0.273 < \Delta \leq 1/2.$$

It was proved recently [Sam01] that the LP approach is limited by the arithmetic mean of this bound and the VG bound.

Of course all the bounds mentioned above for the code rate can be rewritten as bounds for the maximum packing density of Hamming spheres of radius t in H_q^n . For instance, the VG bound for binary codes states that $n^{-1} \log_2 \delta(n, \tau n) \geq H(\tau) - H(\tau/2)$ for $\tau = \text{const}$. For the case of fixed radius t the Hamming bound is tighter. It is known [KP88] that for q a prime power,

$$\lim_{n \rightarrow \infty} \delta_q(n, 1) = 1.$$

In the binary case, the existence of the BCH codes implies that the maximum packing density of Hamming spheres of fixed radius t is separated from zero:

$$\delta_2(n, t) \geq 1/t! + o(n).$$

For $t = 2$, the existence of the Preparata codes (see [MS78]) implies that

$$\limsup \delta_2(n, 2) = 1.$$

In the nonbinary case much less is known. It is an open problem whether the maximum packing density is separated from zero, or in a slightly weaker form whether

$$\lim_{n \rightarrow \infty} \frac{n - \log_q A_q(n, 2t+1)}{\log_q n} = t.$$

This conjecture is known to be true for $t = 2$ and $q = 3, 4$. For $t = 2$ and arbitrary prime power q the best known result [Dum95] states that

$$\frac{n - \log_q A_q(n, 5)}{\log_q n} \leq 7/3.$$

CODES FOR EXOTIC METRICS

Let G be a compact convex O -symmetric body in \mathbb{R}^n , and fix an odd prime p . Regard \mathbb{F}_p as lying in \mathbb{R}^n by making the identification

$$\mathbb{F}_p = \{-(p-1)/2, \dots, -1, 0, 1, \dots, (p-1)/2\}.$$

Thus $\mathbb{F}_p^n = \mathbb{Z}^n \cap pQ$, where Q is the unit hypercube

$$\{x \in \mathbb{R}^n \mid \max(|x_1|, \dots, |x_n|) \leq \frac{1}{2}\}.$$

The **G -norm** of a point x of $GF(p)^n$ is

$$\|x\|_G = \inf\{\mu \geq 0 \mid x \in p\mathbb{Z}^n + \mu G\}.$$

An $[n, k, d, p, G]$ **code** C is a k -dimensional subspace of $GF(p)^n$ such that $\|x - y\|_G \geq d$ whenever $x, y \in C$ with $x \neq y$. For $x \in GF(p)^n$, let

$$B_{d,p,G}^n(x) = \{y \in GF(p)^n \mid \|y - x\|_G \leq d\}$$

be a metric ball and let $V_n(d; p, G) = |B_{d,p,G}^n(x)|$ be its volume. (Note that $V_n(d; p, G)$ does not depend on x .) It can be shown that $V_n(d; p, G) \leq |\mathbb{Z}^n \cap dG|$. An ordinary (in coding theory) count of “bad” parity-check matrices proves the following analogue of the VG bound for these codes: an $[n, k, d, p, G]$ code exists provided $k \leq n - \lceil \log_p V_n(d - 1; p, G) \rceil$.

The second author [Rus89] used $[n, k, d, p, G]$ codes and Construction A of Leech and Sloane (described in the next section) with $G = B^n$ to produce packings of the sphere with density $2^{-n(1+o(1))}$ as $n \rightarrow \infty$.

61.4 CONSTRUCTIONS OF PACKINGS

While we know that $\delta(B^n) \geq \delta_L(B^n) \geq 2^{-n(1+o(1))}$, we don’t know explicit arrangements nearly so dense when n is large. In principle, Minkowski reduction theory makes finding the densest lattice packing of B^n a finite problem (and those imbued with a pure enough mathematical spirit may be satisfied with this) but still it is nice to have explicit arrangements. Typically, there is a tradeoff: the more explicit, or “constructive,” the method is, the worse it fares as $n \rightarrow \infty$.

We mention, below, five constructions of packings from codes. Constructions A, B, and C are due to Leech and Sloane; D to Bos, Conway, and Sloane; and E to Barnes and Sloane. For more details, see [CS99].

CONSTRUCTION A

If C is a binary $[n, k, d]$ code, its Construction A lattice is $\Lambda_A(C) = 2\mathbb{Z}^n + C$. (If C is nonlinear, this gives a periodic but nonlattice arrangement.) We have $\det \Lambda_A(C) = 2^{n-k}$, and the lattice provides a packing for spheres of radius $\min(1, \frac{1}{2}\sqrt{d})$.

If C is an $[n, k, d, p, B^n]$ code, then its Construction A lattice is $\Lambda(C) = p\mathbb{Z}^n + C$. Then $\det \Lambda(C) = p^{n-k}$, and the lattice packs spheres of radius $\frac{1}{2}\min(d, p)$. If C is an $[n, k, d, p, G]$ code, and $d \leq p$, then $\Lambda(C)$ packs the body $\frac{1}{2}dG$.

CONSTRUCTION B

Let C be a binary $[n, k, d]$ code for which every codeword has even Hamming weight. The Construction B lattice of C consists of all those points (x_1, \dots, x_n) of the Construction A lattice for which $x_1 + \dots + x_n$ is divisible by 4. We can call it

$\Lambda_B(C)$. We have $\det \Lambda_B(C) = 2^{n-k+1}$, and the lattice packs spheres of radius $\frac{1}{2} \min(\sqrt{d}, \sqrt{8})$. (If C is a nonlinear even-weight code, this gives a periodic but nonlattice arrangement.)

CONSTRUCTION C

Since this produces nonlattice packings, and Construction D applied to nested linear codes produces lattice packings of equal density, we omit a description of Construction C.

CONSTRUCTION D

Let C_i be a binary $[n, k_i, d_i]$ code, with $C_{i-1} \supset C_i$ and $d_i \geq 4^i/u$ for $1 \leq i \leq t$, where $u \in \{1, 2\}$. Let $C_0 = GF(2)^n$, so that $k_0 = n$ and $d_0 = 1$. Let $C_{t+1} = \{(0, \dots, 0)\}$, so that $k_{t+1} = 0$ and $d_{t+1} = \infty$. Take a row-vector basis

$$c_1 = (c_{11}, c_{12}, \dots, c_{1n}), c_2 = (c_{21}, c_{22}, \dots, c_{2n}), \dots, c_n = (c_{n1}, c_{n2}, \dots, c_{nn})$$

spanning $GF(2)^n$, selected so that these row vectors can be permuted with one another to produce an upper triangular matrix, and so that c_1, c_2, \dots, c_{k_i} span C_i for $0 \leq i \leq t$. The Construction D lattice for this nested set of codes is

$$\Lambda_D(\{C_i\}) = \left\{ x + y \mid x \in 2\mathbb{Z}^n \text{ and } y \in \sum_{i=1}^t \sum_{j=1}^{k_i} b_{ij} \frac{c_{ij}}{2^{i-1}}, \text{ where each } b_{ij} \in \{0, 1\} \right\}.$$

The lattice has determinant $2^{n-(k_1+k_2+\dots+k_t)}$ and can pack spheres of radius $1/\sqrt{u}$.

There is a similar construction, Construction D', which uses parity checks rather than generators, and produces lattices of the same density as those of Construction D. We omit the description.

CONSTRUCTION E

This is a sort of nonbinary version of Construction D. In this subsection only, we permit codes to have nonfield symbol sets. Thus a “linear code” is merely an additive abelian group, not a vector space over the symbol field as elsewhere in this chapter.

Let $\Lambda \subset \mathbb{R}^n$ be a lattice with minimum Euclidean distance d between its points. Let D be a dilatation composed with an orthogonal transformation. Fix integers $p \geq 1$ and $r \geq 0$. Suppose $D\Lambda \subset \Lambda$, and that $pD^{-1} = a_0D^0 + a_1D^1 + \dots + a_rD^r$ for certain integers a_0, \dots, a_r . Suppose all the $p^b - 1$ nonzero congruence classes of $D^{-1}\Lambda/\Lambda$ have minimum distance from the origin at least $d/\sqrt[p]{|\det D|}$.

Let C_i be a p^{bk_i} -element subgroup of E^m , where $E = \Lambda/D\Lambda \cong (\mathbb{Z}/p\mathbb{Z})^b$, and $C_{i-1} \supset C_i$, for $1 \leq i \leq t$. Endowing these with the Hamming metric, we regard C_i as an $[m, k_i, d_i]$ code. We assume that the largest code, C_0 , has parameters $[m, m, 1]$.

Let $x \in \{0, 1, 2, \dots, p-1\}$ belong to the congruence class $\bar{x} \in \mathbb{Z}/p\mathbb{Z}$. Let $V : E \rightarrow \Lambda$ be the map

$$\bar{x}_1 v_1 + \cdots + \bar{x}_b v_b \mapsto x_1 v_1 + \cdots + x_b v_b,$$

and use the same symbol V to denote the map $V : E^m \rightarrow \Lambda^m$ that operates componentwise.

Let row vectors $c_1, c_2, \dots, c_{bk_i}$ be selected,

$$c_1 = (c_{1,1}, c_{1,2}, \dots, c_{1,bk_i}), \dots, c_{bk_i} = (c_{n,1}, c_{n,2}, \dots, c_{bk_i,bk_i}),$$

so that a typical codeword of C_i can be written $\bar{x}_1 c_1 + \cdots + \bar{x}_{bk_i} c_{bk_i}$, where each x_j is in $\{0, 1, 2, \dots, p-1\}$, and so that the rows $c_1, c_2, \dots, c_{bk_i}$ can be permuted with one another to form an upper triangular matrix, for $1 \leq i \leq t$.

The Construction E lattice is the mn -dimensional lattice L_t given as follows: Let

$$M_i = \left\{ x_1 D^{-i} V(c_1) + \cdots + x_{bk_i} D^{-i} V(c_{bk_i}) \mid x_1, \dots, x_{bk_i} \in \{0, 1, 2, \dots, p-1\} \right\}.$$

Let $L_0 = \Lambda^m$. For $1 \leq i \leq t$, we define

$$L_i = L_{i-1} + M_i = \{x + y \mid x \in L_{i-1}, y \in M_i\}.$$

Construction E produces a lattice in \mathbb{R}^{mn} whose determinant is

$$\frac{(\det \Lambda)^m}{\exp_p(b(k_1 + k_2 + \cdots + k_t))}$$

and that lattice-packs spheres of radius

$$(1/2) \min_{0 \leq j \leq t} \left(d |\det D|^{-j/n} \sqrt{d_j} \right).$$

E_8 AND THE LEECH LATTICE Λ_{24}

E_8 and Λ_{24} are anomalously dense and symmetrical lattice packings in \mathbb{R}^8 and \mathbb{R}^{24} , respectively. They have far more constructions than we can mention here.

Let L be the lattice $\{(x_1, \dots, x_8) \in \mathbb{Z}^8 \mid x_1 + \cdots + x_8 \text{ is even}\}$. Then E_8 is

$$L \cup \left(L + \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right) \right).$$

Alternatively, one gets E_8 by applying Construction A to the binary extended Hamming code $[8, 4, 4]$, which is the span over $GF(2)$ of the rows of this array:

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

When scaled so that $\det E_8 = 1$, it packs spheres of radius $\sqrt{1/2}$. Each sphere touches 240 others, and that is known to be the maximum number possible.

Our construction of the Leech lattice will be based on the extended Golay code

G_{24} , with parameters [24, 12, 8], which is the span over $GF(2)$ of the rows of this array:

1	1	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	1	1	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	1	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0
1	1	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0	0
1	1	1	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0
1	0	1	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Let 0 be the all-zeros vector in \mathbb{R}^{24} , and 1 the all-ones vector. Let v vary over G_{24} , and let x_{odd} and x_{even} vary over the points of \mathbb{Z}^{24} for which $\sum_{i=1}^{24} x_i$ is odd or even, respectively. Let $T_1 = \{0 + 2v + 4x_{\text{even}}\}$ and $T_2 = \{1 + 2v + 4x_{\text{odd}}\}$. Then the **Leech lattice** is

$$\Lambda_{24} = (T_1 \cup T_2)/\sqrt{8}.$$

When scaled so that $\det \Lambda_{24} = 1$, it packs spheres of radius 1. Each sphere touches 196,560 others, and that is the highest possible contact number. Its automorphism group modulo reflection through the origin is the finite simple sporadic group Co_0 of order $2^{21}3^95^47^211 \cdot 13 \cdot 23$.

SUPERBALLS

By applying Construction A to $[n, k, d, p, G]$ codes, where G is a rather general type of body called a *superball*, it is possible to get extremely dense packings of these bodies. In fact the density is always at least $2^{-n(1+o(1))}$, like the Minkowski-Hlawka bound. Often the density is much greater. Papers containing details on these matters include [Rus93], which contains further references.

Fix k , and let n be a multiple of k .

A **superball function** $f : \mathbb{R}^k \rightarrow \mathbb{R}$ is a function with the following four properties:

$$f(x) > 0 \quad \text{except that} \quad f(0) = 0;$$

$$f(x) = f(-x);$$

if $t > 0$ then there exists a nonsingular linear transformation A on \mathbb{R}^k such that

$$tf(x) = f(Ax)$$

holds identically in x ; and finally, if $0 \leq \theta \leq 1$ then

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y).$$

A **superball** is a body in \mathbb{R}^n given by

$$f(x_1, \dots, x_k) + f(x_{k+1}, \dots, x_{2k}) + \dots + f(x_{n-k+1}, \dots, x_n) \leq 1,$$

where f is a superball function. Let G_f denote that superball. In [Rus93] it was shown that

$$\delta_L(G_f) \geq \left(\frac{1}{2} \left(\sup_{A \in GL_n(\mathbf{R})} \frac{\int_{x \in \mathbf{R}^k} \exp(-f(Ax)) dV}{\sum_{x \in \mathbf{Z}^k} \exp(-f(Ax))} \right)^{1/k} \right)^{n(1+o(1))} \quad \text{as } n \rightarrow \infty.$$

The second author conjectures that this holds with equality. In the case $k = 1$ and $f(x) = x^2$, the conjecture is that $\delta_L(n) = 2^{-n(1+o(1))}$.

Finally, the first author conjectures that the “covering (or random) type” lower bounds presented in this chapter, such as the Varshamov-Gilbert bound for *binary* error-correcting codes, its analogue for spherical codes (the Shannon bound), and, consequently, the Minkowski-Hlawka bound, are asymptotically tight.

61.5 VERY RECENT DEVELOPMENTS

Just prior to publication, there were two major developments. One involved Musin’s claim of a proof of the kissing number for 3-spheres in \mathbb{R}^4 : $\tau(4) = 24$ [Mus03]. The other was the claim by H. Cohn and A. Kumar of a proof that the Leech lattice is the densest lattice in dimension 24. Both results were being checked as this edition of the Handbook went to press.

61.6 SOURCES AND RELATED MATERIAL

For basic results and further references on the geometry of numbers, see [Cas59, GL87, Min96]; for sphere packing, [CS99]; for packing and covering in general, [Dav64, Fej72, Rog64, Lev98]; for coding theory, [MS78, vL82, TV91, PH98].

RELATED CHAPTERS

- [Chapter 2: Packing and covering](#)
- [Chapter 7: Lattice points and lattice polytopes](#)
- [Chapter 62: Crystals and quasicrystals](#)

REFERENCES

- [Alo97] N. Alon. Packings with large minimum kissing number. *Discrete Math.*, 175:249–251, 1997.
- [ABV01] A. Ashikhmin, A. Barg, and S. Vlăduț. Linear codes with exponentially many light vectors. *J. Combin. Theory Ser. A*, 96:396–399, 2001.
- [Bal93] K.M. Ball. A lower bound for the optimal density of lattice packings. *Internat. Math. Res. Notices*, 10:217–221, 1993.
- [Cas59] J.W.S. Cassels. *An Introduction to the Geometry of Numbers*. Springer-Verlag, New York, 1959.

- [Cha46] T.W. Chaundy. The arithmetic minima of positive quadratic forms I. *Quart. J. Math.*, 17:166–192, 1946.
- [CS96] J.H. Conway and N.J.A. Sloane. The antipode construction for sphere packings. *Invent. Math.*, 123:309–313, 1996.
- [CS99] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*, 3rd edition. Springer-Verlag, New York, 1999.
- [Dav64] H. Davenport. Problems of packing and covering. *Rend. Sem. Mat. Univ. Politec. Torino*, 24:41–48, 1964/1965.
- [Del73] P. Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Res. Rep. Suppl.*, No. 10, 1973.
- [Dum95] I.I. Dumer. Nonbinary double-error-correcting codes designed by means of algebraic varieties, *IEEE Trans. Inform. Theory*, 41:1657–1666, 1995.
- [Fej72] L. Fejes Tóth. *Lagerungen in der Ebene auf der Kugel und im Raum*, 2nd edition. Volume 65 of *Grundlehren Math. Wiss.* Springer-Verlag, Berlin, 1972.
- [GL87] P.M. Gruber and C.G. Lekkerkerker. *Geometry of Numbers*. Elsevier, Amsterdam, 1987.
- [Hal00] T.C. Hales. Cannonballs and honeycombs. *Notices Amer. Math. Soc.*, 47:440–449, 2000.
- [Hil01] D. Hilbert. Mathematische Probleme. *Archiv. Math. Phys.*, 1:44–63, 1901.
- [Hla43] E. Hlawka. Zur Geometrie der Zahlen. *Math. Z.*, 49:285–312, 1943.
- [KL78] G.A. Kabatianski and V.I. Levenshtein. Bounds for packings on the sphere and in space (in Russian). *Problemy Peredachi Informatsii*, 14:3–25, 1978; English translation in *Problems Inform. Transmission*, 14:1–17, 1978.
- [KP88] G.A. Kabatianski and V.I. Panchenko. Packings and coverings of the Hamming space by balls of unit radius (in Russian). *Problemy Peredachi Informatsii*, 24:3–16, 1988; English translation in *Problems Inform. Transmission*, 24:261–272, 1988.
- [Lev79] V.I. Levenshtein. Bounds for packings in n -dimensional Euclidean space (in Russian). *Dokl. Akad. Nauk SSSR*, 245:1299–1303, 1979; English translation in *Soviet Math. Dokl.*, 20:417–421, 1979.
- [Lev83] V.I. Levenshtein. Bounds for packing in metric spaces and some of their applications (in Russian). *Problemi Kibernetiki*, 40:43–110, 1983.
- [Lev98] V.I. Levenshtein. Universal bounds for codes and designs. In V.S. Pless and W.C. Huffman, editors, *Handbook of Coding Theory*. Elsevier, Amsterdam, 1998.
- [Lin86] J.H. Lindsey, II. Sphere packing in R^3 . *Mathematika*, 33:137–147, 1986.
- [vL82] J.H. van Lint. *Introduction to Coding Theory*. Springer-Verlag, New York, 1982.
- [vL90] J.H. van Lint. Algebraic geometric codes. In D. Ray-Chaudhuri, editor, *Coding Theory and Design Theory I*. Springer-Verlag, New York, 1990.
- [MS78] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1978.
- [MRRW77] R.J. McEliece, E.R. Rodemich, H.C. Rumsey, and L.R. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inform. Theory*, 23:157–166, 1977.
- [Min96] H. Minkowski. *Geometrie der Zahlen I*. Teubner, Leipzig, 1896.
- [Min69] H. Minkowski. *Gesammelte Abhandlungen* (reprint). Chelsea, New York, 1969.

- [Mud93] D.J. Muder. A new bound on the local density of sphere packings. *Discrete Comput. Geom.*, 10:351–375, 1993.
- [Mus03] O.R. Musin. The kissing number in 4 dimensions. [arXiv:math.MG/0309430](https://arxiv.org/abs/math/0309430).
- [OS79] A.M. Odlyzko and N.J.A. Sloane. New bounds of the number of unit spheres that can touch a unit sphere in n dimensions. *J. Combin. Theory Ser. A.*, 26:210–214, 1979.
- [PH98] V.S. Pless and W.C. Huffman, editors. *Handbook of Coding Theory*. Elsevier, Amsterdam, 1998.
- [Ran55] R.A. Rankin. The closest packing of spherical caps in n dimensions. *Proc. Glasgow Math. Assoc.*, 2:139–144, 1955.
- [Rog64] C.A. Rogers. *Packing and Covering*. Cambridge Univ. Press, 1964.
- [Rus89] J.A. Rush. A lower bound on packing density. *Invent. Math.*, 98:499–509, 1989.
- [Rus93] J.A. Rush. A bound, and a conjecture, on the maximum lattice-packing density of a superball. *Mathematika*, 40:137–143, 1993.
- [Sam01] A. Samorodnitsky. On the optimum of Delsarte's linear program. *J. Combin. Theory Ser. A*, 96:261–287, 2001.
- [Sha59] C.E. Shannon. Probability of error for optimal codes in a Gaussian channel. *Bell System Tech. J.*, 38:611–656, 1959.
- [SAK⁺01] K.W. Shum, I. Aleshnikov, P.V. Kumar, H. Stichtenoth, and V. Deolalikar. A low-complexity algorithm for the construction of algebro-geometric codes better than the Gilbert-Varshamov bound. *IEEE Trans. Inform. Theory*, 47: 2225–2241, 2001.
- [Sid71] V.M. Sidel'nikov. On mutual correlation of sequences (in Russian). *Problemi Kibernetiki*, 24:15–42, 1971.
- [Tho] J.G. Thompson. Personal communication to N.J.A. Sloane.
- [Thu10] A. Thue. Über die dichteste Zusammenstellung von kongruenten Kreisen in einer Ebene. *Christiania Vidensk. Selsk. Skr.*, 1:1–9, 1910.
- [TV91] M.A. Tsfasman and S.G. Vlăduț. *Algebraic-Geometric Codes*. Kluwer, Dordrecht, 1991.
- [TVZ82] M.A. Tsfasman, S.G. Vlăduț, and T. Zink. Modular curves, Shimura curves, and Goppa codes better than the Varshamov-Gilbert bound. *Math. Nachr.*, 109:21–28, 1982.
- [Var95] A. Vardy. A new sphere packing in 20 dimensions. *Invent. Math.*, 121:119–133, 1995.

62 CRYSTALS AND QUASICRYSTALS

Marjorie Senechal

INTRODUCTION

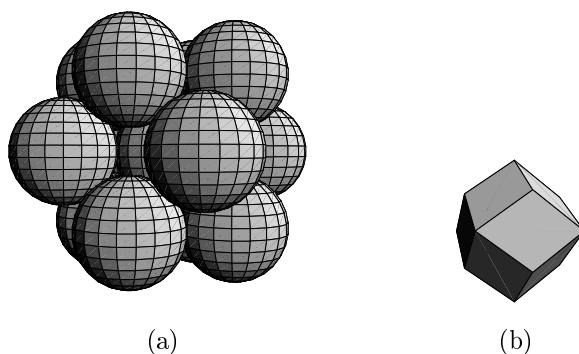
Mathematical crystallography is the branch of discrete geometry that deals with the structure and form of crystals. For over a century the field has been a meeting ground for polytopes, lattices, tilings, and groups. Today, stimulated both by developments internal to mathematics and by the discovery of quasicrystals, the subject is broadening rapidly, and modeling the geometry of crystals requires an ever-expanding mathematical toolbag. In Section 62.1 we survey the classical foundations of the subject; in Section 62.2 we indicate how these foundations are being redesigned to encompass recent developments. *We assume that the reader is familiar with the terminology and results of Chapter 3 of this Handbook.*

62.1 PERIODIC CRYSTALS

The geometrical study of crystals began when Johannes Kepler suggested that snowflakes were comprised of identical spheres arranged in what we now call cubic close-packing. Kepler also noted that if the spheres in such a packing were uniformly compressed, they would assume the forms of rhombic dodecahedra (Figure 62.1.1), and these dodecahedra would tile space. He thus demonstrated the duality between sphere-packing models and tiling models for crystal structure. This close relation between sphere packings and tilings, or more generally between point sets (the centers of the spheres in Kepler's case) and tilings, is exploited in mathematical crystallography to this day.

FIGURE 62.1.1

(a) *Cubic closest packing of spheres.* (b) *When the spheres are uniformly compressed they become space-filling rhombic dodecahedra.*



62.1.1 POINT-SET MODELS

From the middle of the nineteenth century until quite recently, “regular systems of points,” or unions of a finite number of them, have served as the abstract model for crystal structure. The study of crystal geometry amounted to the classification of these point sets by symmetry.

THE CLASSICAL THEORY

Let Γ be a discrete point set in \mathbb{E}^n .

GLOSSARY

Star (of a point $x \in \Gamma$): The configuration of line segments joining x to each of the other points of Γ .

Voronoi cell (of a point $x \in \Gamma$): The set $V(x)$ of points in \mathbb{E}^n that are at least as close to x as to any other point of Γ . (See also [Chapters 3](#) and [23](#).)

Voronoi tiling (associated with Γ): The tiling \mathcal{V}_Γ whose tiles are the Voronoi cells of the points of Γ .

Regular system of points: An infinite discrete point set such that the stars of all its points are congruent; equivalently, a discrete point set that is an orbit of an infinite group of isometries.

Crystallographic group: A group of isometries that acts transitively on a regular system of points. Crystallographic groups are discrete subgroups of the nonabelian group of Euclidean motions of \mathbb{E}^n .

Lattice (of dimension n): A discrete subgroup of \mathbb{R}^n , generated by n linearly independent translations.

Crystal (classical): The union of a finite number of orbits of a crystallographic group.

Point lattice: An orbit of a lattice.

MAIN RESULTS

Table 62.1.1 [BBN⁺78] gives the number of crystallographic groups in \mathbb{E}^2 , \mathbb{E}^3 , and \mathbb{E}^4 , up to isomorphism. For $n \geq 5$ the number of groups is not known.

TABLE 62.1.1 Crystallographic groups.

n	2	3	4
TYPES	17	219	4783

THEOREM 62.1.1 *Bieberbach's Theorem*

A regular system of points is a finite union of translates of congruent lattices (Figure 62.1.2); *the symmetry group G of a regular system of points is a product of a translation group T and a finite group of isometries, such that T is the maximal abelian subgroup of G .*

(See [Sen96] for a discussion of this theorem and for further references.)

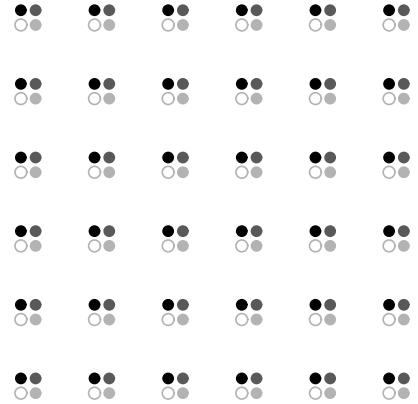


FIGURE 62.1.2

A regular system of points as a union of congruent lattices.



THEOREM 62.1.2 *The Crystallographic Restriction*

The only rotational symmetries possible for a regular system of points are those compatible with a lattice (of the same dimension).

Table 62.1.2 gives the possible orders m , $2 \leq m \leq 13$, of rotational symmetries of a regular system of points and the lowest dimension $d(m)$ in which they can occur. Five-fold rotations, as well as n -fold rotations with $n > 6$, are “forbidden” in \mathbb{E}^2 and \mathbb{E}^3 . (This table is easily computed from the formula given in [Sen96, p. 51].)

TABLE 62.1.2 *m-fold rotational symmetries.*

m	$d(m)$	m	$d(m)$	m	$d(m)$	m	$d(m)$
2	1	5	4	8	4	11	12
3	2	6	2	9	6	12	4
4	2	7	6	10	4	13	12

DELONE'S REFORMULATION OF THE CLASSICAL FOUNDATIONS

In the 1930s Delone, Aleksandrov, and Padurov reformulated the foundations of mathematical crystallography, replacing the regular systems of points with more general discrete point sets, which they called “ (r, R) -systems.”

GLOSSARY

(r, R) system: A set $\Lambda = \Lambda_{r,R}$ of points in \mathbb{E}^n that is uniformly discrete and relatively dense (r is the infimum of the distances between pairs of points of Λ , and every sphere of radius $\geq R$ contains at least one point of Λ).

Delone (or Delaunay) set: The modern term for an (r, R) system.

Finite type: A Delone set Λ is said to be of finite type if $\Lambda - \Lambda$ is a discrete closed set.

c-star (of a point x in a Delone set): The configuration of line segments joining x to the points of the Delone set that lie in $B(x, c)$, the ball with center x and radius c .

MAIN RESULTS

The Voronoi cell of any point x of a Delone set $\Lambda_{r,R}$ is contained in the ball $B(x, R)$; thus the cell is completely determined by $\Lambda \cap B(x, 2R)$. (This is an easy exercise.)

If an orbit of a group of isometries of \mathbb{E}^n is a Delone set, then the group is crystallographic and the Delone set is a regular system of points [DLS98].

A Delone set $\Lambda_{r,R}$ is of finite type if and only if it has a finite number of neighborhoods of radius $2R$, up to translation [Lag99].

THEOREM 62.1.3 *The Local Theorem (for point sets)* [DDSG76]

There is a real number k such that if all the $2Rk$ -stars of a Delone set $\Lambda_{r,R}$ are congruent, then Λ is a regular system of points.

(See also [Section 3.2](#).)

PROBLEM 62.1.4

Does the constant k in the Local Theorem depend only on the dimension n ?

62.1.2 TILING MODELS

Crystal growth is modular: beginning with a relatively tiny cluster of atoms, a crystal grows by the accretion of modules (atoms, molecules) to this “seed.” The position a module assumes on the growing crystal is assumed to be determined by local forces, as are subsequent rearrangements that may be required to minimize surface energy. In models of crystal structure consistent with this process, the modules are sometimes represented as spheres, but more commonly as space-filling polyhedra. In particular, it is convenient to think of a crystal as a tiling of space by congruent tiles. The tiles may be the crystallographer’s “unit cells,” the Voronoi cells of the crystal lattice, or stereohedra.

GLOSSARY

(Closed) unit cell (of a lattice in \mathbb{E}^n): The Minkowski sum of a set of n generating vectors of the lattice.

Zonotope: The Minkowski sum of an arbitrary number of line segments (or vectors).

n -parallelotope: A convex n -polytope that tiles \mathbb{E}^n by translation. (See Section 3.2.) Unit cells and Voronoi cells (of lattice points) are parallelotopes.

Stereohedron: The Voronoi cell of a point of a regular system of points. (A stereohedron is not necessarily a parallelotope.)

MAIN RESULTS

Table 62.1.3 gives the number of combinatorial types of n -parallelotopes in \mathbb{E}^2 , \mathbb{E}^3 , and \mathbb{E}^4 . (See [Sen96, p. 45].)

TABLE 62.1.3 n -parallelotopes.

n	2	3	4
TYPES	2	5	52

A 2-parallelotope is combinatorially equivalent to a quadrilateral or a hexagon. The 3-parallelotopes are, combinatorially, cubes, hexagonal prisms, truncated octahedra, rhombic dodecahedra, and the “elongated” rhombic dodecahedra (which have four hexagonal and eight rhombic faces). The 2-parallelotopes and 3-parallelotopes are zonotopes, but this is not generally true in higher dimensions.

Every 2-, 3-, and 4-parallelotope is an affine image of the Voronoi cell of a lattice in \mathbb{E}^2 , \mathbb{E}^3 , and \mathbb{E}^4 , respectively.

The number of combinatorial types of stereohedra in \mathbb{E}^n is bounded (see Section 3.2).

PROBLEM 62.1.5

Is every n -parallelotope an affine image of the Voronoi cell of some full rank lattice point in \mathbb{R}^n ?

Voronoi proved that the answer is “yes” if exactly $n + 1$ Voronoi cells meet at every vertex of the Voronoi tiling. The answer is also “yes” for zonotopes [Erd99] and for parallelotopes with $2(2^n - 1)$ faces (the maximal number in that dimension) [MRS95].

62.1.3 MODELING X-RAY DIFFRACTION

In 1912 the German physicist Max von Laue demonstrated the light-like nature of X-rays and the plausibility of a lattice structure for crystals by showing that crystals can serve as diffraction gratings for X-rays (this experiment also supported the existence of atoms). X-ray diffraction turned out to be the Rosetta stone that unlocked the solid state. Synthetic pharmaceuticals, electronics, and medical imaging are only three of the many fields of application that have resulted from this discovery.

X-ray diffraction is far-field (Fraunhofer) diffraction. This means that the distances from the X-ray source to the crystal and from the crystal to the photographic plate on which scattered intensities are recorded are sufficiently far that the scattering can be modeled by Fourier transformation [Cow86].

GLOSSARY

Dual lattice: If L is a lattice in \mathbb{E}^n , its dual lattice L^* is the group of vectors $\vec{y} \in \mathbb{E}^n$ such that $\vec{y} \cdot \vec{x} \in \mathbb{Z}$ for every $\vec{x} \in L$; here \cdot denotes the usual scalar product.

Dirac delta “function” at x : Intuitively, the generalized function δ_x that assigns unit mass to the point $x \in \mathbb{E}^n$ and vanishes at all other points.

MAIN RESULTS

Assume, for simplicity, that our regular system of points is a point lattice. We associate to the corresponding lattice L the generalized function

$$\rho(x) = \sum_{x_n \in L} \delta_{x_n}(x); \quad (62.1.1)$$

its Fourier transform is the generalized function

$$\hat{\rho}(s) = \sum_{x_n \in L} \exp(-2\pi i x_n \cdot s), \quad (62.1.2)$$

where $s \in \mathbb{E}^n$. The diffraction pattern that we observe (on a photographic plate) when X-rays are passed through this “crystal” is a density map of the crystal’s “intensity function” (the Fourier transform of the autocorrelation $\rho(x) * \overline{\rho(-x)}$). The “Wiener diagram” below [Sen96], in which \dagger denotes Fourier transformation, describes the relationship between the crystal and the observed intensities.

$$\begin{array}{ccc} \rho(x) & \xrightarrow{\text{autocorrelation}} & \rho(x) * \overline{\rho(-x)} \\ \dagger & & \dagger \\ \hat{\rho}(s) & \xrightarrow{\text{squaring}} & |\hat{\rho}(s)|^2 \end{array}$$

The task of the crystallographer is to deduce $\rho(x)$ from the intensity function, a task greatly complicated by the fact that the intensity is real while $\hat{\rho}(s)$ is complex.

This diagram is widely used in crystallography for heuristic purposes, although it is not valid (in any theory of generalized functions) because convolution and multiplication are not defined for infinite sums of deltas. Nevertheless, there is a sense in which it gives correct information [Hof95]. In particular, sharp bright spots in the diffraction pattern correspond to delta functions in the Fourier transform in the case of periodic point sets (and also in the case of model sets, discussed in Section 62.2 below). The Poisson summation formula (see [Sen96]) states, in effect, that the diffraction pattern of a point lattice is a set of sharp bright spots at the points of its dual point lattice.

THEOREM 62.1.6 Poisson Summation Formula

Let L and L^* be dual lattices, and let $\rho(x)$ be as in (62.1.1) above. Then (62.1.2) can be written in the form

$$\hat{\rho}(s) = \sum_{s_n \in L^*} \delta_{s_n}(s). \quad (62.1.3)$$

62.2 GENERALIZED CRYSTALS AND QUASICRYSTALS

After the discovery of X-ray diffraction in 1912, it was unquestioningly accepted that a crystal is a solid with a periodic atomic structure. Only a periodic structure, it was reasoned, could produce diffraction patterns with sharp bright spots, because—roughly speaking—the spots indicate the repetition, throughout the crystal’s atomic pattern, of congruent c -stars for all $c > 0$. The “long-range order” created by this repetition, it was assumed, must be periodic. But this classical model began to be questioned in the 1970s when it was found that the structures of so-called modulated crystals could not be accounted for by three-dimensional periodicity. The paradigm that had reigned since Laue’s experiment collapsed completely with the discovery, in the early 1980s, of crystals with “forbidden” icosahedral symmetry. Today it is widely agreed that both periodic and nonperiodic crystals exist, but the structure of nonperiodic crystals is still not fully understood. Rather than repeat the mistake of the past by again defining a crystal in terms of some a priori concept of its structure, the Commission on Aperiodic Crystals of the International Union of Crystallography proposed as a working definition: *a crystal is a solid with an essentially discrete diffraction pattern.*

To put this into mathematical language, we follow the periodic model by associating a sum of Dirac deltas to a Delone set Λ , one delta at each point, and computing the Fourier transform of the autocorrelation (when the autocorrelation exists). This transform is a measure, called the spectrum of Λ ; the spectrum can be uniquely decomposed into a sum of discrete and continuous measures. The discrete component of the spectrum is itself a countable sum of weighted Dirac deltas, located at a set of points that we will call Λ_d (when Λ is a lattice, $\Lambda_d = \Lambda^*$). We always have $0 \in \Lambda_d$; if $\Lambda_d \neq \{0\}$, it is said to be nontrivial. (The set Λ_d need not be discrete as a point set; in general it will be everywhere dense.)

Definition: A (*generalized*) *crystal* is a Delone set Λ with nontrivial Λ_d . A *quasicrystal* is a generalized crystal whose intensity function is invariant under a rotational symmetry forbidden by the Crystallographic Restriction. The *symmetry group* (of a quasicrystal) is the group of isometries under which the intensity function is invariant.

Below, we describe some generalizations of the notions of regular systems of points and stereohedra. It must be emphasized that at this early stage, all definitions are subject to change, and very few theorems have been proved in satisfactory generality.

62.2.1 POINT-SET MODELS

A point-set model for a generalized crystal is a suitable generalization of a regular system of points, but there is no agreement yet on what “suitable” should mean. However, most models assume that the point set is a Delone set satisfying additional conditions, for example as in the definition above. Classification by symmetry group is replaced by local isomorphism classes.

GLOSSARY

c-atlas: The set of congruence classes of *c*-stars of the points of a Delone set Λ .

Repetitive point set: A Delone set Λ such that the stars of every *c*-atlas are relatively dense in Λ (i.e., for each such star s there is an $R_s > 0$ such that every ball of radius R_s contains a copy of s).

Local isomorphism class: Two Delone sets in \mathbb{E}^n belong to the same local isomorphism class if every bounded configuration of each also occurs in the other.

Inflation symmetry: A Delone set Λ is said to possess inflation symmetry if there is a $\lambda > 1$ such that $\lambda\Lambda \subset \Lambda$.

X-ray diffraction patterns of real crystals suggest that point-set models for generalized crystals in \mathbb{E}^k should be subsets of \mathbb{Z} -modules of rank $m \geq k$. A \mathbb{Z} -module whose rank m is greater than its dimension k may be everywhere dense. One way to select the points of the crystal is by means of an “acceptance domain” or “**window**.” A crystal obtained in this way is called a *model set* [Mey95].

Model set: Let L be a lattice of rank $m = k + n$ in \mathbb{E}^m ; let p_{\parallel} and p_{\perp} be the orthogonal projections into a k -dimensional subspace $\mathcal{E} = \mathbb{E}^k$ and its orthogonal complement $\mathcal{E}^\perp = \mathbb{E}^n$, respectively. Assume that p_{\parallel} , restricted to L , is one-one and that $p_{\perp}(L)$ is everywhere dense in \mathbb{E}^n , and let Ω be a bounded subset of \mathbb{E}^n with nonempty interior. Then the set

$$\Lambda(\Omega) = \{p_{\parallel}(x) \mid x \in L, p_{\perp}(x) \in \Omega\} \quad (62.2.1)$$

is called a model set. When Ω is a translate of the projection of the Voronoi cell of the lattice into the orthogonal space, the window is said to be **canonical**. Note: model sets can be defined in greater generality than is done here [Moo00].

The ingredients for a one-dimensional model set are shown in Figure 62.2.1, in which $m = 2$, $n = k = 1$, and L is a square lattice. The subspace \mathcal{E} is the solid line of positive slope; the window Ω is the thick line segment in \mathcal{E}^\perp . A lattice point x is projected into \mathcal{E} if and only if $p_{\perp}(x) \in \Omega$ (alternatively, if and only if x lies in the cylinder bounded by the dotted lines). Note that the window in Figure 62.2.1 is *not* canonical.

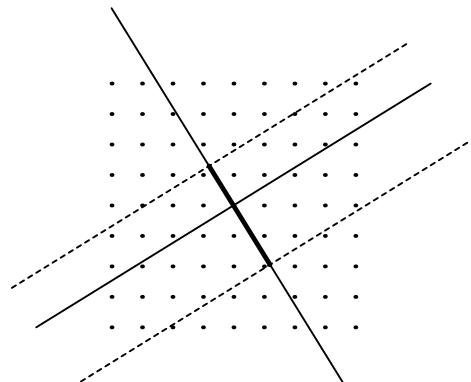


FIGURE 62.2.1

Ingredients for a one-dimensional model set. The subspace \mathcal{E} is the solid line; the window Ω is the thick line segment in \mathcal{E}^\perp .

Meyer set: A Meyer set is a Delone set Λ such that $\Lambda - \Lambda$ is also a Delone set.

Poisson comb: A crystal with purely discrete spectrum.

ϵ -dual (of a Delone set Λ): $\Lambda_\epsilon^* = \{y \in \mathbb{E}^n \mid |\exp(2\pi iy \cdot \lambda) - 1| \leq \epsilon, \forall \lambda \in \Lambda\}$.

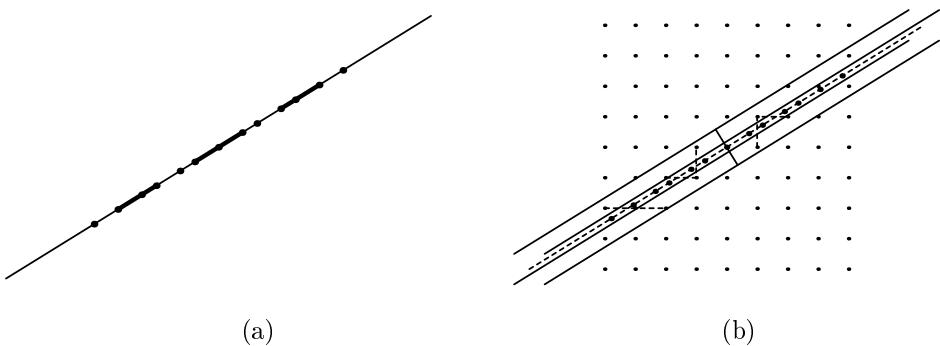
MAIN RESULTS

Model sets are repetitive Delone sets. By translating Ω we get an infinite family of model sets in the same local isomorphism class. If the subspace \mathcal{E} contains no points of the dual lattice L^* , then the model set is nonperiodic, i.e., it is not invariant under any translation [Sen96, Moo97].

When the window Ω is a translate of the projection, into \mathcal{E}^\perp , of the Voronoi cell of the lattice, the relative frequencies of the r -stars of $\Lambda(\Omega)$ are determined by the location of the corresponding points $p_\perp(x)$ in the window (see Figure 62.2.2) [Sen96].

FIGURE 62.2.2

Every point in a one-dimensional model set is the second point in a three-point configuration or star. (a) In this example, there are three translations classes of such stars. (b) Each star is characterized by the interval, in the window Ω , into which the lattice point corresponding to its “center” projects. (\mathcal{E} is the dotted line; the triples of lattice points projecting to the stars are also indicated by dotted lines.)



Every model set is a Meyer set; conversely, every Meyer set is a subset of a set of the form $\Lambda(\Omega) + F$, where $\Lambda(\Omega)$ is a model set and F is finite [Mey95].

Λ is a Meyer set if and only if $\Lambda - \Lambda \subseteq \Lambda + F$, where F is finite [Lag96].

If Λ is a finitely generated Delone set with inflation symmetry, then λ must be an algebraic integer. If Λ is of finite type, then in addition all algebraic conjugates λ' satisfy $|\lambda'| \leq \lambda$. If Λ is a Meyer set, then for all algebraic conjugates λ' , $|\lambda'| \leq 1$ [Lag99].

A Delone set Λ is a Meyer set if and only if for every $\epsilon > 0$, Λ_ϵ^* is relatively dense in \mathbb{E}^n [Moo97].

Every (suitably) repetitive Delone set is a Poisson comb [LP98].

For accounts of other recent results, see [BM03].

OPEN PROBLEMS

CONJECTURE 62.2.1

The converse of the last statement above is also true.

The most important open problem is the one posed by the discovery of nonperiodic crystals:

PROBLEM 62.2.2

What are necessary and sufficient conditions for a discrete point set to be a crystal according to the new definition? (It is not necessary that the set be Delone!)

Partial results have also been obtained [Lag00, LMS03] for the special case of Poisson combs.

PROBLEM 62.2.3

Is there an analogue of the Local Theorem for generalized crystals?

62.2.2 TILING MODELS

Tiling models are useful in the theory of generalized crystals for precisely the same reasons they are useful in the classical periodic case: they give us a clearer picture of how space is partitioned than point-set models do, and they may help us to understand the growth of crystals. Covering models have been proposed as well (see, e.g., [SJS⁺98]) but we will not discuss them here.

Which tilings are appropriate models for generalized crystals? High-resolution electron micrographs of many crystal structures can be interpreted meaningfully as tilings; often these tilings appear to be hierarchical (see [Section 3.4](#)). The hierarchical structure may play a role in crystal formation or stability. Some of the tilings can be derived by the projection method (see below), for which there does not appear to be a physical interpretation. However, the projection method is of great theoretical value.

CANONICALLY PROJECTED TILINGS

Canonically projected tilings are closely related to the model sets defined in [Section 62.2.1](#).

GLOSSARY

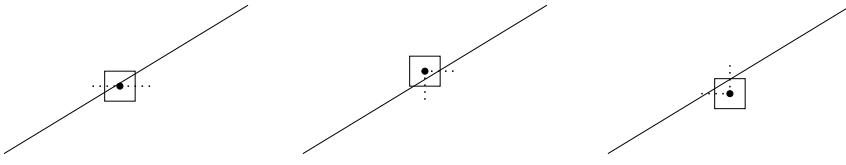
Canonical projection method for tilings: Let L be a lattice in \mathbb{E}^m , \mathcal{E} a k -dimensional subspace, and $\vec{\gamma} \in \mathbb{E}^m$. Let \mathcal{V} be the Voronoi tiling associated with L , and \mathcal{D} the dual Delone tiling (see [Section 3.1](#)). The canonical projection method for tilings projects, onto \mathcal{E} , the $(n-k)$ -dimensional faces of \mathcal{D} that correspond, under duality, to the k -dimensional faces of \mathcal{V} that are cut by $\mathcal{E} + \vec{\gamma}$ ([Figure 62.2.3](#)). Thus, if \mathcal{E} meets the interior of a Voronoi cell $V(x)$ (dimension n), we project x onto \mathcal{E} : x is the vertex (dimension 0) of the Delone tiling that corresponds to $V(x)$ in the duality. The vector $\vec{\gamma}$ is the **shift vector** for the projection.

Canonically projected tiling: A tiling that can be constructed by the canonical projection method.

Note: Some authors require “canonical” to mean, in addition to the above, that L is the standard integer lattice.

FIGURE 62.2.3

The line \mathcal{E} cuts a subset of the one- and two-dimensional faces of the Voronoi tiling (Voronoi cells are indicated by squares); we project the corresponding one- and zero-dimensional faces of the Delone tiling (dotted line segments and their endpoints).



MAIN RESULTS

Let $\mathcal{E} + \vec{\gamma}$ be a translate of a k -dimensional subspace of \mathbb{E}^n , let L be an n -dimensional lattice with Voronoi tiling \mathcal{V} , and let $*$ be the dual map. Denote the set of faces of \mathcal{V} that have nonempty intersection with \mathcal{E} by $\mathcal{V} \wedge \mathcal{E}_\gamma$, and let p_{\parallel} be as above. Then [Sch93]

$$(\mathcal{V} \cap \mathcal{E}_\gamma)^* = p_{\parallel}((\mathcal{V} \wedge \mathcal{E}_\gamma)^*). \quad (62.2.2)$$

Some of the best known projected nonperiodic tilings are listed in Table 62.2.1 (see [Sen96]). In all four cases, the lattice L is the standard integer lattice and the window Ω is a projection of a hypercube (the Voronoi cell of L). The vector $\vec{\gamma}$ is chosen so that $\mathcal{E} + \vec{\gamma}$ does not intersect any faces of \mathcal{V} of dimension less than $n - k$ (thus only a subset of the faces of the Delone tiling \mathcal{D} of dimensions $0, 1, \dots, k$ will be projected). The first three tilings are hierarchical; by an unpublished “folk theorem,” the fourth is, too. \mathcal{E} is a subspace that is stable under some finite rotation group; the rotational symmetry reappears in some of the bounded configurations of the tiling, and also in its diffraction pattern.

TABLE 62.2.1 Canonically projected nonperiodic tilings.

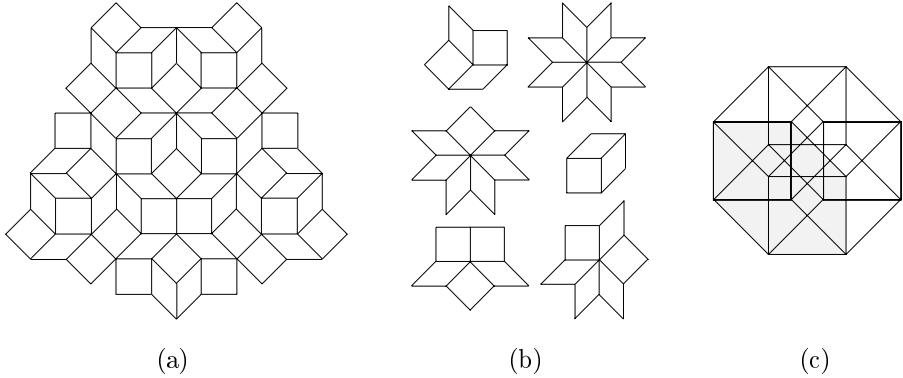
TILING FAMILY	L	\mathcal{E}
Fibonacci tiling	I_2	line with slope $1/\tau$ ($\tau = (1 + \sqrt{5})/2$)
Ammann-Beenker tiling	I_4	plane stable under 8-fold rotation
Generalized Penrose tiling	I_5	plane stable under 5-fold rotation
Penrose 3D tiling	I_6	3-space stable under icosahedral rotation group

The famous Penrose tilings of \mathbb{E}^2 (by rhombs) are precisely those generalized Penrose tilings defined by the flats $\mathcal{E} + \vec{\gamma}$, $\vec{\gamma} \in \mathbb{E}^5$ such that $\vec{\gamma} \cdot \vec{w} \equiv \frac{1}{2} \pmod{1}$, where $\vec{w} = (1, 1, 1, 1, 1)$.

The relative frequencies of the vertex stars of a canonically projected tiling are determined by the window: they are the ratios of volumes of the intersections of the projected faces of $V(0)$ (see Figures 62.2.2 and 62.2.4, and also [Sen96]).

FIGURE 62.2.4

The Ammann-Beenker tiling. (a) A portion of the tiling. (b) The six vertex stars. (c) The projected faces of the 4-cube are hexagons that decompose the window Ω into cells. There are six congruence classes of cells, corresponding to the six classes of stars. (A star of j tiles, $j = 3, \dots, 8$, corresponds to the intersection of the projections of j hexagons.)



THE MULTIGRID METHOD

The multigrid method is an interesting variant of the canonical projection method for tilings. In this version, the tiling is constructed as a dual of an n -grid, which is a superposition of n grids. For special choices of the grids and grid star, the n -grid is precisely the intersection $\mathcal{E}_\gamma \cap \mathcal{V}$ in (62.2.2); in these cases the multigrid and the canonical projection methods produce the same families of tilings. The multigrid method is, however, less studied [Sen97].

GLOSSARY

Grid: A countably infinite family of equispaced parallel ($(k-1)$ -dimensional) hyperplanes in \mathbb{E}^k .

Grid vector: A vector orthogonal to the grid whose length is the distance between adjacent hyperplanes of the grid.

n -grid (also **multigrid**): A union of n grids (in \mathbb{E}^k).

Grid star: The set of grid vectors of an n -grid.

Shift vector (for n -grids in \mathbb{E}^k): We think of the n grids as initially passing through the origin, and then shift them so that at most k grids pass through a single point. The shift vector $\vec{\gamma}$ is the n -tuple of the shifts away from the origin; if at most k hyperplanes of the n -grid meet in any point, $\vec{\gamma}$ is said to be *regular*.

Note: We use the same symbol, $\vec{\gamma}$, for shift vectors and for translations of \mathcal{E} to emphasize the fact that they play precisely the same role in the theory.

Pentagrid: A 5-grid in \mathbb{E}^2 whose star consists of unit vectors pointing from the center to the vertices of a regular pentagon (Figure 62.2.5). A pentagrid is said to be regular if its shift vector is regular.

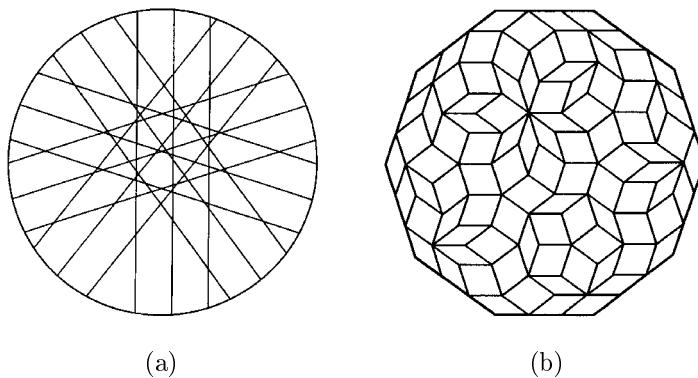
n -grid dual: A tiling dual to an n -grid whose edges are parallel to the vectors of the grid star.

MAIN RESULTS

Many of the main results in the subsection on Canonically Projected Tilings above can be reinterpreted in the language of multigrids and thus derived by the multigrid method.

FIGURE 62.2.5

A portion of a pentagrid (a) and the corresponding patch of a generalized Penrose tiling (b).



HIERARCHICAL TILINGS

These tilings are discussed in Section 3.4.

OPEN PROBLEMS

There is a large class of open problems concerned with the generality of these tiling construction methods and the relations among them.

PROBLEM 62.2.4

Which hierarchical tilings are projected tilings, and vice versa?

PROBLEM 62.2.5

Every tiling of \mathbb{E}^k by zonotopes has a pseudogrid dual (a grid in which the hyperplanes are replaced by pseudohyperplanes—see Chapter 6); which of these pseudogrids are stretchable, and how is this related to the question of whether or not the tiling is a crystal?

PROBLEM 62.2.6

Which tilings can be lifted to a surface (not necessarily contained in a cylinder) of faces of a lattice Delone complex in some higher-dimensional space?

62.2.3 MODELING CRYSTAL “GROWTH”

The classical notion of modeling crystal growth by tilings can be carried over to the more general setting, but now we want the tilings to be nonperiodic. Nonperiodic

tilings can be constructed hierarchically (see [Section 3.4](#)), by projection, and by many other methods. However, if we require that the nonperiodicity be *forced* by “matching rules” of some sort, then there are fewer possibilities. There has been a great deal of progress in matching rule theory during the past decade, but many important problems are still open.

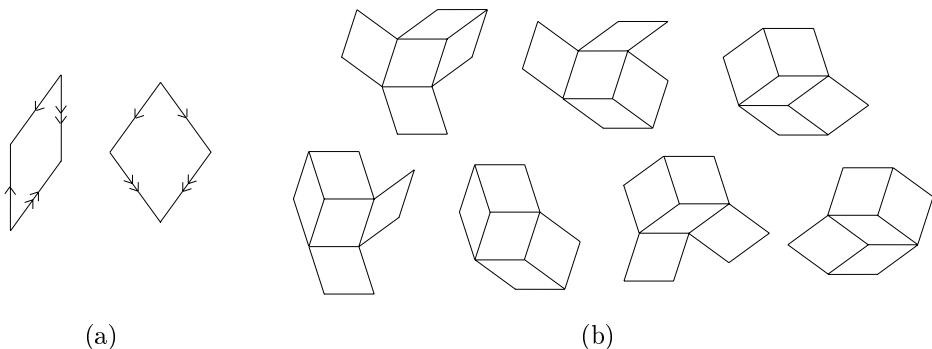
GLOSSARY

Aperiodic prototile set: A set of prototiles that admits only nonperiodic tilings (see [Chapter 3](#)). Some prototiles of aperiodic sets are simple shapes (with or without markings on them), but others can be quite complicated.

Matching rule (for a set of prototiles): A finite atlas, of some finite radius, of allowed configurations of marked or unmarked prototiles, such as face pairs, vertex stars, or face coronas (for definitions, see Chapter 3).

FIGURE 62.2.6

The matching rules for the Penrose tilings. (a) The tiles can be marked, as shown here; the rule is that both the type (double or single) and direction of the arrows must match. (b) The configurations shown here constitute another (equivalent) matching rule for the Penrose tiles: each tile must be matched to four other tiles in one of these ways.



Definition: A matching rule (for an aperiodic set of prototiles) is:
perfect, if it enforces nonperiodicity and repetitivity, and defines a single local isomorphism class;
strong, if it enforces nonperiodicity and repetitivity, but admits more than one local isomorphism class;
weak, if it enforces nonperiodicity but not repetitivity.

To prove that a prototile set is aperiodic, one must exhibit a matching rule that is at least weak.

There does not seem to be a name for matching rules that force periodic structures, but such rules do exist for certain prototile sets. For example, a tiling by squares will be periodic if we insist on a single vertex star, four squares meeting at a vertex. See also the classification of isohedral tilings in [GS87].

In cases when it is useful to distinguish between the prototiles and their shapes (for example, between marked and unmarked prototiles), it is helpful to make the following further distinctions: a matching rule, if it exists, is said to be *local* if a

finite atlas of configurations of unmarked tiles suffices to characterize the matching rule (as for the Penrose tiles in [Figure 62.2.6](#) above), and *nonlocal* if the tiles of the atlas must be decorated to characterize the rule. The matching rules for the Ammann-Beenker tiling (see [GS87] and [Figure 62.2.4](#)) are nonlocal.

MAIN RESULTS

Given any aperiodic prototile set in \mathbb{E}^2 , it is possible to construct a region homeomorphic to an annulus whose interior cannot be tiled with those prototiles [DS95]. (This property, well-known empirically to anyone who has ever played with Penrose tiles, is thus completely general; it follows that untileable “holes” in Penrose and other aperiodic tilings cannot be avoided by strengthening their matching rules.)

The proofs of the following results rely heavily on various theorems in discrete geometry, for example theorems of Helly type.

The atlas of face coronas of the Penrose tilings ([Figure 62.2.6\(b\)](#)) is a perfect local matching rule [deB96]; the Ammann-Beenker “octagonal” tiling does not have a local matching rule of any radius [Bur88].

Those generalized Penrose tilings for which $\vec{\gamma} \cdot \vec{w} \in \frac{1}{2} + \mathbb{Z}[\tau]$ are in the same mutually locally derivable class as the Penrose tilings, and hence are self-similar and have perfect, local, matching rules [Le97].

Nonlocal rules exist for all canonically projected tilings for which L is the integer lattice, $k = n$, and \mathcal{E} is quadratic [LP93], and for all generalized Penrose tilings such that $\vec{\gamma} \cdot \vec{w} \in Q[\tau]$ [Le95].

OPEN PROBLEMS

Matching rules have been found for a large number of hierarchical tilings; in most cases, they are nonlocal. Do matching rules exist for all hierarchical tilings? Which are local and which are not?

Which tilings constructed by projection can be equipped with matching rules? Which are local and which are not?

62.3 SOURCES AND RELATED MATERIAL

SURVEYS

The following useful surveys contain a wealth of material, much of it beyond the scope of this chapter.

[DS91]: Essays, mostly by physicists, on various aspects of quasicrystals and quasicrystal models.

[Jar89]: A collection of introductory essays on tiling models for quasicrystals.

[AG95]: Proceedings of the conference “Beyond Quasicrystals” held in Les Houches, France, March, 1994.

[Sen96]: A monograph devoted to the geometry of quasicrystal models.

[Moo97]: Proceedings of the NATO Advanced Study Institute “Mathematics of Aperiodic Order,” held in Waterloo, Canada, August, 1995.

[BM00]: This volume is 2000 state-of-the art.

RELATED CHAPTERS

[Chapter 3: Tilings](#)

[Chapter 6: Oriented Matroids](#)

[Chapter 16: Basic properties of convex polytopes](#)

[Chapter 19: Symmetry of polytopes and polyhedra](#)

[Chapter 23: Voronoi diagrams and Delaunay triangulations](#)

[Chapter 61: Sphere packing and coding theory](#)

REFERENCES

- [AG95] F. Axel and D. Gratias, editors. *Beyond Quasicrystals*. Collection du Centre de Physique des Houches, Editions de Physique, Springer-Verlag, Berlin, 1995.
- [BM00] M. Baake and R. Moody, editors. *Directions in Mathematical Quasicrystals*. Volume 13 of CRM Monograph Series, Amer. Math. Soc., Providence, 2000.
- [BM03] M. Baake and R. Moody. Pure point diffraction. Preprint.
- [BBN⁺78] H. Brown, R. Bülow, J. Neubüser, H. Wondratschek, and H. Zassenhaus. *Crystallographic Groups of Four-dimensional Space*. Wiley, New York, 1978.
- [Bur88] S.E. Burkov. Absence of weak local rules for the planar quasicrystalline tiling with 8-fold symmetry. *Comm. Math. Phys.*, 119:667–675, 1988.
- [Cow86] J.M. Cowley. *Diffraction Physics*. North Holland, Amsterdam, 1986.
- [DDSG76] B. Delone, N. Dolbilin, M. Shtogrin, and R. Galiulin. A local test for the regularity of a system of points. *Dokl. Akad. Nauk. SSSR*, 227:19–21, 1976. English translation: *Soviet Math. Dokl.*, 17:319–322, 1976.
- [deB96] N.G. de Bruijn. Remarks on Penrose tilings. In R.L. Graham and J. Nesetril, editors, *The Mathematics of Paul Erdős*, Volume 2. Springer-Verlag, Berlin, 1996, pages 264–283.
- [DS91] D. DiVincenzo and P.J. Steinhardt. *Quasicrystals: The State of the Art*. World Scientific, Singapore, 1991.
- [DLS98] N. Dolbilin, J. Lagarias, and M. Senechal. Multiregular point systems. *Discrete Comput. Geom.*, 20:477–498, 1998.
- [DS95] S. Dworkin and J.I. Shieh. Deceptions in quasicrystal growth. *Comm. Math. Phys.*, 168:337–352, 1995.
- [Erd99] R. Erdahl. Dicings, zonotopes, and Voronoi’s conjecture on parallelohedra. *European J. Combin.*, 20:527–549, 1999.
- [GS87] B. Grünbaum and G.C. Shephard. *Tilings and Patterns*. Freeman, New York, 1987.
- [Hof95] A. Hof. On diffraction by aperiodic structures. *Comm. Math. Phys.*, 169:25–43, 1995.
- [Jar89] M.V. Jaric, editor. *Introduction to the Mathematics of Quasicrystals*. Academic Press, San Diego, 1989.

- [Lag96] J. Lagarias. Meyer's concept of quasicrystal and quasiregular sets. *Comm. Math. Phys.*, 179:365–376, 1996.
- [Lag99] J. Lagarias. Geometric models for quasicrystals I. Delone sets of finite type. *Discrete Comput. Geom.*, 21:161–191, 1999.
- [Lag00] J. Lagarias. Mathematical quasicrystals and the problem of diffraction. In [BM00].
- [LP98] J. Lagarias and P. Pleasants. Repetitive Delone sets and perfect quasicrystals. Preprint, 1998.
- [Le95] T.Q.T. Le. Local rules for pentagonal quasicrystals. *Discrete Comput. Geom.*, 14:31–70, 1995.
- [Le97] T.Q.T. Le. Local rules for quasiperiodic tilings. In [M0097].
- [LP93] T.Q.T. Le and S. Piunikhin. Local rules for multidimensional quasicrystals. *Diff. Geom. Appl.*, 5:13–31, 1993.
- [LMS03] J.-Y. Lee, R. Moody, and B. Solomyak. Consequences of pure point diffraction spectra for multiset substitution systems. *Discrete Comput. Geom.*, 29:525–560, 2003.
- [Mey95] Y. Meyer. Quasicrystals, Diophantine approximation and algebraic numbers. In F. Axel and D. Gratias, editors, *Beyond Quasicrystals*, pages 3–16. Collection du Centre de Physique des Houches, Les Éditions de Physique, Springer-Verlag, Berlin, 1995.
- [MRS95] L. Michel, S.S. Ryshkov, and M. Senechal. An extension of Voronoï's theorem on primitive parallelotopes. *European J. Combin.*, 16:59–63, 1995.
- [Moo95] R. Moody. Meyer sets and the finite generation of quasicrystals. In B. Gruber, editor, *Symmetries in Science*. Plenum, New York, 1995.
- [Moo97] R. Moody. *The Mathematics of Long-Range Aperiodic Order*. Volume 489 of NATO Advanced Science Institutes Series C, Kluwer, Dordrecht, 1997.
- [Moo00] R. Moody. Model sets: a survey. In F. Axel and J.-P. Gazeau, editors, *From Quasicrystals to More Complex Systems*, Les Éditions de Physique, Springer-Verlag, Berlin, 2000.
- [Mos95] R. Mosseri. Random tilings. In F. Axel and D. Gratias, editors, *Beyond Quasicrystals*, pages 335–354. Collection du Centre de Physique des Houches, Les Éditions de Physique, Springer-Verlag, Berlin, 1995.
- [Rad94] C. Radin. The pinwheel tilings of the plane. *Ann. of Math.*, 139:661–702, 1994.
- [Sch93] M. Schlottmann. Periodic and quasi-periodic Laguerre tilings. *Internat. J. Modern Phys. B*, 7:1351–1363, 1993.
- [Sen96] M. Senechal. *Quasicrystals and Geometry*. Paperback edition, Cambridge University Press, 1996.
- [Sen97] M. Senechal. A critique of the projection method. In [M0097].
- [SJS⁺98] P.J. Steinhardt, H.-C. Jeong, K. Saitoh, M. Tanaka, E. Abe, and A.P. Tsai. Experimental verification of the quasi-uni-cell model of quasicrystal structure. *Nature*, 396:55–57, 1998.

63 BIOLOGICAL APPLICATIONS OF COMPUTATIONAL TOPOLOGY

Herbert Edelsbrunner

INTRODUCTION

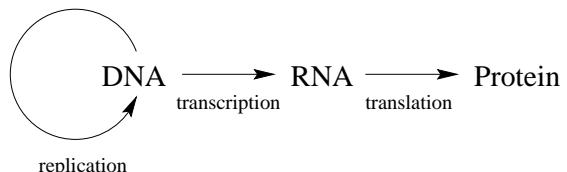
Structural molecular biology is a relatively recent application area for computational geometry and topology, but one with enormous potential. We currently observe a bipartition of computational research in this field: the *bioinformatics* branch focuses on strings, which are abstractions of the hereditary information stored in the DNA of living organisms, while the *molecular simulation* branch studies organic molecules in their natural three-dimensional habitat. Perhaps it is not surprising that the application of numerical algorithms is significantly more developed than that of geometric algorithms. One of the goals of this chapter is to raise the general consciousness about the importance of geometric methods in elucidating the mysterious foundations of our very existence. Another goal is the broadening of what we consider a geometric algorithm. There is plenty of valuable no-man's-land between combinatorial and numerical algorithms, and it seems opportune to explore this land with a computational-geometric frame of mind.

63.1 BIOMOLECULES

GLOSSARY

Central dogma: The proven claim that proteins are created in two steps by transcribing genes to RNA and translating RNA to protein.

FIGURE 63.1.1
The DNA gets replicated as a whole. Pieces of DNA referred to as genes are transcribed into pieces of RNA, which are then translated into proteins.



DNA: Deoxyribonucleic acid. The material that carries hereditary information. A double-stranded helix that encodes information into two antiparallel sequences of nucleotides.

Replication: Process in which the two strands of DNA are separated and both strands are complemented to form new double strands.

Genome: Complete set of genetic material of a living organism. For humans, it is divided into twenty-three *chromosomes*, each a long double strand of DNA.

Gene: Subsequence of DNA capable of being transcribed to produce a functional RNA molecule.

Transcription: Process in which the two strands of DNA are locally separated and one strand is copied to a piece of RNA.

RNA: Ribonucleic Acid. A single-stranded structure that is chemically almost identical to DNA.

Translation: Process in which a strand of RNA is read by the *ribosome* and translated into a protein.

Protein: A linear sequence of amino acids connected by peptide bonds.

Amino acid: Consists of a central carbon atom (C_α) linked to an amino group, a carboxyl group, one hydrogen atom, and a side chain. A *residue* is an amino acid whose $CC_\alpha N$ sequence is linked into the polypeptide chain of a protein.

Protein backbone: Polypeptide chain consisting of repeated $CC_\alpha N$ units. The bond between N and C is rigid, but the bonds connecting C_α to C and C α to N can be rotated around the connecting edges.

Protein folding: Process in which a polypeptide chain folds up to a usually globular shape that is characteristic for the type of protein.

FROM DNA TO PROTEIN

Organic life is based on a surprisingly small number of molecule types. Most prominently, we have *DNA*, *RNA*, and *protein*. Each of them has the simple structure of a linear sequence consisting of a chain or backbone with attached side chains. DNA and RNA each uses an alphabet of only four nucleotides, while proteins use an alphabet of twenty amino acids. As discovered by Watson and Crick [WC53], the natural form of DNA consists of two sequences or strands that are held together by complementary nucleotide pairs. DNA has the ability to *replicate* itself, which is done by separating the two strands and complementing both with the matching strand made from free nucleotides in the surrounding solution. DNA is the memory of evolution that gives coherence to all living species; it forms the material basis of heredity as studied by Mendel in the nineteenth century [Men66]. Apparently, only a small fraction of the DNA in any organism represents used information. The used pieces are the *genes*, which are *transcribed* into RNA in a process similar to replication. RNA remains single-stranded and most of it gets *translated* into protein. This happens in the *ribosome*, which functions as a large molecular machine made of proteins and RNA molecules. A single strand of RNA is fed into the ribosome, and each triplet of nucleotides is translated into an amino acid, which is appended to the growing peptide chain. Upon completion, this chain leaves the ribosome as the final protein. This scenario is reminiscent of the Turing machine model of computing, in which information is read from an input tape and the results of the computations are printed on an output tape.

FROM SEQUENCE TO FUNCTION

When the protein leaves the ribosome, it folds up to form a shape that is characteristic for its sequence of amino acids. The proteins constitute the workforce

that maintains organic life. Specific proteins fulfill specific functions within the organism, and the particular shape it assumes is crucial:

$$\text{Sequence} \implies \text{Form} \implies \text{Function.}$$

This is why geometry is important in molecular biology. It is essential to learn the shapes of all proteins and to understand what is important about them. Most functions are tied up in the interaction of proteins with each other and with other molecules. The replication of DNA, the transcription to RNA, and the translation to protein are but three examples, and each is served by a complicated machine made of different proteins and RNA molecules. In other words, proteins are the pieces of a huge three-dimensional dynamic puzzle whose solution requires, among others things, a good understanding of the shapes involved. A major difficulty in the field of molecular biology is the minuscule scale of space and time at which the processes take place. The actors and their scripts are complicated and observations are indirect. Experimental work is generally complemented by computational simulations, which are referred to as theoretical work in this area.

63.2 GEOMETRIC MODELS

Proteins are complicated objects, which have been abstracted into a number of different models emphasizing different aspects of their behavior. We may think of them as curves in space modeling the backbone, or as collections of balls or spheres representing it at the level of individual atoms.

GLOSSARY

Space-filling diagram: Model that represents a protein by the space it occupies.

Most commonly, each atom is represented by a ball (a solid sphere), and the protein is the union of these balls.

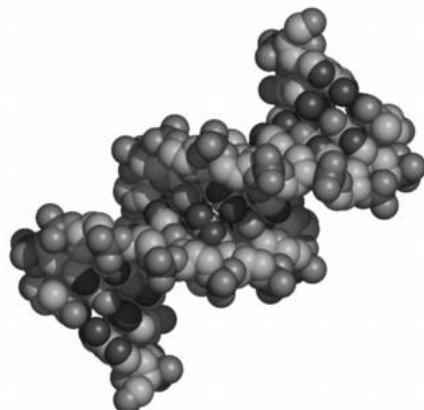


FIGURE 63.2.1

A short segment of a DNA double helix in space-filling representation. DNA uses an alphabet of four nucleotides: adenine (A), guanine (G), cytosine (C), and thymine (T). In the picture many of the nucleotides are barely visible since they are packed in the middle, using hydrogen bonds to hold the strands together.

Van der Waals surface: Boundary of space-filling diagram defined as the union of balls with **van der Waals radii**. The sizes of these balls are chosen to reflect the transition from an attractive to a repulsive van der Waals force.

Solvent-accessible surface: Boundary of space-filling diagram in which each van der Waals ball is enlarged by the radius of the solvent sphere. Alternatively, it is the set of centers of solvent spheres that touch but do not otherwise intersect the van der Waals surface.

Molecular surface: Boundary of the portion of space inaccessible to the solvent. It is obtained by rolling the solvent sphere about the van der Waals surface.

Power distance: Square length of tangent line segment from a point x to a sphere with center z and radius r . It is also referred to as the *weighted square distance* and formally defined as $\|x - z\|^2 - r^2$.

Voronoi diagram: Decomposition of space into convex polyhedra. Each polyhedron belongs to a sphere in a given collection and consists of all points for which this sphere minimizes the power distance. This decomposition is also known as the *power diagram* and the *weighted Voronoi diagram*.

Delaunay triangulation: Dual to the Voronoi diagram. For generic collections of spheres, it is a simplicial complex consisting of tetrahedra, triangles, edges, and vertices. This complex is also known as the *regular triangulation*, the *coherent triangulation*, and the *weighted Delaunay triangulation*.

Dual complex: Dual to the Voronoi decomposition of a union of balls. It is a subcomplex of the Delaunay triangulation.

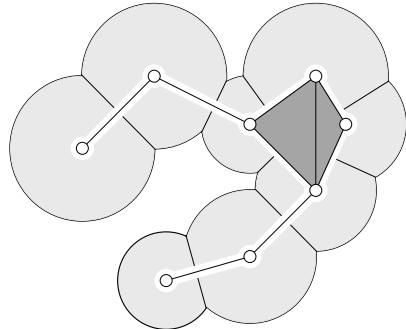


FIGURE 63.2.2

Each Voronoi polygon intersects the union of disks in a convex set, which is the intersection with its defining disk. The drawing shows the Voronoi decomposition of the union and the dual complex superimposed.

Growth model: Rule for growing all spheres in a collection continuously and simultaneously. The rule that increases the square radius r^2 to $r^2 + t$ at time t keeps the Voronoi diagram invariant at all times.

Alpha complex: The dual complex at time $t = \alpha^2$ for a collection of spheres that grow while keeping the Voronoi diagram invariant. The *alpha shape* is the underlying space of the alpha complex.

Filtration: Nested sequence of complexes. The prime example here is the sequence of alpha complexes.

SPACE-FILLING DIAGRAMS

Our starting point is the *van der Waals force*, which is based on quantum mechanical effects. At short range up to a few Angstrom, the force is attractive but significantly weaker than covalent or ionic bonds. At very short range, the force is strongly repulsive. We may assign *van der Waals radii* to the atoms so that the force changes from attractive to repulsive when the corresponding spheres touch

[GR01]. The *van der Waals surface* is the boundary of the space-filling diagram made up of the balls with van der Waals radii. In the 1970s, Richards and collaborators extended this idea to capture the interaction of a protein with the surrounding solvent [LR71, Ric77]. The *solvent-accessible surface* is the boundary of the space-filling diagram in which the balls are grown by the radius of the sphere that models a single solvent molecule. Usually the solvent is water, represented by a sphere of radius 1.4 Angstrom. The *molecular surface* is obtained by rolling the solvent sphere over the van der Waals surface and filling in the inaccessible crevices and cusps. This surface is sometimes referred to as the *Connolly surface*, after the creator of the first software representing this surface by a collection of dots [Con83].

DUAL STRUCTURES

We complement the space-filling representations of proteins with geometrically dual structures. A major advantage of these dual structures is their computational convenience. We begin by introducing the *Voronoi diagram* of a collection of balls or spheres, which decomposes the space into convex polyhedra [Vor07]. Next we intersect the union of balls with the Voronoi diagram and obtain a decomposition of the space-filling diagram into convex *cells*. Indeed, these cells are the intersections of the balls with their corresponding Voronoi polyhedra. The *dual complex* is the collection of simplices that express the intersection pattern between the cells: we have a vertex for every cell, an edge for every pair of cells that share a common facet, a triangle for every triplet of cells that share a common edge, and a tetrahedron for every quadruplet of cells that share a common point [EKS83, EM94]. This exhausts all possible intersection patterns in the assumed generic case. We get a natural embedding if we use the sphere centers as the vertices of the dual complex.

GROWTH MODEL

One and the same Voronoi diagram corresponds to more than just one collection of spheres. For example, if we grow the square radius r_i^2 of the i th sphere to $r_i^2 + t$, for every i , we get the same Voronoi diagram. Think of t as time parametrizing this particular growth model of the spheres. While the Voronoi diagram remains fixed, the dual complex changes. The cells in which the balls intersect the Voronoi polyhedra grow monotonically with time, which implies that the dual complex can acquire but not lose simplices. We thus get a nested sequence of dual complexes,

$$\emptyset = K_0 \subset K_1 \subset \dots \subset K_m = D,$$

which begins with the empty complex at time $t = -\infty$ and ends with the Delaunay triangulation [Del34] at time $t = \infty$. We refer to this sequence as a *filtration* of the Delaunay triangulation and think of it as the dual representation of the protein at all scale levels.

63.3 MESHING

We introduce yet another surface bounding a space-filling diagram of sorts. The *molecular skin* is the boundary of the union of infinitely many balls. Besides

the balls with van der Waals radii representing the atoms, we have balls interpolating between them that give rise to blending patches and, all together, to a tangent-continuous surface. The molecular skin is rather similar in appearance to the molecular surface but uses hyperboloids instead of tori to blend between the spheres [Ede99]. The smoothness of the surface permits a mesh whose triangles are all approximately equiangular [CDES01]. Applications of this mesh include the representation of proteins for visualization purposes and the solution of differential equations defined over the surface by finite-element and other numerical methods.

GLOSSARY

Molecular skin: Surface of a molecule that is geometrically similar to the molecular surface but uses hyperboloid instead of torus patches for blending.

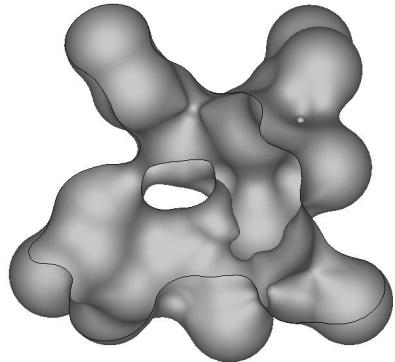


FIGURE 63.3.1

Cutaway view of the skin of a small molecule. We see a blend of sphere and hyperboloid patches. The surface is inside-outside symmetric: it can be defined by a collection of spheres on either of its two sides.

Mixed complex: Decomposition of space into shrunken Voronoi polyhedra, shrunken Delaunay tetrahedra, and shrunken products of corresponding Voronoi polygons and Delaunay edges as well as Voronoi edges and Delaunay triangles. It decomposes the skin surface into sphere and hyperboloid patches.

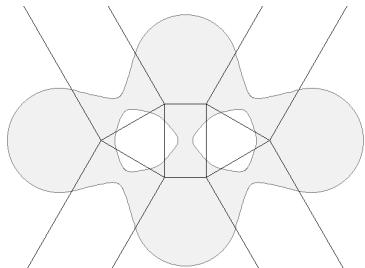


FIGURE 63.3.2

The skin curve defined by four circles in the plane. The mixed complex decomposes the curve into pieces of circles and hyperbolas.

Maximum normal curvature: The larger absolute value $\kappa(x)$ of the two principal curvatures at a point x of the surface.

ε -sampling: A collection S of points on the molecular skin M such that every point $x \in M$ has a point $u \in S$ at distance $\|x - u\| \leq \varepsilon/\kappa(x)$.

Restricted Delaunay triangulation: Dual to the restriction of the (three-dimensional) Voronoi diagram of S to the molecular skin M .

Shape space: Locally parametrized space of shapes. The prime example here is the $(k-1)$ -dimensional space generated by k shapes, each specified by a collection of spheres in \mathbb{R}^3 .

TRIANGULATION

The molecular skin has geometric properties that can be exploited to construct a numerically high-quality mesh and to maintain that mesh during deformation. The most important of these is the continuity of the *maximum normal curvature* function $\kappa : M \rightarrow \mathbb{R}$. To define it, consider the 1-parameter family of geodesics passing through x and let $\kappa(x)$ be the maximum of their curvatures at x . We use this function to guide the local density of the points distributed over M that are used as vertices of the mesh. Given such a collection S of points, we construct a mesh using its Voronoi diagram restricted to M . The polyhedra decompose the surface into patches, and the mesh is constructed as the dual of that decomposition [Che93]. As proved in [ES97], the mesh is homeomorphic to the surface if the pieces of the restricted Voronoi diagram are topologically simple sets of the appropriate dimensions. In other words, the intersection of each Voronoi polyhedron, polygon, or edge with M is either empty or a topological disk, interval, or single point. Because of the smoothness of M , this topological property is implied if the points form an ε -sampling, with $\varepsilon = 0.279$ or smaller [CDES01].

DEFORMATION AND SHAPE SPACE

The variation of the maximum normal curvature function can be bounded by the one-sided Lipschitz condition $|1/\kappa(x) - 1/\kappa(y)| \leq \|x - y\|$, where the distance is measured in \mathbb{R}^3 . The continuity over \mathbb{R}^3 and not just over M is crucial when it comes to maintaining the mesh while changing the surface. This leads us to the topic of deformations and shape space. The latter is constructed as a parametrization of the deformation process. The deformation from a shape A_0 to another shape A_1 can be written as $\lambda_0 A_0 + \lambda_1 A_1$, with $\lambda_1 = 1 - \lambda_0$. Accordingly, we may think of the unit interval as a one-dimensional shape space. We can generalize this to a k -dimensional shape space as long as the different ways of arriving at $(\lambda_0, \lambda_1, \dots, \lambda_k)$, with $\sum \lambda_i = 1$ and $\lambda_i \geq 0$ for all i , all give the same shape $A = \sum \lambda_i A_i$. How to define deformations so that this is indeed the case is explained in [CEF01].

63.4 CONNECTIVITY AND SHAPE FEATURES

Protein connectivity is often understood in terms of its covalent bonds, in particular along the backbone. In this section, we discuss a different notion, namely the topological connectivity of the space assigned to a protein by its space-filling diagram. We mention *homeomorphisms*, *homotopies*, *homology groups*, and *Euler characteristics*, which are common topological concepts used to define and talk about connectivity. Of particular importance are the homology groups and their ranks, the *Betti numbers*, as they lend themselves to efficient algorithms. In addition to computing the connectivity of a single space-filling diagram, we study how

the connectivity changes when the balls grow. The sequence of space-filling diagrams obtained this way corresponds to the filtration of dual complexes introduced earlier. We use this filtration to define basic shape features, such as pockets in proteins and interface surfaces between complexed proteins and molecules.

GLOSSARY

Topological equivalence: Equivalence relation between topological spaces defined by *homeomorphisms*, which are continuous bijections with continuous inverses.

Homotopy equivalence: Weaker equivalence relation between topological spaces \mathbb{X} and \mathbb{Y} defined by maps $f : \mathbb{X} \rightarrow \mathbb{Y}$ and $g : \mathbb{Y} \rightarrow \mathbb{X}$ whose compositions $g \circ f$ and $f \circ g$ are homotopic to the identities on \mathbb{X} and on \mathbb{Y} .

Deformation retraction: A homotopy between the identity on \mathbb{X} and a retraction of \mathbb{X} to $\mathbb{Y} \subseteq \mathbb{X}$ that leaves \mathbb{Y} fixed. The existence of the deformation implies that \mathbb{X} and \mathbb{Y} are homotopy-equivalent.

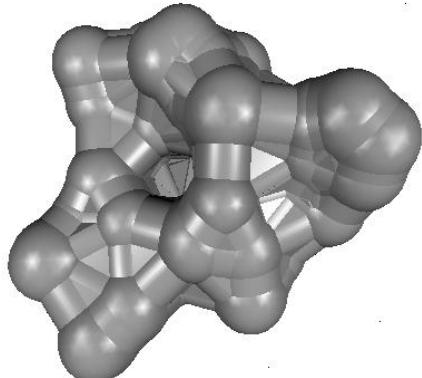


FIGURE 63.4.1

Snapshot during the deformation retraction of the space-filling representation of gramicidin to its dual complex. The spheres shrink to vertices while the intersection circles become cylinders that eventually turn into edges.

Homology groups: Quotients of cycle groups and their boundary subgroups. There is one group per dimension. The *kth Betti number*, β_k , is the rank of the *kth homology group*.

Euler characteristic: The alternating sum of Betti numbers: $\chi = \sum_{k \geq 0} (-1)^k \beta_k$.

Voids: Bounded connected components of the complement. Here, we are primarily interested in voids of space-filling diagrams embedded in \mathbb{R}^3 .

Pockets: Maximal regions in the complement of a space-filling diagram that become voids before they disappear. Here, we assume the growth model that preserves the Voronoi diagram of the spheres.

Persistent homology groups: Quotients of the cycle groups at some time t and their boundary subgroups a later time $t + p$. The ranks of these groups are the *persistent Betti numbers*.

Protein complex: Two or more docked proteins. A complex can be represented by a single space-filling diagram of colored balls.

Molecular interface surface: Surface consisting of bichromatic Voronoi polygons that separate the proteins in the complex. The surface is retracted to the region in which the proteins are in close contact.

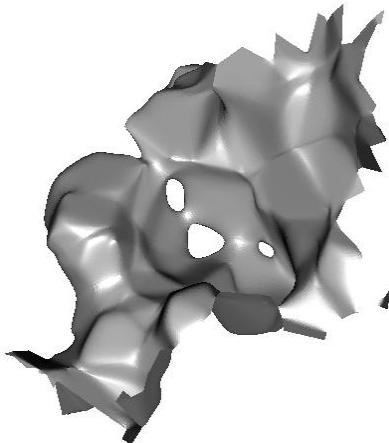


FIGURE 63.4.2

Molecular interface surface of the neurotoxic vipoxin complex. The surface has nonzero genus, which is unusual. In this case, we have genus equal to three, which implies the existence of three loops from each protein that are linked with each other. The linking might explain the unusually high stability of the complex, which remains for years in solution. The piecewise linear surface has been smoothed to improve visibility.

CLASSIFICATION

The connectivity of topological spaces is commonly discussed by forming equivalence classes of spaces that are connected the same way. Sameness may be defined as being homeomorphic, being homotopy-equivalent, having isomorphic homology groups, or having the same Euler characteristic. In this sequence, the classification gets progressively coarser but also easier to compute. Homology groups seem to be a good compromise as they capture a great deal of connectivity information and have fast algorithms. The classic approach to computing homology groups is algebraic and considers the incidence matrices of adjacent dimensions. Each matrix is reduced to *Smith normal form* using a Gaussian-elimination-like reduction algorithm. The ranks and torsion coefficients of the homology groups can be read off directly from the reduced matrices [Mun84]. Depending on which coefficients we use and exactly how we reduce, the running time can be anywhere between cubic in the number of simplices and exponential or worse.

INCREMENTAL ALGORITHM

Space-filling diagrams are embedded in \mathbb{R}^3 and enjoy properties that permit much faster algorithms. To get started, we use the existence of a deformation retraction from the space-filling diagram to the dual complex, which implies that the two have isomorphic homology groups [Ede95]. The embedding in \mathbb{R}^3 prohibits nonzero torsion coefficients [AH35]. We therefore limit ourselves to Betti numbers, which we compute incrementally, by adding one simplex at a time in an order that agrees with the filtration of the dual complexes. When we add a k -dimensional simplex σ , the k th Betti number goes up by one if σ belongs to a k -cycle, and the $(k-1)$ st Betti number goes down by one if σ does not belong to a k -cycle. The two cases can be distinguished in a time that, for all practical purposes, is constant per operation, leading to an essentially linear time algorithm for computing the Betti numbers of all complexes in the filtration [DE95].

PERSISTENCE

To get a handle on the stability of a homology class, we observe that the simplices that create cycles can be paired with the simplices that destroy cycles. The *persistence* is the time lag between the creation and the destruction [ELZ02]. The idea of pairing lies also at the heart of two types of shape features relevant in the study of protein interactions. A *pocket* in a space-filling diagram is a portion of the outside space that becomes a void before it disappears [Kun92, EFL98]. It is represented by a triangle-tetrahedron pair: the triangle creates a void and the tetrahedron is the last piece that eventually fills that same void. The *molecular interface surface* consists of all bichromatic Voronoi polygons of a protein complex. To identify the essential portions of this surface, we again observe how voids are formed and retain the bichromatic polygons inside pockets while removing all others [BER03]. A different geometric formalization of the same biochemical concept can be found in [VBR⁺95]. Preliminary experiments suggest that the combination of molecular interfaces and the idea of persistence can be used to predict the hot-spot residues in protein-protein interactions [Wel96].

63.5 DENSITY MAPS

Continuous maps over manifolds arise in a variety of settings within structural molecular biology. One is *X-ray crystallography*, which is the most common method for determining the three-dimensional structure of proteins [BJ76, Rho00]. While casting X-rays on a crystal of purified protein, we observe diffraction patterns, from which the electron density of the protein can be obtained via an inverse Fourier transform. Another setting is *molecular mechanics*, whose central object is the force field that drives atomic motions. We may, for example, be interested in the electrostatic potential induced by a protein and visualize it as a density map over three-dimensional space or over a surface embedded in that space. As a third setting, we mention the *protein docking* problem. Given two proteins, or a protein and a ligand, we try to fit protrusions of one into the cavities of the other [Con86]. We make up continuous functions related to the shapes of the surfaces and identify protrusions and cavities as local extremes of these functions. Morse theory is the natural mathematical framework for studying these maps [Mil63, Mat02].

GLOSSARY

Morse function: Generic smooth map on a manifold, $f : \mathbb{M} \rightarrow \mathbb{R}$. In particular, the genericity assumption requires that all critical points be nondegenerate and have different function values.

Gradient, Hessian: The vector of first derivatives and the matrix of second derivatives.

Critical point: Point at which the gradient of f vanishes. It is *nondegenerate* if the Hessian is invertible. The *index* of a nondegenerate critical point is the number of negative eigenvalues of the Hessian.

Integral line: Maximal curve whose velocity vectors agree with the gradient of the Morse function. Two integral lines are either disjoint or the same.

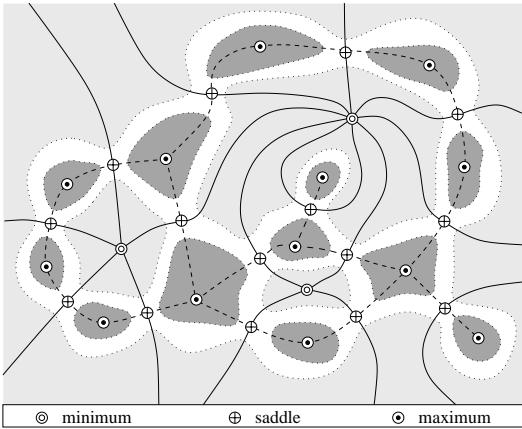
Stable manifold: Union of integral lines converging to the same critical point.

We get *unstable manifolds* if we negate f and thus effectively reverse the gradient.

Morse-Smale complex: Collection of cells obtained by intersecting stable with unstable manifolds. We require f to be a *Morse-Smale function* satisfying the additional genericity assumption that these intersections are transversal.

FIGURE 63.5.1

Portion of the Morse-Smale complex of a Morse-Smale function over a 2-manifold. The solid stable 1-manifolds and the dashed unstable 1-manifolds are shown together with two dotted level sets. Observe that all two-dimensional regions of the complex are quadrangular.



Cancellation: Local change of the Morse function that removes a pair of critical points. Their indices are necessarily contiguous.

CRITICAL POINTS

Classic Morse theory applies only to generic smooth maps on manifolds, $f : M \rightarrow \mathbb{R}$. Maps that arise in practice are rarely smooth and generic, or, more precisely, the information we are able to collect about maps is rarely enough to go beyond a piecewise linear representation. To illustrate this point, we discuss critical points, which for smooth functions are characterized by a vanishing gradient: $\nabla f = 0$. If we draw a small circle around a noncritical point u on a 2-manifold, we get one arc along which the function takes on values less than $f(u)$ and a complementary arc along which the function is greater than or equal to $f(u)$. Call the former arc the *lower link* of u . We get different lower links for critical points: the entire circle for a *minimum*, two arcs for a *saddle*, and the empty set for a *maximum*. A typical representation of a piecewise linear map is a triangulation with function values specified at the vertices and linearly interpolated over the edges and triangles. The lower link of a vertex can still be defined and the criticality of the vertex can be determined from the topology of the lower link [Ban67].

MORSE-SMALE COMPLEXES

In the smooth case, each critical point defines a *stable manifold* of points that converge to it by following the gradient flow. Symmetrically, it defines an *unstable manifold* of points that converge to it by following the reversed gradient flow. These manifolds define decompositions of the manifold into simple cells [Tho49].

Extensions of these ideas to construct similar cell decompositions of manifolds with piecewise linear continuous functions can be found in [EHZ03]. In practice, it is essential to be able to simplify these decompositions, which can be done by canceling critical points in pairs in the order of increasing persistence [ELZ02].

63.6 MATCH AND FIT

Proteins can be similar in a variety of ways: they can have similar residue sequences, they can have backbones that are laid out similarly in space, and they can have similar shapes after folding. The first two notions are important in gaining insight into the evolutionary development of proteins. The corresponding computational problems are sequence alignment and structure alignment. The question of shape similarity, and, in particular, of partial shape similarity, is relevant in understanding the interaction between proteins and their substrates, which can be proteins or other molecules. Indeed, many interactions seem to require a high degree of partial shape complementarity, which we interpret as a high degree of partial shape similarity between the protein and the complement of its substrate.

GLOSSARY

Rigid motion: Orientation- and distance-preserving motion. The primary examples here are rigid motions of three-dimensional space, $\mu : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. Each rigid motion can be decomposed into a rotation followed by a translation.

RMSD: Root-mean-square distance. Root of the average square distance between two sets of points with a given bijection.

Dynamic programming: Algorithmic paradigm which computes the optimum from precomputed optimal solutions to subproblems.

Sequence alignment: Collection of monotonically increasing maps to the integers, one per sequence. Each letter gets either matched or skipped.

Structure alignment: Collection of monotonically increasing maps to the integers, one per chain of points modeling a protein backbone.

Protein docking: Process in which a protein forms a complex with another molecule. The complex usually exists only temporarily and facilitates an interaction between the molecules.

STRUCTURE ALIGNMENT

There are two approaches to structure alignment. The first compares the matrices of internal sequences between the points [HS93]. We discuss only the second approach, which is a direct extension of the work on sequence alignments in bioinformatics [Gus97]. Instead of letters representing residues, we align points in space, which are the centers of the alpha carbon atoms along the two backbones. For decomposable score functions, we can find the optimal alignment with dynamic programming in time that is quadratic in the number of points. One such function suggested in [SLL93] penalizes unmatched points and, for every matched pair (u_i, v_j) , adds

$$\delta(u_i, v_j) = \frac{100}{5 + \|u_i - v_j\|^2}$$

to the score. The dynamic programming approach works only for two fixed sequences, and the six degrees of freedom we gain by allowing rigid motions complicate matters considerably. Nevertheless, it is possible to compute an approximation to the optimal alignment in time that is polynomial in the number of points and the tolerated error [KL02].

RIGID MOTIONS

Let u_1, u_2, \dots, u_n and v_1, v_2, \dots, v_n be two sequences of points in \mathbb{R}^3 . For a given rigid motion μ , the *root-mean-square distance* between the sequences is

$$f(\mu) = \sqrt{\frac{1}{n} \sum_{i=1}^n \|u_i - \mu(v_i)\|^2}.$$

It is perhaps surprising that the dependence of f on μ can be expressed by a quadratic function which, in the generic case, has a unique local minimum. To describe the minimizing rigid motion, we decompose it into a translation followed by a rotation. Under the assumption that the centroid of the u_i lies at the origin, the optimum translation moves the centroid of the v_i to the origin, and the optimum rotation can be computed by solving a straightforward eigenvalue problem. One of the earliest references to this result is Kabsch [Kab78]. A lucid description of the proof using quaternions to represent rotations can be found in Horn [Hor87].

PROTEIN DOCKING

A good local geometric fit is a necessary condition for a complex between two or more proteins to be formed. There are, however, additional factors, such as electrostatic and hydrophobic forces. (Some side chains of proteins are attracted to water molecules, others repelled by them. The former are called ‘hydrophilic,’ the latter ‘hydrophobic.’ This turns out to be a significant factor in the protein folding process.) To further complicate the issue, proteins are somewhat flexible and can sometimes avoid otherwise prohibitive steric clashes [ESM01]. Taking all these factors into account seems prohibitive and most computational approaches to protein docking explore the space of rigid motions using relatively simple geometric score functions [HMWN02]. An example is the number of atoms in close but not too close distance from each other. The space of rigid motions is six-dimensional and exploring it is time-consuming, even with simple score functions. The idea of Connolly to use critical points of Morse functions to identify motions [Con86] seems promising, but is not yet fully explored. It is usually combined with geometric hashing to enumerate the motions suggested by the critical point patterns [NLWN94].

63.7 MEASURES AND DERIVATIVES

Computing the volume and the surface area of a space-filling diagram are two of the most fundamental means to characterize the geometry of a protein. To mention

a specific application, we consider the computation of the *solvation energy*, which is central in the simulation of folding and docking processes. Many simulations use *implicit solvent models* and describe the hydrophobic part of the solvation energy as a weighted sum of the accessible surface area or, alternatively, as a weighted sum of volumes. The weights are experimentally determined *solvation parameters* that assess the contributions of different atom types to the hydrophobic term [EM86]. A molecular dynamics simulation requires the weighted area or volume and its derivative in order to estimate the contribution of the hydrophobic term to the energy that drives the process.

GLOSSARY

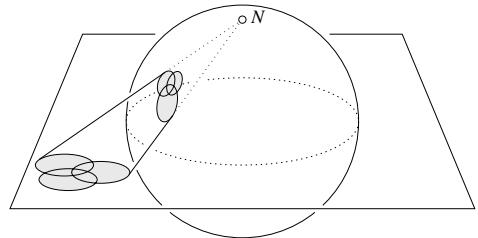
Indicator function: Maps a point x to 1 if $x \in P$ and to 0 if $x \notin P$, where P is some fixed set. Here, we are interested in convex polyhedra P and can therefore use the alternating sum of the number of faces of various dimensions visible from x as indicator. (For further details, see [Ede95].)

Inclusion-exclusion: Principle used to compute the volume of a union of bodies as the alternating sum of volumes of k -fold intersections, for $k \geq 1$.

Stereographic projection: Mapping of the 3-sphere minus a point to the three-dimensional Euclidean space. The map preserves spheres and angles.

FIGURE 63.7.1

Stereographic projection from the north pole. The preimage of a circle in the plane is a circle on the sphere, which is the intersection of the sphere with a plane. By extension, the preimage of a union of disks is the intersection of the sphere with the complement of a convex polyhedron.



Atomic solvation parameters: Experimentally determined numbers that assess the hydrophobicity of different atoms.

Weighted volume: Volume of a space-filling diagram in which the contribution of each individual ball is weighted by its atomic solvation parameter. Also a function $V : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ obtained by parametrizing a space-filling diagram by the coordinates of its n ball centers.

Weighted-volume derivative: The linear map $DV_z : \mathbb{R}^{3n} \rightarrow \mathbb{R}$ defined by $DV_z(t) = \langle v, t \rangle$, where $z \in \mathbb{R}^{3n}$ specifies the space-filling diagram, $t \in \mathbb{R}^{3n}$ lists the coordinates of the motion vectors, and $v = \nabla V(z)$ is the gradient of V at z . It is also the map $DV : \mathbb{R}^{3n} \rightarrow \mathbb{R}^{3n}$ that maps z to v .

GEOMETRIC INCLUSION-EXCLUSION

Work on computing the volume and the area of a space-filling diagram $F = \bigcup_i B_i$ can be divided into approximate [Row63] and exact methods [Ric74]. According to the principle of inclusion-exclusion, the volume of F can be expressed as an

alternating sum of volumes of intersections:

$$\text{vol } F = \sum_{\Lambda} (-1)^{\text{card } \Lambda + 1} \text{vol} \bigcap_{i \in \Lambda} B_i,$$

where Λ ranges over all nonempty subsets of the index set. The size of this formula is exponential in the number of balls, and the individual terms can be quite complicated. Most of the terms are redundant, however, and a much smaller formula based on the dual complex K of the space-filling diagram F has been given [Ede95]:

$$\text{vol } F = \sum_{\sigma \in K} (-1)^{\dim \sigma} \text{vol} \bigcap \sigma,$$

where $\bigcap \sigma$ denotes the intersection of the $\dim \sigma + 1$ balls whose centers are the vertices of σ . The proof is based on the Euler formula for convex polyhedra and uses stereographic projection to relate the space-filling diagram in \mathbb{R}^3 with a convex polyhedron in \mathbb{R}^4 . Precursors of this result include the existence proof of a polynomial size inclusion-exclusion formula [Kra78] and the presentation of such a formula using the simplices in the Delaunay triangulation [NW92]. We note that it is straightforward to modify the formula to get the weighted volume: decompose the terms $\text{vol} \bigcap \sigma$ into the portions within the Voronoi cells of the participating balls and weight each portion accordingly.

DERIVATIVES

The relationship between the weighted- and unweighted-volume derivatives is less direct than that between the weighted and unweighted volumes. Just to state the formula for the weighted-volume derivative requires more notation than we are willing to introduce here. Instead, we describe the two geometric ingredients, both of which can be computed by geometric inclusion-exclusion [EK03]. The first ingredient is the area of the portion of the disk spanned by the circle of two intersecting spheres that belongs to the Voronoi diagram. This facet is the intersection of the disk with the corresponding Voronoi polygon. The second ingredient is the weighted average vector from the center of the disk to the boundary of said facet. The weight is the infinitesimal contribution to the area as we rotate the vector to sweep out the facet.

63.8 SOURCES AND RELATED MATERIAL

FURTHER READING

For background reading in **algorithms** we recommend: [CLR90], which is a comprehensive introduction to combinatorial algorithms; [Gus97], which is an algorithms text specializing in bioinformatics; [Str93], which is an introduction to linear algebra; and [Sch02], which is a numerical algorithms text in molecular modeling.

For background reading in **geometry** we recommend: [Ped88], which is a geometry text focusing on spheres; [Nee97], which is a lucid introduction to geometric

transformations; [FT72], which studies packing and covering in two and three dimensions; and [Ede01], which is an introduction to computational geometry and topology, focusing on Delaunay triangulations and mesh generation.

For background reading in **topology** we recommend: [Ale61], which is a compilation of three classical texts in combinatorial topology; [Gib77], which is a very readable introduction to homology groups; [Mun84], which is a comprehensive text in algebraic topology; and [Mat02], which is a recent introduction to Morse theory.

For background reading in **biology** we recommend: [ABL⁺94], which is a basic introduction to molecular biology on the cell level; [Str88], which is a fundamental text in biochemistry; and [Cre93], which is an introduction to protein sequences, structures, and shapes.

RELATED CHAPTERS

- [Chapter 2: Packing and covering](#)
- [Chapter 23: Voronoi diagrams and Delaunay triangulations](#)
- [Chapter 25: Triangulations and mesh generation](#)
- [Chapter 32: Computational topology](#)

REFERENCES

- [ABL⁺94] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell*. Garland, New York, 1994.
- [Ale61] P.S. Aleksandrov. *Elementary Concepts of Topology*. Dover, New York, 1961.
- [AH35] P.S. Aleksandrov and H. Hopf. *Topologie I*. Julius Springer, Berlin, 1935.
- [BER03] Y.-E. Ban, H. Edelsbrunner, and J. Rudolph. A definition of interface surfaces for protein oligomers. Manuscript, Duke Univ., Durham, 2003.
- [Ban67] T.F. Banchoff. Critical points and curvature for embedded polyhedra. *J. Differential Geom.*, 1:245–256, 1967.
- [BJ76] T. Blundell and L. Johnson. *Protein Crystallography*. Academic Press, New York, 1976.
- [CDES01] H.-L. Cheng, T.K. Dey, H. Edelsbrunner, and J. Sullivan. Dynamic skin triangulation. *Discrete Comput. Geom.*, 25:525–568, 2001.
- [CEF01] H.-L. Cheng, H. Edelsbrunner, and P. Fu. Shape space from deformation. *Comput. Geom. Theory Appl.*, 19:191–204, 2001.
- [Che93] L.P. Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, 1993, pages 274–280.
- [Con83] M.L. Connolly. Analytic molecular surface calculation. *J. Appl. Crystallogr.*, 6:548–558, 1983.
- [Con86] M.L. Connolly. Measurement of protein surface shape by solid angles. *J. Molecular Graphics*, 4:3–6, 1986.
- [CLR90] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, 1990.
- [Cre93] T.E. Creighton. *Proteins*. Freeman, New York, 1993.
- [Del34] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7:793–800, 1934.

- [DE95] C.J.A. Delfinado and H. Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Comput. Aided Geom. Design*, 12:771–784, 1995.
- [Ede95] H. Edelsbrunner. The union of balls and its dual shape. *Discrete Comput. Geom.*, 13:415–167, 1995.
- [Ede99] H. Edelsbrunner. Deformable smooth surface design. *Discrete Comput. Geom.*, 21:87–115, 1999.
- [Ede01] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge Univ. Press, 2001.
- [EFL98] H. Edelsbrunner, M.A. Facello, and J. Liang. On the definition and the construction of pockets in macromolecules. *Discrete Appl. Math.*, 88:83–102, 1998.
- [EHZ03] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchy of Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete Comput. Geom.*, 30:87–107, 2003.
- [EKS83] H. Edelsbrunner, D.G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory*, 29:551–559, 1983.
- [EK03] H. Edelsbrunner and P. Koehl. The weighted-volume derivative of a space-filling diagram. *Proc. Nat. Acad. Sci. U.S.A.*, 100:2203–2208, 2003.
- [ELZ02] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.
- [EM94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graphics*, 13:43–72, 1994.
- [ES97] H. Edelsbrunner and N.R. Shah. Triangulating topological spaces. *Internat. J. Comput. Geom. Appl.*, 7:365–378, 1997.
- [EM86] D. Eisenberg and A. McLachlan. Solvation energy in protein folding and binding. *Nature*, 319:199–203, 1986.
- [ESM01] A.H. Elcock, D. Sept, and J.A. McCammon. Computer simulation of protein-protein interaction. *J. Phys. Chem.*, 105:1504–1518, 2001.
- [FT72] L. Fejes Tóth. *Lagerungen in der Ebene auf der Kugel und im Raum*, 2nd Ed. Springer-Verlag, Berlin, 1972.
- [GR01] M. Gerstein and F.M. Richards. Protein geometry: distances, areas, and volumes. In M.G. Rossman and E. Arnold, editors, *The International Tables for Crystallography*, Vol. F, Chapter 22, pages 531–539. Kluwer, Dordrecht, 2001.
- [Gib77] P.J. Giblin. *Graphs, Surfaces, and Homology. An Introduction to Algebraic Topology*. Chapman and Hall, London, 1977.
- [Gus97] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge Univ. Press, 1997.
- [HMWN02] I. Halperin, B. Mao, H. Wolfson, and R. Nussinov. Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins*, 47:409–443, 2002.
- [HS93] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Molecular Biol.*, 233:123–138, 1993.
- [Hor87] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer. A*, 4:629–642, 1987.
- [Kab78] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. Sect. A*, 34:827–828, 1978.
- [KL02] R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. Manuscript, Stanford Univ., 2002.

- [Kra78] K.W. Kratky. The area of intersection of n equal circular disks. *J. Phys. A*, 11:1017–1024, 1978.
- [Kun92] I.D. Kuntz. Structure-based strategies for drug design and discovery. *Science*, 257:1078–1082, 1992.
- [LR71] B. Lee and F.M. Richards. The interpretation of protein structures: estimation of static accessibility. *J. Molecular Biol.*, 55:379–400, 1971.
- [Mat02] Y. Matsumoto. *An Introduction to Morse Theory*. Amer. Math. Soc., Providence, 2002.
- [Men66] G. Mendel. Versuche über Pflanzen-Hybriden. *Verh. naturforsch. Ver.*, Abh., Brünn, 4:3–47, 1866.
- [Mil63] J. Milnor. *Morse Theory*. Princeton Univ. Press, 1963.
- [Mun84] J.R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Redwood City, 1984.
- [NW92] D.Q. Naiman and H.P. Wynn. Inclusion-exclusion-Bonferroni identities and inequalities for discrete tube-like problems via Euler characteristics. *Ann. Statist.*, 20:43–76, 1992.
- [Nee97] T. Needham. *Visual Complex Analysis*. Clarendon Press, Oxford, 1997.
- [NLWN94] R. Norel, S.L. Lin, H. Wolfson, and R. Nussinov. Shape complementarity at protein-protein interfaces. *Biopolymers*, 34:933–940, 1994.
- [Ped88] D. Pedoe. *Geometry. A Comprehensive Course*. Dover, New York, 1988.
- [Rho00] G. Rhodes. *Crystallography Made Crystal Clear*, 2nd ed. Academic Press, San Diego, 2000.
- [Ric74] F.M. Richards. The interpretation of protein structures: total volume, group volume distributions and packing density. *J. Molecular Biol.*, 82:1–14, 1974.
- [Ric77] F.M. Richards. Areas, volumes, packing, and protein structures. *Ann. Rev. Biophys. Bioeng.*, 6:151–176, 1977.
- [Row63] J.S. Rowlinson. The triplet distribution function in a fluid of hard spheres. *Molecular Phys.*, 6:517–524, 1963.
- [Sch02] T. Schlick. *Molecular Modeling and Simulation*. Springer-Verlag, New York, 2002.
- [Str93] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, 1993.
- [Str88] L. Stryer. *Biochemistry*. Freeman, New York, 1988.
- [SLL93] S. Subbiah, D.V. Laurents, and M. Levitt. Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Current Biol.*, 3:141–148, 1993.
- [Tho49] R. Thom. Sur une partition en cellules associée à une fonction sur une variété. *C. R. Acad. Sci. Paris*, 228:973–975, 1949.
- [VBR⁺95] A. Varshney, F.P. Brooks, Jr., D.C. Richardson, W.V. Wright, and D. Manocha. Defining, computing, and visualizing molecular interfaces. In *Proc. IEEE Visualization, 1995*, pages 36–43.
- [Vor07] G.F. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *J. Reine Angew. Math.*, 133:97–178, 1907, and 134:198–287, 1908.
- [WC53] J.D. Watson and F.H.C. Crick. Molecular structure of nucleic acid: a structure for deoxyribose nucleic acid. Genetic implications of the structure of deoxyribonucleic acid. *Nature*, 171:737–738 and 964–967, 1953.
- [Wel96] J.A. Wells. Binding in the growth hormone receptor complex. *Proc. Nat. Acad. Sci. U.S.A.*, 93:1–6, 1996.

64 SOFTWARE

Michael Joswig

INTRODUCTION

This survey is intended as a guide through the ever growing jungle of geometry software. Software comes in many guises. There are the fully-fledged systems consisting of a hundred thousand lines of code which meet professional standards in software design and user support. But there are also the tiny code fragments scattered over the Internet which can nonetheless be valuable for research purposes. And, of course, the very many individual programs and packages in between.

Likewise today we find a wide group of users of geometry software. On the one hand, there are researchers in geometry, teachers, and their students. On the other hand, geometry software has found its way into numerous applications in the sciences as well as industry. Because it seems impossible to cover every possible aspect, we focus on software packages which are interesting from the researcher's point of view, and, to a lesser extent, from the student's.

This bias has a few implications. Most of the packages listed are designed to run on UNIX/Linux¹ machines. Moreover, the researcher's genuine desire to understand produces a natural inclination toward open source software. This is clearly reflected in the selection below. Major exceptions to these rules of thumb will be mentioned explicitly.

In order to keep the (already long) list of references as short as possible, in most cases only the Web address of each software package is listed rather than manuals and printed descriptions found elsewhere. At least for the freely available packages, this allows one to access the software directly. On the other hand, this may seem careless, since some Web addresses do not last long. This disadvantage usually can be compensated by relying on modern search engines.

The chapter is organized as follows. We start with a discussion of some technicalities (independent of particular systems). Since, after all, a computer is a technical object, the successful use of geometry software may depend on such things. The main body of the text consists of two halves. First, we browse through the topics of this handbook. Each of its major parts is linked to related software systems. Remarks on the algorithms are added mostly in areas where many implementations exist. Second, some of the software systems mentioned in the first part are listed in alphabetical order. We give a brief overview of some of their features. The libraries CGAL [F+02] and LEDA [led] are discussed in depth in [Chapter 65](#).

This survey is a snapshot as of Summer 2003. It is unlikely that it is complete in any sense. Even worse, the situation is changing so rapidly that the information given will be outdated soon. All this makes it almost impossible for the non-expert

¹No attempt is made to comment on differences between various UNIX platforms. Today's default UNIX is probably between Sun's Solaris and any Linux distribution. FreeBSD and its derivative Mac OS X come quite close. Many (text-based) UNIX programs can also be ported to various flavors of Windows via Cygwin [cyg03].

to get any impression of what software is available. Therefore, this is an attempt to provide an overview in spite of the obvious complications.

Nina Amenta authored the chapter on software in the first edition of this Handbook. Although much has changed during the last five years, her chapter still provided a good starting point for this survey. Moreover, this version of the chapter benefits from her constructive criticism.

GLOSSARY

Software can have various forms from the technical point of view. In particular, the amount of technical knowledge which is required by the user to use software varies considerably. The notions explained below are intended as guidelines.

Stand-alone software: This is a program which usually comes “as-is” and can be used immediately if properly installed. No programming skills are required.

Libraries: A collection of software components which can be accessed by writing a main program that calls functions implemented in the library. Good libraries come with example code that illustrates how (at least some of) the functions can be used. However, in order to exploit all the features, the user is expected to do some programming work. On the other hand, libraries have the advantage that they can be integrated into existing code. Some stand-alone programs can also be used as libraries; if they appear in this category, too, then there are substantial differences between the two versions, or the library has additional functionality.

Modules for general-purpose systems: Computer algebra and symbolic computation systems like **Mathematica** [mat03a], **Maple** [map03], **Matlab** [mat03b], **MuPAD** [mup03], and **REDUCE** [H⁺99] are integrated systems with an elaborate user interface which incorporate numerous algorithms from essentially all areas of mathematics. In this survey only functionality or extensions are listed which the author finds particularly noteworthy.

Additional Web pages: There are very many software overviews on the Web. A few of them that are focused on a specific topic are mentioned in the main text. Sometimes these pages have additional pieces of source code which may be useful. A short list of more comprehensive Web pages is given further below.

GENERAL SOURCES

There are several major web sites which are of general interest to the discrete and computational geometry community. Some of them also collect references to software, which are updated more or less frequently. We mention Amenta’s “Directory of Computational Geometry Software” [Ame97], Eppstein’s “Geometry Junkyard” [Epp03b], and Erickson’s “Computational Geometry Code” [Eri99].

For each of the major general-purpose computer algebra systems there exists a Web site with many additional packages and individual solutions. See the Web addresses of the respective products.

For those who are beginning to learn how to develop geometry software it will probably be too hard to do so by reading the source code of mature systems only. O’Rourke’s book [O’R98] can help fill this gap. Its numerous example programs in

C and Java are also electronically available [O'R00].

“The Stony Brook Algorithm Repository” maintained by Skiena [Ski01] gives an extensive overview of algorithms from several areas. Section 1.6 is dedicated to computational geometry and it contains links to implementations.

Although aging, the **Graphics Gems** by Kirk, Heckberth, Paeth, and many others [KHP⁺95], remains a useful resource. The **Gems** form a large collection of C source code examples for basic (and some more advanced) problems in computational geometry and computer graphics, originally published in a series of books [Gla93, Kir92, Arv91, Hec94].

ARITHMETIC

Depending on the application, issues concerning the arithmetic used for implementing a geometric algorithm can be essential. Using any kind of exact arithmetic is expensive, but the overhead induced also strongly depends on the application. A principal choice for exact arithmetic is *unlimited precision integer* or *rational* arithmetic as implemented in the **GNU Multiprecision Library** (GMP) [gmp03]. However, some problems require nonrational constructions. To cover such instances libraries like **Core** [YD03] and **LEDA** ([led], Chapter 65) offer special data types which allow for exact computation with certain radical expressions.

Geometric algorithms often rely on a few primitives like: Decide whether a point is on a hyperplane or, if not, on which side. Thus exact coordinates for geometric objects are sometimes less important than their true relative position. It is therefore natural to use techniques like interval arithmetic. **Floating-point filters** can be understood as an improved kind of interval arithmetic which employs higher precision or exact methods if needed. For more detailed information see [Chapter 41](#).

Yet another arithmetic concept is the following: Compute with machine size integers but halt (or trigger an exception) if an overflow occurs. Typically such an implementation depends on the hardware and thus requires at least a few lines of assembler code. Useful applications for such an approach are situations where the overflow signals that the computation is expected to become too large to finish in any reasonable amount of time. For instance, `t_homology` from **polymake**'s [GJ03] **TOPAZ** module implements Smith-Normal-Form in this way. Similarly, `hull` [Clab] uses exact integer arithmetic for convex hull computation and signals an overflow.

Instead of using a form of exact arithmetic, some implementations perform combinatorial post-processing in order to repair flawed results coming from rounding errors. An example is the convex hull code `qhull` by Barber, Dobkin, and Huhdanpaa [BDH01b]. Usually, this is only partially successful; see the discussion on the corresponding Web page [BDH01a] in `qhull`'s documentation.

FURTHER TECHNICAL REMARKS

While the programming language in which a software package is implemented often does not affect the user, this can obviously become an issue for the administrator who does the installation. Many of the software systems listed below are distributed as source code written in C or C++. Additionally, some of the larger packages are offered as precompiled binaries for common platforms.

C is usually easy. If the source code complies with the ANSI standard, it should be possible to compile it with any C compiler. The situation is quite different for C++. In spite of the fact that there is also an ANSI C++ standard, this standard is considerably more involved and thus far more difficult for the compiler constructors to implement. In fact, *none* of the currently existing C++ compilers fully conforms with the standard. They differ quite a bit in how much and in what respect they deviate; the main issue is template code. Therefore, at the moment it is quite unreasonable to expect that modern C++ code can be compiled with every C++ compiler. To the contrary: For the successful installation of C++ libraries it is often of the utmost importance to use the proper compiler, as specified in the respective installation instructions.

64.1 SOFTWARE SORTED BY TOPIC

This section should give a first indication of what software to use for solving a given problem. The subsections reflect the overall structure of the whole Handbook. References to CGAL [F⁺02] and LEDA [led] usually are omitted, since these large projects are covered in detail in [Chapter 65](#).

64.1.1 COMBINATORIAL AND DISCRETE GEOMETRY

This section deals with software handling the combinatorial aspects of finitely many objects, such as points, lines, or circles, in Euclidean space. Polytopes are described in Section 64.1.2.

STAND-ALONE SOFTWARE

The simplest geometric objects are clearly points. Therefore, essentially all geometry software can deal with them in one way or another. A key concept to many nontrivial properties of finite point sets in \mathbb{R}^d is the notion of an *oriented matroid*. For oriented matroid software and the computation of the set of all triangulations of a given point set see **TOPCOM** by Rambau [Ram03]. Bokowski's **omawin** [Bok99] can be used for low rank oriented matroid visualization. In order to have correct combinatorial results, arbitrary precision arithmetic is essential.

Stephenson's **CirclePack** [Ste02] can create, manipulate, store, and display circle packings.

Lattice points in convex polytopes are related to volume computations and, via Gröbner bases, to problems in commutative algebra. A specialized implementation in this area is **Ehrhart** by Clauss, Loechner, and Wilde [CLW99]. There is also **intpoint** by Emiris [Emi01] in the context of mixed volumes. Various volume computation algorithms for polytopes, using exact and floating point arithmetic, are implemented in **vinci** by Büeler, Enge, and Fukuda [BEF03].

Dynamic geometry software allows the creation of geometrical constructions from points, lines, circles, and so on, which later can be rearranged interactively. Objects depending, e.g., on intersections, change accordingly. Among other features, such systems can be used for visualization purposes and, in particular, also for working with polygonal linkages. Commercial products include Laborde

and Bellemain's **Cabri** [LB93] as well as **Cinderella** [RGK] by Kortenkamp and Richter-Gebert. Current dynamic geometry software systems seem to be restricted to planar constructions.

Graph theory certainly is a core topic in discrete mathematics and therefore naturally plays a role in discrete and computational geometry. There is an abundance of algorithms and software packages, but they are not especially well suited to geometry, and so they are skipped here. Often symmetry properties of geometric objects can be reduced to automorphisms of certain graphs. While the complexity status of the graph isomorphism problem remains open, McKay's **nauty** [McK03] works quite well for many practical purposes.

Theorem 14.2.3 establishes the existence of a center point in any Lebesgue measurable subset of \mathbb{R}^d . The discrete analogue has a nice approximative algorithmic solution [CEM⁺96] which has been implemented by Clarkson [Claa].

LIBRARIES

Ehrhart polynomials and integer points in polytopes are also accessible via Loechner's **PolyLib** [Loe02].

MODULES FOR GENERAL PURPOSE SYSTEMS

Parts of **TOPCOM**'s [Ram03] functionality are also available in De Loera's **Maple** package **Puntos** [DL96].

ADDITIONAL WEB PAGES

Huson's Web page [Hus03] contains information on tilings and related software.

Circle packings are related to several other topics in discrete geometry and complex analysis. Boll maintains a Web page [Bol00] on the subject with additional code and many links.

For polyominoes, see Eppstein's Geometry Junkyard [Epp03a] and [Chapter 15](#).

The **LattE** project by De Loera, Hemmecke et al. [LH⁺] offers an email service for computing lattice points in convex polytopes.

64.1.2 POLYTOPES AND POLYHEDRA

In this section we discuss software related to the computational study of convex polytopes. The distinction between polytopes and unbounded polyhedra is not essential since, up to a projective transformation, each polyhedron is the product of an affine subspace and a polytope.

A key problem in the algorithmic treatment of polytopes is the convex hull problem, which is addressed in the next section.

STAND-ALONE SOFTWARE

polymake [GJ03] is a comprehensive framework for dealing with polytopes in terms of vertex or facet coordinates as well as on the combinatorial level. The system offers a wide functionality which is augmented by interfacing to many other programs operating on polytopes. Among the combinatorial algorithms implemented is the recent method for enumerating all the faces of a polytope given in terms of the

vertex-facet incidences by Kaibel and Pfetsch [KP02].

Triangulations of polytopes can be rather large and intricate. Rambau's TOPCOM [Ram03] is primarily designed to examine the set of all triangulations of a given polytope (or arbitrary point configurations). Pfeifle and Rambau [PR03] combined TOPCOM with `polymake` to compute secondary polytopes; see also Section 17.6.

The combinatorial equivalence of polytopes can be reduced to a graph isomorphism problem. As mentioned above, graph isomorphism can be checked by McKay's `nauty` [McK03].

The Geometry Center's `Geomview` [Geo02] and `JavaView` [PKP⁺02], by Polthier et al., can both be used for (much more than) the visualization of 3-polytopes and (Schlegel diagrams of) 4-polytopes.

LIBRARIES

`PolyLib` [Loe02] by Loechner is a library for working with rational polytopes; it is primarily designed for computing Ehrhart polynomials.

`polymake` [GJ03] comes with an C++ template library that is compatible with the Standard Template Library (STL). This allows one to access all the functionality, including the interfaced programs, from the programmer's own code. Further, the library offers a variety of container classes suitable for the manipulation of polytopes.

MODULES FOR GENERAL PURPOSE SYSTEMS

`convex` by Franz [Fra03] is a package for the investigation of rational polytopes and polytopal fans in `Maple`.

64.1.3 FUNDAMENTAL GEOMETRIC OBJECTS

The computation of convex hulls and Delaunay triangulations/Voronoi diagrams is of key importance. For correct combinatorial output it is crucial to rely on arbitrary-precision arithmetic. On the other hand, some applications, e.g., volume computation, are content with floating point arithmetic for approximate results. Some algorithmically more advanced but theoretically yet basic topics in this section are related to topology and real algebraic geometry.

In our terminology the *convex hull problem* asks for enumerating the facets of the convex hull of finitely many points in \mathbb{R}^d . The dual problem of enumerating the vertices and extremal rays of the intersection of finitely many halfspaces is equivalent by means of cone polarity. There is the related problem of deciding which points among a given set are extremal, that is, vertices of the convex hull. This can be solved by means of linear optimization.

STAND-ALONE SOFTWARE

`XYZGeoBench` for the Macintosh is an implementation of many fundamental algorithms by Schorn [Sch99]. For many of these algorithms there is an animated visualization.

Many convex hull algorithms are known, and there are several implementations. However, there is currently no algorithm for computing the convex hull which is

polynomial in the combined input and output size, unless the dimension is considered constant. The behavior of each known algorithm depends greatly on the specific combinatorial properties of the polytope on which it is working. One way of summarizing the computational results from Avis, Bremner, and Seidel [ABS97] and [Jos03] is: Essentially for each known algorithm there is a family of polytopes for which the given algorithm is superior to any other, and there is a second family for which the same algorithm is inferior to any other. For these families of polytopes we do have a theoretical, asymptotic analysis which explains the empirical results; see [Chapter 22](#). Moreover, there are families of polytopes for which none of the existing algorithms performs well. Which algorithm or implementation works best for certain purposes will thus depend on the class of polytopes which is typical in those applications. For an overview of general convex hull codes see Table 64.1.1.²

Additionally, there are a few specialized codes: **Zerone** by Lübecke [Lüb99] is designed to compute the vertices of a polytope with 0/1-coordinates from an inequality description by iteratively solving linear programs. There is a parallel computation version of **lrs** based on Marzetta's **ZRAM** library [Mar98]. The same library is also used in Fukuda's very recent code **rs_zotope** [Fuk02] which enumerates (also in parallel) the vertices of a zonotope defined by a vector configuration.

TABLE 64.1.1 Overview of convex hull codes.

Exact arithmetic		
PROGRAM	ALGORITHM	REMARKS
beneath_beyond	Beneath-beyond method [Ede87, 8.3.1]	Part of polymake [GJ03]
cddr+ [Fuk03a]	Dual Fourier-Motzkin elimination [Zie95, 1.2]	
ch3d [Emi01]	Beneath-beyond method	Dimension ≤ 3
lrs [Avi01]	Reverse search [AF92]	
porta [CL03]	Fourier-Motzkin elimination	
pd [Mar97]	Primal-dual method [BFM98]	

Non-exact arithmetic		
PROGRAM	ALGORITHM	REMARKS
2dch [Cla96]	Horizontal sweep	dimension 2
cddf+ [Fuk03a]	Dual Fourier-Motzkin elimination	
chD [Emi01]	Beneath-beyond method	
hull [Clab]	Randomized incremental [CMS93]	Assumes input in gen. pos.; Exact computation unless Overflow signaled
qhull [BDH01b]	Quickhull [BDH96]	

The computation of Delaunay triangulations in d dimensions can be reduced to a $(d+1)$ -dimensional convex hull problem; see [Section 23.1](#). Thus, in principle, each of the convex hull implementations can be used to generate Voronoi diagrams. Additionally, however, some codes directly support Voronoi diagrams, no-

²We call an implementation *exact* if it, intentionally (but there may be programming errors, of course), gives correct results for *all* possible inputs. The non-exact convex hull codes use floating-point arithmetic or more advanced methods, but for each of them input is known which makes them fail. The quality of the output of the non-exact convex hull codes varies considerably.

tably Clarkson's `hull` [Clab], `qhull` by Barber, Dobkin, and Huhdanpaa [BDH01b], and, among the programs with exact rational arithmetic, `lrs` by Avis [Avi01].

The following codes are specialized for 2-dimensional Voronoi diagrams: Shewchuk's `Triangle` [She96] and Fortune's `voronoi` [For01]. See also `cdt` by Lischinski [Lis98] for incremental constrained 2-dimensional Delaunay triangulation. For 3-dimensional problems there is `Detri` by Mücke [Müc95] and `tess` by Hazelwood [Haz94]. Delaunay triangulations and, in particular, constrained Delaunay triangulations, play a significant role in meshing. Therefore, several of the Voronoi/Delaunay packages also have features for meshing and vice versa.

Alpha shapes form a technique to describe subsets of Euclidean space by means other than convex hulls of finitely many points ([Chapter 63](#)). There is a dedicated software package named `Alpha shapes` by Fu, Edelsbrunner et al. [FE⁺96] which deals with 2- and 3-dimensional alpha shapes in exact arithmetic. `hull` computes alpha shapes in arbitrary dimension.

For the special case of triangulating a simple polygon ([Chapter 26](#)), there is Seidel's randomized algorithm with almost linear running time. The implementation by Narkhede and Manocha is part of the `Graphics Gems` [KHP⁺95, Part VI]. This archive and also Skiena's collection of algorithms [Ski01] contain more specialized code and algorithms for polygons.

Mesh generation is a vast area with numerous applications; see [Chapter 25](#). This is reflected by the fact that there is an abundance of commercial and noncommercial implementations. We mention only a few. From the theoretical point of view the main categories are formed by 2-dimensional triangle meshes, 2-dimensional quadrilateral meshes, 3-dimensional tetrahedral meshes, 3-dimensional cubical (also called hexahedral) meshes, and other structured meshes. A focus on the applications leads to entirely different categories, which here is completely ignored. `Triangle` produces triangle meshes. `QMG` is a program for quadtree/octree 2- and 3-dimensional finite element meshing written by Vavasis [Vav00]. `CUBIT` [cub03] can do many different variants of 2- and 3-dimensional meshing; it is a commercial product which is free for scientific use. Note that, depending on the context, triangle or tetrahedra meshes are also called triangulations.

In applications geometric objects are sometimes given as point clouds meant to represent a curve or surface. With the introduction of 3D-scanners and similar devices, appropriate techniques and related software became increasingly important. Obviously, this problem is directly related to mesh generation. `Cocone` by Dey et al. [DGG⁺02] and `Power Crust` by Amenta, Choi, and Kolluri [ACK02] are designed to produce "water tight" surfaces; see [Chapter 30](#). `Studio` [stu02] is a commercial product dedicated to generating meshes from 3D-scans.

`VisPak` by Wismath et al. [W⁺98] is built on top of `LEDA` and can be used for the generation of visibility graphs of line segments and several kinds of polygons.

Smallest enclosing balls of a point set in arbitrary dimension can be computed with Gärtner's `Miniball` [Gär99b].

Recent years saw an increasing use of methods from combinatorial topology in discrete and computational geometry. A basic operation is to compute the homology of a finite simplicial complex. Although polynomial time methods (in the size of the boundary matrices) are known for most problems, the (worst case exponential) elimination methods seem to be superior in practice; see Dumas et al. [HDSW03]. Implementations include `homology` by Heckenbach [Hec98] (see also the more recent implementation as a GAP package by Dumas et al. [DHS⁺03]) and `t_homology` which is part of `polymake`'s [GJ03] combinatorial topology module `TOPAZ`.

As for the opposite direction, more computational tools become available for the study of topological objects: Lutz's **BISTELLAR** [Lut02] is the implementation of a heuristic approach to find (vertex) minimal triangulations of a given space by applying bistellar flips. **SnapPea** by Weeks [Wee00] is a program for creating and examining hyperbolic 3-manifolds. **Geomview**'s [Geo02] extension package **Maniview** can be used to visualize 3-manifolds from within.

The computer algebra system **Magma** by Cannon et al. [C⁺03] has some support for real algebraic geometry. Visualization of curves and surfaces can be done with **surf** by Endrass [End03] and **spicy** [Lab03] by Labs.

LIBRARIES

cddlib [Fuk03b] and **lrslib** [Avi01] are the C library versions of Fukuda's **cdd** and Avis' **lrs**, respectively. They offer exact convex hull computation and exact linear optimization. **cddlib** uses the **GMP** [gmp03] arithmetic, while **lrslib** can be compiled with **GMP** arithmetic, but also has its own implementation. **polymake**'s [GJ03] functionality is available as a C++ library. This includes interfaces to **cdd/cddlib** and **lrs/lrslib**.

There is a C library version of **qhull** [BDH01b] which performs convex hulls and Voronoi diagrams in floating point arithmetic. Moreover, **cddlib** and **polymake** also have a limited support for floating point arithmetic.

The computation of Voronoi diagrams, arrangements, and related information is a particular strength of **CGAL** [F⁺02] and **LEDA** [led]. See [Chapter 65](#).

For triangle meshes in \mathbb{R}^3 there is the **GNU Triangulated Surface Library** [gts03] written in C. Its functionality comprises dynamic Delaunay and constrained Delaunay triangulations, robust set operations on surfaces, and surface refinement and coarsening for the control of level-of-detail.

Bhaniramka and Wenger have a set of C++ classes for the construction of isosurface patches in convex polytopes of arbitrary dimension [BW03]. These can be used in marching cubes like algorithms for isosurface construction.

MODULES FOR GENERAL PURPOSE SYSTEMS

Plain **Maple** [map03] and **Mathematica** [mat03a] only offer 2-dimensional convex hulls and Voronoi diagrams. Higher dimensional convex hulls can be computed via the **Maple** package **convex** [Fra03].

Mitchell [Mit] has implemented some of his algorithms related to mesh generation in **Matlab** [mat03b]. The finite element meshing program **QMG** by Vavasis can also be used with **Matlab**.

The **REDUCE** [H⁺99] package **REDLOG** by Dolzmann and Sturm [DS99] can do quantifier elimination over the reals (and other domains).

ADDITIONAL WEB PAGES

Emiris maintains a Web page [Emi01] with several programs which address problems related to convex hull computations and applications in elimination theory.

Web based surface reconstruction is available from INRIA's project page [CSD02].

A Web page [Owe03] by Owen contains a quite comprehensive survey on software related to meshing. See also Schneiders' page [Sch].

Morris provides interactive visualization of algebraic surfaces on-line: The pro-

gram **SingSurf** [Mor03] uses **JavaView** [PKP⁺02] for the visualization.

The recently announced **EXACUS** project [M⁺02a] deals with the exact computation of arrangements of planar algebraic curves as well as surfaces in \mathbb{R}^3 . Currently there are only partial prototype implementations.

64.1.4 GEOMETRIC DATA STRUCTURES AND SEARCHING

LIBRARIES

Geometric data structures form the core of the C++ libraries **CGAL** [F⁺02] and **LEDA** [led]. The algorithms implemented include several different techniques for point location, collision detection, and range searching. See [Chapter 65](#).

As already mentioned above, graph theory plays a role for some of the more advanced geometric algorithms. Several libraries for working with graphs have been developed over the years. It is important to mention in this context the **Boost Graph Library** [SLL02]. This is part of a general effort to provide free peer-reviewed portable C++ source libraries which extend the **STL**.

ZRAM by Marzetta [Mar98] is a library of parallel search algorithms and the corresponding data structures. The implementation is application-independent and machine-independent. It is used in parallel versions of the convex hull codes **lrs** by Avis [Avi01] and **rs_topo** (for zonotopes) by Fukuda [Fuk02].

64.1.5 APPLICATIONS

Applications of computational geometry are abundant and so are the related software systems. Here we list only very few items which may be of interest to a general audience.

STAND-ALONE SOFTWARE

For linear programming problems, essential choices for algorithms include Simplex type algorithms or interior point methods. While commercial solvers tend to offer both, the freely available implementations seem to be restricted to either one. Additionally, there are implementations of a few special algorithms for low dimensions which belong to neither category.

Exact rational linear programming can be done with **cdd** [Fuk03a]. It uses either a dual simplex algorithm or the criss-cross method. An alternative exact linear programming code is **lrs** [Avi01] which implements a primal simplex algorithm.

SoPlex by Wunderling et al. [W⁺02] implements the revised Simplex algorithm both in primal and dual form. For an implementation of interior point methods see **PCx** by Czyzyk et al. [CMW⁺98]. These codes rely on floating-point arithmetic.

CPLEX [cpl02], **OSL** [osl01], and **XPress** [xpr03] are widespread commercial solvers for linear, integer, and mixed integer programming. Each program offers a wide range of optimization algorithms. However, none of the commercial products can do exact rational linear optimization.

Clarkson's **opt** [Cla95] is the floating point implementation of a Las Vegas type algorithm which runs in expected linear time (for fixed dimension). See also Hohmeyer's code **linprog** [Hoh96] for an implementation of Seidel's algorithm.

These algorithms are described in Section 45.4.

Another topic with many applications is graph drawing. **GraphViz** [gra02] is an extensible package which offers tailor made solutions for a wide range of applications in this area. **Tulip** [AB⁺03] specializes in the visualization of large graphs. For commercial graph drawing software see **yFiles** [yfi03]; previous versions of yFiles used the **LEDA** and **AGD** [M⁺02c] libraries.

LIBRARIES

cddlib [Fuk03b] and **lrs** offer C libraries for exact LP solving. **CPLEX**, **OSL**, **PCx**, and **XPress** can also be used as C libraries, while **SoPlex** has a C++ library version. Other free C libraries for linear and mixed integer programming include **GLPK** [Mak03] and **lp_solve** [Ber03].

AGD [M⁺02c] and **GDT toolkit** [gdt00] both are C++ libraries for graph drawing which are built on top of **LEDA**. Both are free for academic use.

In order to meet certain quality criteria post-processing of mesh data is important. **QSLIM** by Garland [Gar99a] is a C++ library for the automatic simplification of polygonal surfaces with the goal to reduce the number of polygons.

MODULES FOR GENERAL PURPOSE SYSTEMS

The linear optimization package **PCx** comes with an interface to **Matlab** [mat03b].

ADDITIONAL WEB PAGES

For more information about linear programming there is an FAQ [Fou03] maintained by Fourer.

Recently, IBM started to foster various open source software projects; see the COIN Web page [coi] for optimization related software packages.

One of the topics related to computational geometry that we have not discussed in this survey is computer graphics. We refer the reader to O'Rourke's FAQ for the Usenet newsgroup **comp.graphics.algorithms** [O'R03].

The **Prisme** project [B⁺01] studies a variety of applications of computational geometry methods. **Galaad** [M⁺02b] is a related project with a focus on curves and surfaces. See also **EXACUS** [M⁺02a].

64.2 FEATURES OF SELECTED SOFTWARE SYSTEMS

All the software packages listed here have been mentioned previously. In many cases, however, we list features not accounted for so far.

AGD [M⁺02c]: C++ library for graph drawing based on **LEDA**. AGD offers many different layout algorithms, including planarization based methods, planar straight-line methods, hierarchical layouts, and various specialized applications. Graph layout visualization possible via **Graphlet** [B⁺99], **LEDA/GraphWin**, and other systems. Free for academic use. Also available for Windows 95/98/NT.

Boost Graph Library [SLL02]: C++ library for graphs and graph algorithms.

The library is *generic* in the sense that the implementations of the algorithms do not rely on specific implementation details of the data structures. It is developed

in the spirit of the **Standard Template Library** (STL) as described in the ANSI C++ Standard. Similar software design concepts are used in **CGAL** and **polymake**.

cdd [Fuk03a, Fuk03b]: convex hull code which is based on the double description method which is dual to Fourier-Motzkin elimination. It also implements a dual simplex algorithm and the criss-cross method for linear optimization. **cdd** comes as a stand-alone program; its C library version is called **cddlib**. The user can choose between exact rational arithmetic (based on the **GMP**) or floating point arithmetic.

Cinderella [RGK]: commercial dynamic geometry software written in **Java**. It supports standard constructions with point, lines, and quadrics. **Cinderella** is based on a sound mathematical model by computing in the complex projective plane. Special features include loci of moving points which are constrained by a geometric construction and a randomized theorem prover. Runs on platforms supporting **Java**. Constructions can be integrated into Web pages as applets.

Cocone [DGG⁺02] is a set of programs related to the reconstruction of surfaces from point clouds in \mathbb{R}^3 via discrete approximation to the medial axis transform: **Tight Cocone** produces “water-tight” surfaces from arbitrary input, while **Cocone/SuperCocone** is responsible for detecting the surface’s boundary. **Geomview** output. Based on **CGAL** and **LEDA**. Not available for commercial use.

Computational Geometry in C [O'R98, O'R00]: collection of C and **Java** programs including 2- and 3-dimensional convex hull codes, Delaunay triangulations, and segment intersection.

CUBIT [cub03] is a commercial meshing tool for surfaces and 3-dimensional objects to be used in finite element analysis. Mesh generation algorithms include quadrilateral and triangular paving, 2- and 3-dimensional mapping, hex sweeping and multi-sweeping, and others. There is also a Windows version. Free for noncommercial research.

Geomview [Geo02] is a tool for interactive visualization. It can display objects in hyperbolic and spherical space as well as Euclidean space. **Geomview** comes with several external modules for specific visualization purposes. The user can write additional external modules in C. **Geomview** can be used as a visualization back end, e.g., for **Maple** [map03] and **Mathematica** [mat03a]. The extension package **Maniview** can visualize 3-manifolds.

GraphViz [gra02]: package with various graph layout tools. This includes hierarchical layouts and spring embedders. The system comes with a customizable graphical interface. Also runs on Windows.

GMP [gmp03]: The **GNU Multiprecision Library** is the standard implementation for long integer, exact rational, and arbitrary precision floating-point arithmetic. Four different algorithms for the multiplication of integers are implemented including the asymptotically optimal method due to Schönhage and Strassen [SS71, Sch82]. Highly optimized back-ends for many common microprocessors written in assembler.

hull [Clab] computes the convex hull of a point set in general position. The program can also compute Delaunay triangulations, alpha shapes, and volumes of Voronoi regions. The program uses exact machine floating-point arithmetic, and it signals overflow. **Geomview** output.

JavaView [PKP⁺02]: visualization package for Euclidean, spherical, and hyperbolic space which — as the name suggests — is written in Java. Wide functionality with a focus on applications in differential geometry. There is also an applet version `jv_lite` which allows for interactive visualization embedded in HTML pages. `JavaViewLib` is an add-on package for interfacing with `Maple`. Likewise, access to `Mathematica` is supported via `J/Link`. Runs on platforms supporting Java.

lrs [Avi01]: convex hull code based on the reverse search algorithm due to Avis and Fukuda [AF92]. Exact rational arithmetic, e.g. via the `GMP`. In addition to convex hull computations, `lrs` can do linear optimization (via a primal Simplex algorithm), volume computation, and Voronoi diagrams. Also comes as a C library.

nauty [McK03] can compute a permutation group representation of the automorphism group of a given finite graph. As one interesting application this gives rise to an effective method for deciding whether two graphs are isomorphic or not. Such a check for isomorphism can be performed directly.

PolyLib [Loe02] — a library of polyhedral functions. Allows for basic geometric operations on parametrized polyhedra. As a key feature `PolyLib` can compute Ehrhart polynomials, which permits counting the number of integer points in a given polytope.

polymake [GJ03] is a system for examining the geometrical and combinatorial properties of polytopes. It offers convex hull computation, standard constructions, and visualization. Some of the functionality relies — via interfaces — on external programs including `cdd`, `Geomview`, `Graphlet`, `JavaView`, and `lrs`. STL compatible C++ library; computations in exact rational arithmetic based on `GMP`. Separate module `TOPAZ` for simplicial complexes. Its functionality so far includes simplicial homology computation and intersection forms of 4-manifolds.

Power Crust [ACK02] performs surface reconstruction via a discrete approximation of the medial axis transform. The key concept for the algorithm are *power diagrams*, which are certain weighted Voronoi diagrams. **Power Crust** uses `hull` for Voronoi diagrams, and it offers `Geomview` output.

qhull [BDH01b] computes convex hulls, Delaunay triangulations, Voronoi diagrams, furthest-site Delaunay triangulations, and furthest-site Voronoi diagrams. The algorithm implemented is Quickhull [BDH96]. `qhull` uses floating-point arithmetic only, but the authors incorporated several heuristics to improve the quality of the output. This is discussed in detail on a special Web page [BDH01a] in `qhull`'s documentation; it is an important source for everyone interested in using or implementing computational geometry software based on floating-point arithmetic.

SoPlex [W⁺02] — The sequential object-oriented simplex (C++) class library. Also available as stand-alone. `SoPlex` implements the revised Simplex linear optimization algorithm in primal and dual form.

TOPCOM [Ram03]: package for examining point configurations via oriented matroids. The main purpose is to investigate the set of all triangulations of a given point configuration. Symmetric point configurations can be treated more efficiently if the user provides information about automorphisms. `TOPCOM` can check whether a given triangulation is regular.

Triangle [She96] produces 2-dimensional meshes. It generates exact Delaunay triangulations, constrained Delaunay triangulations, and quality conforming Delaunay triangulations. The latter can be generated while avoiding small angles, and are thus suitable for finite element analysis. There is an add-on **ShowMe** for the visualization of triangulations.

vinci [BEF03] can be seen as an experimental framework for comparing volume computation algorithms. Exact and floating-point arithmetic. Implemented are Cohen & Hickey-triangulations [CH79], Delaunay triangulations (via `cdd` or `qhull`), and others.

XYZGeoBench [Sch99] is an interactive program for the Apple Macintosh (OS version $\geq 6.0.5$). Many basic algorithms for planar (and a few higher-dimensional) problems are implemented and can be animated.

REFERENCES

- [AB⁺03] D. Auber, M. Bertrand, et al. Tulip, Version 1.2.4.
<http://www.tulip-software.org>, 2003.
- [ABS97] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Comput. Geom.*, 7(5–6):265–301, 1997.
- [ACK02] N. Amenta, S. Choi, and R.K. Kolluri. Power Crust, Unions of Balls, and the Medial Axis Transform, Version 1.2. <http://www.cs.utexas.edu/users/amenta/powercrust>, 2002.
- [AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992. ACM Sympos. Comput. Geom. (North Conway, NH, 1991).
- [Ame97] N. Amenta. Directory of Computational Geometry Software.
<http://www.geom.umn.edu/software/cglist/welcome.html>, 1997.
- [Arv91] J. Arvo, editor. *Graphics Gems II*. Academic Press, Boston, 1991.
- [Avi01] D. Avis. lrs, lrslib, Version 4.1. <http://cgm.cs.mcgill.ca/~avis/C/lrs.html>, 2001.
- [B⁺99] F.J. Brandenburg et al. Graphlet, Version 5.0.1.
<http://www.infosun.fmi.uni-passau.de/Graphlet>, 1999.
- [B⁺01] J.-D. Boissonnat et al. Prisme. <http://www-sop.inria.fr/prisme>, 2001.
- [BDH96] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Software*, 22:469–483, 1996.
- [BDH01a] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. Imprecision in qhull.
<http://www.thesa.com/software/qhull/html/qh-impre.htm>, 2001.
- [BDH01b] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. qhull, Version 3.1.
<http://www.thesa.com/software/qhull>, 2001.
- [BEF03] B. Büeler, A. Enge, and K. Fukuda. vinci, Version 1.0.5.
<http://www.lix.polytechnique.fr/Labo/Andreas.Engel/Vinci.html>, 2003.
- [Ber03] M. Berkelaar. Ipsolve, Version 4.0. ftp://ftp.ics.ele.tue.nl/pub/Ip_solve, 2003.
- [BFM98] D. Bremner, K. Fukuda, and A. Marzetta. Primal-dual methods for vertex and facet enumeration. *Discrete Comput. Geom.*, 20:333–357, 1998.

- [Bok99] J. Bokowski. omawin, Version 1.0.0. <http://www.mathematik.tu-darmstadt.de/~bokowski/omawin.html>, 1999.
- [Bol00] D. Boll. Optimal packing of circles and spheres. <http://www.frii.com/~davejen/packing.html>, 2000.
- [BW03] P. Bhaniramka and R. Wenger. Isotable, Version 2.0. <http://www.cis.ohio-state.edu/research/graphics/isotable>, 2003.
- [C⁺03] J. Cannon et al. Magma, Version 2.10. <http://magma.maths.usyd.edu.au/magma>, 2003.
- [CEM⁺96] K.L. Clarkson, D. Eppstein, G.L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with iterative radon points. *Internat. J. Comput. Geom. Appl.*, 6:357–377, 1996.
- [CH79] J. Cohen and T. Hickey. Two algorithms for determining volumes of convex polyhedra. *J. Assoc. Comput. Mach.*, 26:401–414, 1979.
- [CL03] T. Christof and A. Löbel. Porta, Version 1.4.0. <http://www.zib.de/Optimization/Software/Porta>, 2003.
- [Claa] K.L. Clarkson. Center Point. <http://cm.bell-labs.com/who/clarkson/center.html>.
- [Clab] K.L. Clarkson. Hull, Version 1.0. <http://netlib.bell-labs.com/netlib/voronoi/hull.html>.
- [Cla95] K.L. Clarkson. opt. <http://cm.bell-labs.com/who/clarkson/lp2.html>, 1995.
- [Cla96] K.L. Clarkson. 2dch. <http://cm.bell-labs.com/who/clarkson/2dch.c>, 1996.
- [CLW99] P. Clauss, V. Loehner, and D. Wilde. The Ehrhart polynomials and parametric vertices program, Version 4.10. <http://icps.u-strasbg.fr/Ehrhart/program/program.html>, 1999.
- [CMS93] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom.*, 3:185–212, 1993.
- [CMW⁺98] J. Czyzyk, S. Mehrotra, M. Wagner, and S. Wright. PCx, Version 1.1. <http://www-fp.mcs.anl.gov/otc/Tools/PCx>, 1998.
- [coi] COmputational INfrastructure for Operations Research. International Business Machines, <http://www.ibm.com/developerworks/oss/coin>.
- [cpl02] CPLEX, Version 8.0. ILOG, Inc., <http://www.ilog.com/products/cplex>, 2002.
- [CSD02] D. Cohen-Steiner and F. Da. Surface Reconstruction. <http://cgal.inria.fr/Reconstruction>, 2002.
- [cub03] CUBIT, Version 8.0.1. Sandia National Laboratories, <http://endo.sandia.gov/cubit>, 2003.
- [cyg03] Cygwin, Version 1.3.22-1. Red Hat, <http://www.cygwin.com>, 2003.
- [DGG⁺02] T.K. Dey, J. Giesen, S. Goswami, J. Hudson, and W. Zhao. Cocone softwares. <http://www.cis.ohio-state.edu/~tamaldey/cocone.html>, 2002.
- [DHS⁺03] J.-G. Dumas, F. Heckenbach, B.D. Saunders, and V. Welker. Simplicial Homology, a (proposed) GAP share package, Version 1.4.1. <http://www.eecis.udel.edu/~dumas/Homology>, 2003.
- [DL96] J.A. De Loera. Puntos, Version 3. http://www.math.ucdavis.edu/~deloera/RECENT_WORK/puntos2000, 1996.
- [DS99] A. Dolzmann and T. Sturm. REDLOG, Version 2.0. <http://www.fmi.uni-passau.de/~redlog>, 1999.

- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, Berlin, 1987.
- [Emi01] I.Z. Emiris. Computational geometry. http://www-sop.inria.fr/galaad/logiciels/emiris/soft_geo.html, 2001.
- [End03] S. Endrass. surf, Version 1.0.4. <http://surf.sourceforge.net>, 2003.
- [Epp03a] D. Eppstein. Polyominoes and other animals. <http://www.ics.uci.edu/~eppstein/junkyard/polyomino.html>, 2003.
- [Epp03b] D. Eppstein. The Geometry Junkyard. <http://www.ics.uci.edu/~eppstein/junkyard>, 2003.
- [Eri99] J. Erickson. Computational Geometry Code. <http://compgeom.cs.uiuc.edu/~jeffe/compgeom/code.html>, 1999.
- [F⁺02] A. Fabri et al. CGAL, Version 2.4. <http://www.cgal.org>, 2002.
- [FE⁺96] P. Fu, H. Edelsbrunner, et al. Alpha shapes, Version 4.1. <http://www.alphashapes.org/alpha>, 1996.
- [For01] S.J. Fortune. voronoi. <http://cm.bell-labs.com/who/sjf>, 2001.
- [Fou03] R. Fourer. Linear Programming Frequently Asked Questions. <http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html>, 2003.
- [Fra03] M. Franz. convex—a Maple package for convex geometry, Version 1.0 alpha. <http://www-fourier.ujf-grenoble.fr/~franz/convex>, 2003.
- [Fuk02] K. Fukuda. RS_TOPE, Version 020713. <http://www.cs.mcgill.ca/~fukuda/download/mink>, 2002.
- [Fuk03a] K. Fukuda. cdd+, Version 0.77beta. http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html, 2003.
- [Fuk03b] K. Fukuda. cddlib, Version 0.93. http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html, 2003.
- [Gar99a] M. Garland. QSlim, Version 2.0. <http://graphics.cs.uiuc.edu/~garland/software/qslim.html>, 1999.
- [Gär99b] B. Gärtner. Miniball, Version 1.4. <http://www.inf.ethz.ch/personal/gaertner/miniball.html>, 1999.
- [gdt00] GDToolkit, Version 3.0. Dipartimento di Informatica e Automazione, Università di Roma Tre, Rome, Italy, <http://www.dia.uniroma3.it/~gdt>, 2000.
- [Geo02] Geomview, Version 1.8.1. The Geometry Center, <http://www.geomview.org>, 2002.
- [GJ03] E. Gawrilow and M. Joswig. polymake, Version 2.0. <http://www.math.tu-berlin.de/polymake>, 2003.
- [Gla93] A.S. Glassner, editor. *Graphics Gems*. Academic Press, Boston, 1993.
- [gmp03] GNU multiprecision library, Version 4.1.2. <http://www.swox.com/gmp>, 2003.
- [gra02] GraphViz, Version 1.8.5. AT&T Lab – Research, <http://www.research.att.com/sw/tools/graphviz>, 2002.
- [gts03] GNU Triangulated Surface Library, Version 0.7.1. <http://gts.sourceforge.net>, 2003.
- [H⁺99] A.C. Hearn et al. REDUCE, Version 3.7. <http://www.uni-koeln.de/REDUCE>, 1999.
- [Haz94] C. Hazlewood. tess. ftp://ftp.geom.umn.edu/pub/contrib/comp_geom, 1994.

- [HDSW03] F. Heckenbach, J.-G. Dumas, B.D. Saunders, and V. Welker. Computing simplicial homology based on efficient Smith Normal Form algorithms. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 177–206. Springer-Verlag, Berlin, 2003.
- [Hec94] P.S. Heckbert, editor. *Graphics gems IV*. Academic Press, Boston, 1994.
- [Hec98] F. Heckenbach. homology, Version 3.0. <http://www.mi.uni-erlangen.de/~heckenb>, 1998.
- [Hoh96] M. Hohmeyer. linprog. <http://www.cs.sunysb.edu/~algorithm/implement/linprog/implement.shtml>, 1996.
- [Hus03] D.H. Huson. Home Page. http://www-ab.informatik.uni-tuebingen.de/people/huson/old_homepage/Welcome.html, 2003.
- [Jos03] M. Joswig. Beneath-and-beyond revisited. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 1–21. Springer-Verlag, Berlin, 2003.
- [KHP⁺95] D. Kirk, P.S. Heckbert, A.W. Paeth, et al. Graphics gems. <ftp://ftp-graphics.stanford.edu/pub/Graphics/GraphicsGems>, 1995.
- [Kir92] D. Kirk, editor. *Graphics gems III*. Academic Press, Boston, 1992.
- [KP02] V. Kaibel and M.E. Pfetsch. Computing the face lattice of a polytope from its vertex-facet incidences. *Comput. Geometry*, 23:281–290, 2002.
- [Lab03] O. Labs. Spicy, Version 0.61b. <http://enriques.mathematik.uni-mainz.de/spicy>, 2003.
- [LB93] J.-M. Laborde and F. Bellemain. Cabri Geometry II. <http://www.cabri.net/index-e.html>, 1993.
- [led] LEDA, Version 4.3. Algorithmic Solution Software GmbH, <http://www.algorithmic-solutions.com/enleda.htm>.
- [LH⁺] J.A. De Loera, R. Hemmecke, et al. Latte. <http://www.math.ucdavis.edu/~latte>.
- [Lis98] D. Lischinski. cdt. <http://www.cs.huji.ac.il/~danix/code/cdt.tar.gz>, 1998.
- [Loe02] V. Loehner. PolyLib - A library of polyhedral functions, Version 5.11.1. <http://icps.u-strasbg.fr/PolyLib>, 2002.
- [Lüb99] M.E. Lübecke. Zerone, Version 1.8.1. <http://www.math.nat.tu-bs.de/mo/research/zerone.html>, 1999.
- [Lut02] F.H. Lutz. BISTELLAR, Version 05/02. <http://www.math.TU-Berlin.DE/diskregeom/stellar/bistellar.tar.gz>, 2002.
- [M⁺02a] K. Mehlhorn et al. EXACUS — Efficient and Exact Algorithms for Curves and Surfaces. <http://www.mpi-sb.mpg.de/projects/EXACUS>, 2002.
- [M⁺02b] B. Mourrain et al. Galaad. <http://www-sop.inria.fr/galaad>, 2002.
- [M⁺02c] P. Mutzel et al. AGD, Version 1.2. <http://aragorn.ads.tuwien.ac.at/AGD>, 2002.
- [Mak03] A. Makhorin. GNU Linear Programming Kit, Version 4.0. <http://www.gnu.org/software/glpk/glpk.html>, 2003.
- [map03] Maple, Version 9. Waterloo Maple, Inc., <http://www.maplesoft.com>, 2003.
- [Mar97] A. Marzetta. pd. <http://www.cs.unb.ca/~bremner/pd>, 1997.
- [Mar98] A. Marzetta. ZRAM. <http://www.cs.unb.ca/~bremner/zram>, 1998.
- [mat03a] Mathematica, Version 5. Wolfram Research, Inc., <http://www.wolfram.com>, 2003.
- [mat03b] Matlab, Version 6.5. The Mathworks, Inc., <http://www.mathworks.com/products/matlab>, 2003.

- [McK03] B. McKay. nauty, Version 2.2 (beta5). <http://cs.anu.edu.au/~bdm/nauty>, 2003.
- [Mit] S.A. Mitchell. Computational Geometry Triangulation Results. <http://endo.sandia.gov/~samitch/csstuff/csguide.html>.
- [Mor03] R. Morris. SingSurf, Version 0.78615138. <http://www.comp.leeds.ac.uk/pfaf/lsmpl/SingSurf.html>, 2003.
- [Müc95] E.P. Mücke. Detri, Version 2.6a. <http://www.geom.umn.edu/software/cglist/GeomDir>, 1995.
- [mup03] MuPAD, Version 2.5.2. SciFace Software GmbH & Co. KG, <http://www.mupad.de>, 2003.
- [O'R98] J. O'Rourke. *Computational Geometry in C*, 2nd edition. Cambridge University Press, 1998.
- [O'R00] J. O'Rourke. Computational geometry in C. <http://cs.smith.edu/~orourke/books/ftp.html>, 2000.
- [O'R03] J. O'Rourke. Comp.graphics.algorithms FAQ. <http://cs.smith.edu/~orourke/FAQ.html>, 2003.
- [osl01] Optimization Solutions and Library, Version 3. International Business Machines, <http://www.ibm.com/software/data/bi/osl>, 2001.
- [Owe03] S. Owen. Meshing Research Corner. <http://www.andrew.cmu.edu/user/sowen/mesh.html>, 2003.
- [PKP⁺02] K. Polthier, S. Khadem, E. Preuss, and U. Reitebuch. JavaView, Version 2.21. <http://www.javaview.de>, 2002.
- [PR03] J. Pfeifle and J. Rambau. Computing triangulations using oriented matroids. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 49–75. Springer-Verlag, Berlin, 2003.
- [Ram03] J. Rambau. TOPCOM, Version 0.13.0. <http://www.zib.de/rambau/TOPCOM>, 2003.
- [RGK] J. Richter-Gebert and U.H. Kortenkamp. Cinderella, Version 1.2. <http://www.cinderella.de>.
- [Sch] R. Schneiders. Software: list of public domain and commercial mesh generators. <http://www-users.informatik.rwth-aachen.de/~roberts/software.html>.
- [Sch82] A. Schönhage. Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients. In *Computer Algebra*, Marseille, pages 3–15. Springer-Verlag, Berlin, 1982.
- [Sch99] P. Schorn. XYZGeobench, Version 5.0.5. <http://www.jn.inf.ethz.ch/geobench>, 1999.
- [She96] J.R. Shewchuk. Triangle, Version 1.3. <http://www.cs.cmu.edu/~quake/triangle.html>, 1996.
- [Ski01] S.S. Skiena. The Stony Brook Algorithm Repository. <http://www.cs.sunysb.edu/~algorith/index.html>, 2001.
- [SLL02] J. Siek, L.-Q. Lee, and A. Lumsdaine. The Boost Graph Library (BGL), Version 1.28.0. <http://www.boost.org/libs/graph/doc/index.html>, 2002.
- [SS71] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)*, 7:281–292, 1971.
- [Ste02] K. Stephenson. CirclePack, Version 6.0. <http://www.math.utk.edu/~kens>, 2002.
- [stu02] Studio, Version 4.1. Raindrop Geomagic, Inc., <http://www.geomagic.com/products/studio>, 2002.

- [Vav00] S.A. Vavasis. QMG, Version 2.0, patch 2. <http://www.cs.cornell.edu/Info/People/vavasis/qmg-home.html>, 2000.
- [W⁺98] S.K. Wismath et al. VisPak, Version 2.0. <http://www.cs.uleth.ca/~wismath/vis.html>, 1998.
- [W⁺02] R. Wunderling et al. The Sequential object-oriented simplex class library, Version 1.2.1. <http://www.zib.de/Optimization/Software/Soplex/soplex.php>, 2002.
- [Wee00] J. Weeks. SnapPea, Version 3.0d3. <http://www.geometrygames.org/SnapPea>, 2000.
- [xpr03] Xpress-MP. Dash Optimization, <http://www.dashoptimization.com>, 2003.
- [YD03] C.K. Yap and Z. Du. Core Library (CORE), Version 1.6. <http://cs.nyu.edu/exact/core>, 2003.
- [yfi03] yFiles, Version 2.1. yWorks GmbH, http://www.yworks.de/en/products_yfiles_about.htm, 2003.
- [Zie95] G.M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, New York, 1995.

65 TWO COMPUTATIONAL GEOMETRY LIBRARIES: LEDA AND CGAL

Lutz Kettner and Stefan Näher

INTRODUCTION

In the last decade, two major software libraries that support a wide range of geometric computing have appeared: LEDA, the **Library of Efficient Data Types and Algorithms**, and CGAL, the **Computational Geometry Algorithms Library**. We start with an introduction of common aspects of both libraries and major differences. We continue with two sections that describe each library in detail.

Both libraries are written in C++. LEDA is based on the object-oriented paradigm and CGAL is based on the generic programming paradigm. They provide a collection of flexible, efficient, and correct software components for computational geometry. Users should be able to easily include existing functionality into their programs. Additionally, both libraries have been designed to serve as platforms for the implementation of new algorithms.

Of course, correctness is of crucial importance for a library, even more so in the case of geometric algorithms where correctness is harder to achieve than in other areas of software construction. Two well-known reasons are the *exact arithmetic assumption* and the *nondegeneracy assumption* that are often used in computational geometry algorithms. However, both assumptions usually do not hold: floating point arithmetic is not exact and inputs are frequently degenerate. See [Chapter 41](#) for details.

EXACT ARITHMETIC

There are basically two scientific approaches to the exact arithmetic problem. One can either design new algorithms that can cope with inexact arithmetic or one can use exact arithmetic. Instead of requiring the arithmetic itself to be exact, one can guarantee correct computations if the so-called *geometric primitives* are exact. So, for instance, the predicate for testing whether three points are collinear must always give the right answer. This allows an efficient implementation of these exact primitives by using floating-point filters or lazy evaluation techniques.

This approach is known as exact geometric computing paradigm and both libraries, LEDA and CGAL, advocate this approach. However, they also offer straight floating point implementations.

DEGENERACY HANDLING

An elegant (theoretical) approach to the degeneracy problem is *symbolic perturbation*. However, this method of forcing input data into general position can cause

some serious problems in practice. In many cases, it increases the complexity of (intermediate) results considerably; and furthermore, the final limit process turns out to be difficult in particular in the presence of combinatorial structures. For this reason, both libraries follow a different approach. They cope with degeneracies directly by treating the degenerate case as the “normal” case. This approach proved to be effective for many geometric problems.

However, symbolic perturbation is used in some places. For example, in CGAL the 3D Delaunay triangulation uses it to realize consistent point insert and removal functions in the degenerate case of more than four points on a sphere [DT03].

LIBRARY STRUCTURE

CGAL and LEDA both support a style of coding which we call *geometric programming*. This is a type of higher level programming that deals with geometric objects and their corresponding primitives rather than working directly on coordinates or numerical representations. In this way the machinery for solving the exact arithmetic problem can be encapsulated in the implementation of the basic geometric operations.

COMMON ROOTS AND DIFFERENCES

LEDA is a general-purpose library of algorithms and data structures, whereas CGAL is focused on geometry. They have a different look and feel and different design principles, but they are compatible with each other and can be used together. A LEDA user can benefit from more geometry algorithms in CGAL, and a CGAL user can benefit from the exact number types and graph algorithms in LEDA, as will be detailed in the individual sections on LEDA and CGAL. There are also joint developments that work with both libraries, e.g., GeoWin for visualization and demos [BN02].

CGAL started six years after LEDA. CGAL learned from the successful decisions and know-how in LEDA (also supported by the fact that LEDA’s founding institute is also a partner in developing CGAL). So CGAL followed LEDA in priority on correctness, geometric programming style, and layout principles for the reference manuals.

The later start allowed CGAL to rely on a better C++ language support, e.g., with templates and traits classes, which led the developers to adopt successfully the new *generic programming paradigm* and shift the design focus more toward flexibility.

A successful spin-off company¹ has been created around LEDA. After an initial free licensing for academic institutions, all LEDA licenses are now fee-based.

A new spin-off company² has been created around CGAL. CGAL also started with a free academic license, but has in contrast now moved with the CGAL release 3.0 to a dual license model with a free open-source license and a commercial license for companies that do not want their developments to become open source.

¹Algorithmic Solutions Software GmbH <www.algorithmic-solutions.com>.

²GeometryFactory Sarl <www.geometryfactory.com>.

GLOSSARY

Exact arithmetic: Foundation layer of the *exact computation paradigm* in computational geometry software that builds correct software layer by layer. Exact arithmetic can be as simple as a built-in integer type as long as its precision is not exceeded or can involve more complex number types, such as, `leda::real` from LEDA [BFMS00] or `Expr` from CORE [KLPY99].

Floating point filter: Technique to speed up exact computations for common easy cases; a fast floating-point interval arithmetic is used unless the error intervals overlap, in which case the computation is repeated with exact arithmetic.

Coordinate representation: Cartesian and homogeneous coordinates are supported by both libraries. Homogeneous coordinates are used to optimize exact rational arithmetic with a common denominator, and not for projective geometry.

Geometric object: Atomic part of a geometric kernel. Examples are points, segments, lines, and circles in the 2D case, and planes, tetrahedra, and spheres in the 3D case. The corresponding data types have value semantics; variants with and without reference-counted representations exist.

Predicate: Geometric primitive returning a value from a finite domain that expresses a geometric property of the arguments (geometric objects), for example, `CGAL::do_intersect(p,q)` returning a Boolean or `leda::orientation(p,q,r)` returning the sign of the area of the triangle (p, q, r) . A *filtered predicate* uses a floating-point filter to speed up computations.

Construction: Geometric primitive constructing a new object, such as the point of intersection of two straight lines.

Geometric kernel: The collection of geometric objects together with the related predicates and constructions. A *filtered kernel* uses filtered predicates.

Program checkers: Technique for writing programs that check their work. A checker for a program computing a function f takes an instance x and an output y . It returns true if $y = f(x)$ and false, otherwise.

65.1 LEDA



LEDA aims at being a comprehensive software platform for the entire area of combinatorial and geometric computing. It provides a sizable collection of data types and algorithms. This collection includes most of the data types and algorithms described in the textbooks of the area ([AHU74, Meh84, Tar83, CLR90, O'R98, Woo93, Sed91, Kin90, van88, NH93]). LEDA supports a broad range of applications. It has already been used in such diverse areas as code optimization, VLSI design, graph drawing, graphics, robot motion planning, traffic scheduling, geographic information systems, machine learning, and computational biology.

The LEDA project was started in the fall of 1988 by Kurt Mehlhorn and Stefan Näher. The first six months was devoted to the specification of different data types and on selecting the implementation language. At that time the item concept

arose as an abstraction of the notion “pointer into a data structure.” Items provide direct and efficient access to data and are similar to iterators in the standard template library. The item concept worked successfully for all test cases and is now used for most data types in LEDA. Concurrently with searching for the correct specifications, several languages were investigated for their suitability as an implementation platform. Among the candidates were Smalltalk, Modula, Ada, Eiffel, and C++. The language had to support abstract data types and type parameters (genericity) and should be widely available. Based on the experiences with different example programs, C++ was selected because of its flexibility, expressive power, and availability.

We next discuss some of the general aspects of the LEDA system.

EASE OF USE

The LEDA library is easy to use. In fact, only a small fraction of the users are algorithms experts and many users are not even computer scientists. For these users the broad scope of the library, its ease of use, and the correctness and efficiency of the algorithms in the library are crucial. The LEDA manual [MNSU] gives precise and readable specifications for the data types and algorithms mentioned above. The specifications are short (typically not more than a page), general (so as to allow several implementations) and abstract (so as to hide all details of the implementation).

EXTENSIBILITY

Combinatorial and geometric computing is a diverse area and hence it is impossible for a library to provide ready-made solutions for all application problems. For this reason it is important that LEDA is easily extensible and can be used as a platform for further software development. In many cases LEDA programs are very close to the typical textbook presentation of the underlying algorithms. The goal is the equation: *Algorithm + LEDA = Program*.

LEDA *extension packages* (LEPs) extend LEDA into particular application domains and areas of algorithmics not covered by the core system. LEDA extension packages satisfy requirements, which guarantee compatibility with the LEDA philosophy. LEPs have a LEDA-style documentation, they are implemented as platform independent as possible, and the installation process permits a close integration into the LEDA core library. Currently, the following LEPs are available: PQ-trees, dynamic graph algorithms, a homogeneous d -dimensional geometry kernel, and a library for graph drawing.

CORRECTNESS

Geometric algorithms are frequently formulated under two unrealistic assumptions: computers are assumed to use exact real arithmetic (in the sense of mathematics) and inputs are assumed to be in general position. The naive use of floating point arithmetic as an approximation to exact real arithmetic very rarely leads to correct implementations. In a sequence of papers [BMS94b, See94, MN94b, BMS94a, FGK⁺00], these degeneracy and precision issues were investigated and LEDA was

extended based on this theoretical work. It now provides exact geometric kernels for 2D and higher-dimensional computational geometry [MMN⁺98], and also correct implementations for basic geometric tasks, e.g., 2D convex hulls, Delaunay diagrams, Voronoi diagrams, point location, line segment intersection, and higher-dimensional convex hulls and Delaunay triangulations.

Programming is a notoriously error-prone task; this is even true when programming is interpreted in a narrow sense: translating a (correct) algorithm into a program. The standard way to guard against coding errors is program testing. The program is exercised on inputs for which the output is known by other means, typically as the output of an alternative program for the same task. Program testing has severe limitations. It is usually only performed during the testing phase of a program. Also, it is difficult to determine the “correct” suite of test inputs. Even if appropriate test inputs are known it is usually difficult to determine the correct outputs for these inputs: alternative programs may have different input and output conventions or may be too inefficient to solve the test cases.

Given that program verification—i.e., formal proof of correctness of an implementation—will not be available on a practical scale for some years to come, *program checking* has been proposed as an extension to testing [BK89, BLR90]. The cited papers explored program checking in the area of algebraic, numerical, and combinatorial computing. In [MNS⁺99, MM95, HMN96] program checkers are presented for planarity testing and a variety of geometric tasks. LEDA uses program checkers for many of its implementations.

AVAILABILITY AND USAGE

LEDA is realized in C++ and can be used on many different platforms with many different compilers. LEDA is now used at more than 1500 academic sites. A commercial version of LEDA is marketed by Algorithmic Solutions Software GmbH.

65.1.1 THE STRUCTURE OF LEDA

LEDA uses templates for the implementation of parameterized data types and for generic algorithms. However, it is not a pure template library and therefore is based on a number of object code libraries of precompiled code. Programs using LEDA data types or algorithms have to include the appropriate LEDA header files into their source code and must link to one or more of these libraries. The four object code libraries are built on top of one another. Here, we only give a brief overview. The LEDA user manual ([MNSU]) or the LEDA book ([MN00]) includes detailed descriptions.

- The Basic Library (*libL*).
Contains system-dependent code, basic data structures, numbers and types for linear algebra, dictionaries, priority queues, partitions, and many more basic data structures and algorithms.
- The Graph Library (*libG*)
Contains different types of graphs and a large collection of graph and network algorithms
- The 2D Geometry Library (*libP*).

Contains the 2D geometric kernels (Section 65.1.2) advanced geometric data structures, and a large number of algorithms for 2D geometric problems (Section 65.1.4).

- The 3D Geometry Library (*libD3*).
Contains the 3D kernels and some algorithms for 3D problems.
- The Window Library(*libW*).
Supports graphical output and user interaction for both the X11 platform (Unix) and Microsoft Windows systems. It also contains animation support, a powerful graph editor (GraphWin), and GeoWin, a interactive tool for the visualization of geometric algorithms. See Section 65.1.5 for details.

65.1.2 GEOMETRY KERNELS

LEDA offers kernels for 2D and 3D geometry, a kernel of arbitrary dimension is available as an extension package. In either case there exists a version of the kernel based on floating point Cartesian coordinates (called *float-kernel*) as well as a kernel based on rational homogeneous coordinates (called *rat-kernel*). All kernels provide a complete collection of geometric objects (points, segments, rays, lines, circles, simplices, polygons, planes, etc.) together with a large set of geometric primitives and predicates (orientation of points, side-of-circle tests, side-of-hyperplane, intersection tests and computation, etc.). For a detailed discussion and the precise specification, see [Chapter 9](#) of the LEDA book ([MN00]). Note that only for the rational kernel, which is based on exact arithmetic and floating-point filters, all operations and primitives are guaranteed to compute the correct result.

65.1.3 DATA STRUCTURES

In addition to the basic kernel data structures LEDA provides many advanced data types for computational geometry. Examples include:

- A general polygon type (`gen_polygon` or `rat_gen_polygon`) with a complete set of Boolean operations. Its implementation is based on an efficient and robust plane sweep algorithms for the construction of the arrangement of a set of straight line segments (see [MN94a] and [MN00, Ch. 10.7]).
- Two- and higher-dimensional geometric tree structures, such as range, segment, interval and priority search trees.
- Partially and fully persistent search trees.
- Different kinds of geometric graphs (triangulations, Voronoi diagrams, and arrangements).
- A dynamic `point_set` data type supporting update, search, closest point, and different types of range query operations on one single representation based on a dynamic Delaunay triangulation (see [MN00, Ch. 10.6]).

65.1.4 ALGORITHMS

The LEDA project never had the goal of providing a complete collection of the algorithms from computational geometry (nor for other areas of algorithms). Rather, it was designed and implemented to establish a *platform* for combinatorial and geometric computing enabling programmers to implement these algorithms themselves more easily and customized to their particular needs. But of course the library already contains a considerable number of basic geometric algorithms. Here we give a brief overview and refer the reader to the user manual for precise specifications and to [Chapter 10](#) of the LEDA-book ([MN00]) for detailed descriptions and analyses of the corresponding implementations. The current version of LEDA offers different implementation of algorithms for the following 2D geometric problems:

- convex hull algorithms (also 3D)
- halfplane intersection
- (constraint) triangulations
- closest and farthest Delaunay and Voronoi diagrams
- Euclidean minimum spanning trees
- closest pairs
- Boolean operations on generalized polygons
- segment intersection and construction of line arrangements
- Minkowski sums and differences
- nearest neighbors and closest points
- minimum enclosing circles and annuli
- curve reconstruction

65.1.5 VISUALIZATION (GeoWin)

In computational geometry, visualization and animation of programs are important for the understanding, presentation, and debugging of algorithms. Furthermore, the animation of geometric algorithms is cited as among the strategic research directions in this area. *GeoWin* [BN02] is a generic tool for the interactive visualization of geometric algorithms. *GeoWin* is implemented as a C++ data type. Its design and implementation was influenced by LEDA's graph editor *GraphWin* ([MN00, Ch. 12]). Both data types support a number of programming styles which have shown to be very useful for the visualization and animation of algorithms. The animations use *smooth transitions* to show the result of geometric algorithms on dynamic user-manipulated input objects, e.g., the Voronoi diagram of a set of moving points or the result of a sweep algorithm that is controlled by dragging the sweep line with the mouse (see [Figure 65.1.1](#)).

A GeoWin maintains one or more geometric scenes. A geometric *scene* is a collection of geometric objects of the same type. A collection is simply either a standard C++ list (STL-list) or a LEDA-list of objects. GeoWin requires that the objects provide a certain functionality, such as stream input and output, basic geometric transformations, drawing and input in a LEDA window. A precise definition

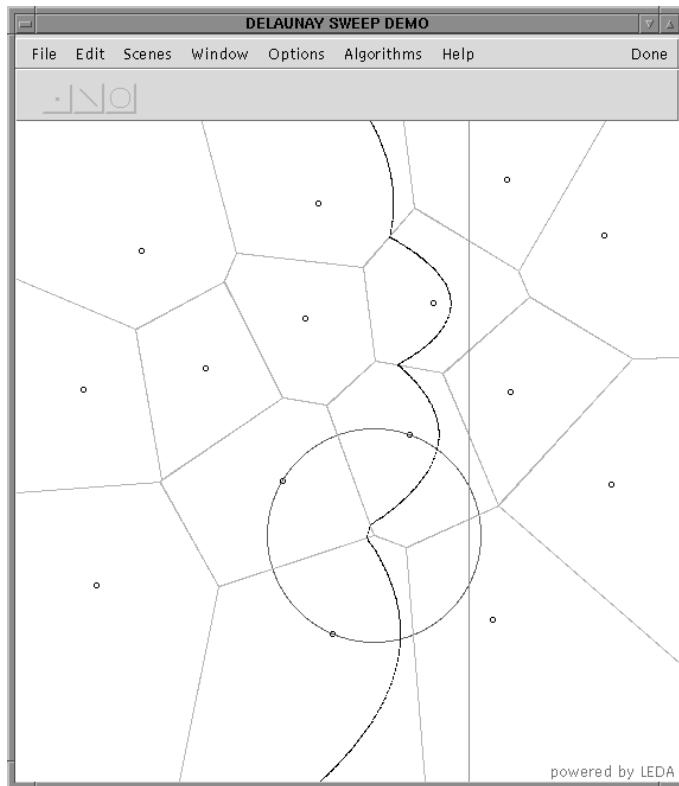


FIGURE 65.1.1
GeoWin animating Fortune's sweep algorithm.

of the required operations can be found in the manual pages [MNSU]. GeoWin can be used for any collection of basic geometric objects (geometry kernel) fulfilling these requirements. Currently, it is used to visualize geometric objects and algorithms from both the CGAL and LEDA libraries.

The visualization of a scene is controlled by a number of attributes, such as color, line width, line style, etc. A scene can be subject to user interaction and it may be defined from other scenes by means of an algorithm (a C++ function). In the latter case the scene (also called *result scene*) may be recomputed whenever one of the scenes on which it depends is modified. There are three main modes for recomputation: user-driven, continuous, and event-driven.

GeoWin has both an interactive and a programming interface. The interactive interface supports the interactive manipulation of input scenes, the change of geometric attributes, and the selection of scenes for visualization.

65.1.6 PROGRAM EXAMPLES

We now give two programming examples showing how LEDA can be used to implement basic geometric algorithms in an elegant and readable way. The first example is the computation of the *upper convex hull* of a point set in the plane. It uses points and the orientation predicate and lists from the basic library. The second example shows how the LEDA *graph* data type is used to represent triangulations in the

implementation of a function that turns an arbitrary triangulation into a Delaunay triangulation by edge flipping. It uses points, lists, graphs, and the side-of-circle predicate.

UPPER CONVEX HULL

In our first example we show how to use LEDA for computing the upper convex hull of a given set of points. We assume that we are in LEDA's namespace, otherwise all LEDA names would have to be used with the prefix `leda::`. Function `UPPER_HULL` takes a list L of rational points (type `rat_point`) as input and returns the list of points of the upper convex hull of L in clockwise ordering from left to right. The algorithm is a variant of Graham's Scan [Gra72].

First we sort L according to the lexicographic ordering of the Cartesian coordinates and remove multiple points. If the list contains not more than two points after this step we stop. Before starting the actual Graham Scan we first skip all initial points lying on or below the line connecting the two extreme points. Then we scan the remaining points from left to right and maintain the upper hull of all points seen so far in a list called *hull*. Note however that the last point of the hull is not stored in this list but in a separate variable p . This makes it easier to access the last two hull points as required by the algorithm. Note also that we use the rightmost point as a sentinel avoiding the special case that *hull* becomes empty.

```
list<rat_point> UPPER_HULL(list<rat_point> L) {
    L.sort();
    L.unique();

    if (L.length() <= 2) return L;

    rat_point p_min = L.front(); // leftmost point
    rat_point p_max = L.back(); // rightmost point

    list<rat_point> hull;           // result list
    hull.append(p_max);            // use rightmost point as sentinel
    hull.append(p_min);            // first hull point

    // goto first point p above (p_min,p_max)
    while (! L.empty() && ! left_turn(p_min, p_max, L.front())) L.pop();
    if (L.empty()) {               // upper hull consists of only 2 points
        hull.reverse();
        return hull;
    }

    rat_point p = L.pop();         // second (potential) hull point
    rat_point q;
    forall(q,L) {
        while (! right_turn(hull.back(), p, q)) p = hull.pop_back();
        hull.append(p);
        p = q;
    }
    hull.append(p);                // add last hull point
    hull.pop();                   // remove sentinel
    return hull;
}
```

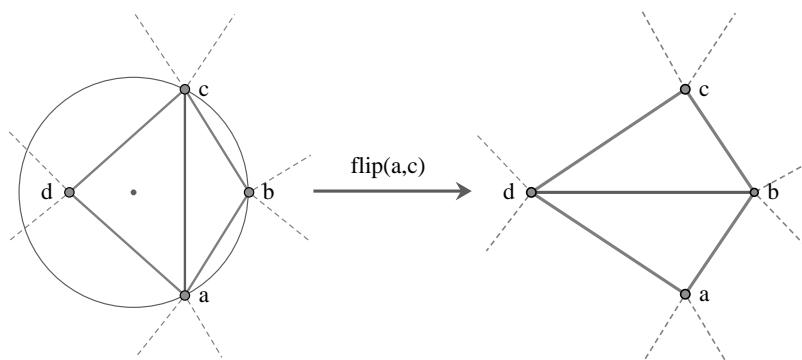
DELAUNAY FLIPPING

LEDA represents triangulations by bidirected plane graphs (from the graph library) whose nodes are labeled with points and whose edges may carry additional information, e.g., integer flags indicating the type of edge (hull edge, triangulation edge, etc.). All edges incident to a node v are ordered in counterclockwise ordering and every edge has a reversal edge. In this way the faces of the graph represent the triangles of the triangulation. The graph type offers methods for iterating over the nodes, edges, and adjacency lists of the graph. In the case of plane graphs there are also operations for retrieving the reverse edge and for iterating over the edges of a face. Furthermore, edges can be moved to new nodes. This graph operation is used in the following program to implement edge flips.

Function `DELAUNAY_FLIPPING` takes as input an arbitrary triangulation and turns into a Delaunay triangulation by the well-known flipping algorithm. This algorithm performs a sequence of local transformations as shown in Figure 65.1.2 to establish the Delaunay property: for every triangle the circumscribing circle does not contain any vertex of the triangulation in its interior. The test whether an edge has to be flipped or not can be realized by a so-called *side_of_circle* test. This test takes four points a, b, c, d and decides on which side of the oriented circle through the first three points a, b , and c the last point d lies. The result is positive or negative if d lies on the left or on the right side of the circle, respectively, and the result is zero if all four points lie on one common circle. The algorithm uses a list of candidates which might have to be flipped (initially all edges). After a flip the four edges of the corresponding quadrilateral are pushed onto this candidate list. Note that $G[v]$ returns the position of node v in the triangulation graph G . A detailed description of the algorithm and its implementation can be found in the LEDA book ([MN00]).

FIGURE 65.1.2

Flipping to establish the Delaunay property.



```

void DELAUNAY_FLIPPING(GRAPH<rat_point, int>& G) {
    list<edge> S = G.all_edges();
    while (! S.empty()) {
        edge e = S.pop();
        edge r = G.rev_edge(e);

        edge e1 = G.face_cycle_succ(r); // e1,e2,e3,e4: edges of quadrilateral
        edge e2 = G.face_cycle_succ(e1); // with diagonal e
        edge e3 = G.face_cycle_succ(e);
        edge e4 = G.face_cycle_succ(e3);

        rat_point a = G[G.source(e1)]; // a,b,c,d: corners of quadrilateral
        rat_point b = G[G.target(e1)];
        rat_point c = G[G.source(e3)];
        rat_point d = G[G.target(e3)];

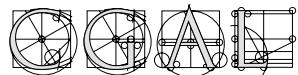
        if (side_of_circle(a,b,c,d) > 0) {
            S.push(e1); S.push(e2); S.push(e3); S.push(e4);
            G.move_edge(e,e2,source(e4)); // flip diagonal
            G.move_edge(r,e4,source(e2));
        }
    }
}

```

65.1.7 PROJECTS ENABLED BY LEDA

A large number of academic and industrial projects from almost every area of combinatorial and geometric computing have been enabled by LEDA. Examples are graph drawing, algorithm visualization, geographic information systems, location problems, visibility algorithms, DNA sequencing, dynamic graph algorithms, map labeling, covering problems, railway optimization, route planning and many more. The page <<http://www.mpi-sb.mpg.de/LEDA/friends>> lists academic projects in detail, and <<http://www.algorithmic-solutions.com/enreferenzen.htm>> describes selected industrial projects based on LEDA.

65.2 CGAL



The development of CGAL, the Computational Geometry Algorithms Library, began in 1995 and the first public release 0.9 appeared in June 1997. The presentation here is based on the CGAL release 3.0 from October 2003, available from CGAL's home page <www.cgal.org>.

CGAL is developed by a consortium consisting of ETH Zürich (Switzerland), Freie Universität Berlin (Germany), INRIA Sophia-Antipolis (France), Martin-Luther-Universität Halle-Wittenberg (Germany), Max-Planck Institut für Informatik, Saarbrücken (Germany), RISC Linz (Austria), Tel-Aviv University (Israel), and Utrecht University (The Netherlands). This work was the central task of two successive ESPRIT IV LTR projects named CGAL and GALIA. It is the goal of these

projects to

make the large body of geometric algorithms developed in the field of computational geometry available for industrial application.

CGAL's main design goals are correctness, flexibility, efficiency, and ease of use. Its focus is on a broad foundation in computational geometry. Important related issues, for example visualization, are supported with standard formats and interfaces.

The design of CGAL and our decision to use the C++ language are thoroughly covered in [FGK⁺00]. Generic programming aspects are discussed in [BKS⁰⁰]. New developments in the CGAL kernel are presented in [HHK⁺01], the d -dimensional kernel in [MMN⁺98]. Older descriptions of design and motivation are in [Ove96, FGK⁺96, Vel97]. In particular, precision and robustness aspects are discussed in [Sch96], and the influence of different kernels in [Sch99, BBP01].

LIBRARY STRUCTURE

CGAL is structured in layers: The *core library* with nongeometric support functions and types, the *geometric kernel* for constant-size geometric objects, predicates and constructions, the *basic library* with data structures and algorithms, and the *support library* with number types, geometric object generators, file I/O, visualization, and more nongeometric functions and types. CGAL follows the generic programming paradigm in the spirit of the STL (Standard Template Library) of the C++ Standard. As a consequence, the different parts of CGAL are highly modular and independent of each other.

GENERIC PROGRAMMING IDIOMS

Concept: Set of requirements for a C++ template parameter.

Model for a concept: A type in C++ that fulfills all requirements of that concept and can therefore be used as template argument in places where the concept was requested.

Function object: Implements a function as a C++ class with an `operator()`. It is more efficient and type-safe compared to a C function pointer or object-oriented class hierarchies.

FLEXIBILITY

CGAL has a *modular* design of layers and packages that is transparent in the documentation, although currently only the whole library can be installed. The algorithms and data structure in CGAL are *adaptable* to already existing user code; see the geometric traits class example on page 1453. The library is *extendible*, users can add implementations in the same style as CGAL. The library is *open* and supports important standards, such as the C++ standard with its Standard Template Library STL, or important other libraries, such as LEDA or GMP, the Gnu Multiple Precision Arithmetic Library [Gra02].

CORRECTNESS

CGAL addresses the robustness problems in geometric computing, as formulated in the introduction, by relying on exact predicate evaluation, e.g., with exact arithmetic, and explicit degeneracy handling.

In addition, we have a well-established software process and communication set up for our distributed developers community. We use CVS for revision management and run an automatic test-suite twice a week on all supported platforms and compilers. An editorial board reviews new submissions and supervises design homogeneity. We also had one-round of peer-reviewing and testing for the basic library packages.

EASE OF USE

Users with a base knowledge of C++ and the STL will experience a smooth learning curve with CGAL since many concepts are already known from the STL, and the powerful flexibility is often hidden behind sensible defaults. A novice reader should not be discouraged by some of the advanced examples illustrating CGAL's power.

CGAL has a uniform design, aims for complete and minimal interfaces, yet rich and complete functionality in the area of computational geometry. The extensive manual follows the layout style of the LEDA manuals.

EFFICIENCY

CGAL follows the generic programming paradigm and uses templates in C++ to realize most of its flexibility. Thus, the flexibility is resolved at compile time and does not have any runtime cost. That allows us to realize flexibility at places normally not considered because of runtime costs, e.g., on the number-type level.

Furthermore, the flexibility allows to pick the best of the available choices for a particular application. Tradeoffs between space and time in some data structures, or between different number types of different power and speed can be made at the application level, not in the library. This also encourages experimental research.

65.2.1 GEOMETRIC KERNEL

CGAL offers a wide variety of kernels. The geometric objects, predicates, and constructions—classified according to dimension two, three, and arbitrary d —are summarized in [Table 65.2.1](#). The kernels available in CGAL can be classified along the following orthogonal concepts:

Dimension: The dimension of the affine space. The specializations for dimension two and three offer functionality that would not be available in the arbitrary-dimension kernel.

Number type: CGAL kernels use one number type uniformly to store coordinates and coefficients, and to compute the expressions in predicates and constructions.

CGAL distinguishes four concepts of number types: a *ring* for exact integer arithmetic, an *Euclidean ring* that adds integer division and a *gcd* (greatest

TABLE 65.2.1 Kernel objects, selected predicates and constructions.

DIM	GEOMETRIC OBJECTS	PREDICATES	CONSTRUCTIONS
<i>all</i>	Point, Vector, Direction, Line, Ray, Segment, Aff_transformation	compare_lexicographically, do_intersect, orientation	intersection, midpoint transform, squared_distance
2	Triangle, Iso_rectangle, Bbox, Circle	collinear, left_turn, side_of_oriented_circle	bbox, centroid, circumcenter, squared_radius, rational_rotation_approximation
3	Plane, Tetrahedron, Triangle, Iso_cuboid, Bbox, Sphere	coplanar, left_turn, side_of_oriented_sphere	bbox, centroid, circumcenter, cross_product, squared_radius
<i>d</i>	Hyperplane, Sphere	side_of_oriented_sphere	center_of_sphere, lift_to_paraboloid

common divisor) computation, a *field* with exact division, and a number type that supports the exact sign evaluation for expressions with roots.

Exceptions to the one-number-type principle arise in the homogeneous kernel, where a field type (the quotient type `CGAL::Quotient<Field_type>` by default) is associated with the ring type, and in predicates that are specialized on particular number types. The specialized predicates might use a different arithmetic to evaluate an expression exactly although the number type itself would be too limited. The specialized predicates might also use floating point filters.

Coordinate representation: The Cartesian representation requires a *field* as a number type. The homogeneous representation requires a *Euclidean ring* as a number type. The homogeneous coordinate is used to optimize exact rational arithmetic with a common denominator, and not for projective geometry. CGAL implements strictly affine geometry.

Reference counting: Reference counting is used to optimize copying and assignment of kernel objects. It is recommended for exact number types with larger memory size. The kernel objects have value-semantics and cannot be modified, which simplifies reference counting. However, a copy-on-write strategy is available for modifiable objects elsewhere in CGAL. The nonreference counted kernels are recommended for small and fast number types, such as the built-in `double`.

Let `RT` be a *Euclidean ring*, and `FT` a *field* number type. The kernels in CGAL are:

<code>CGAL::Cartesian<FT></code>	Cartesian, reference counted, 2D and 3D
<code>CGAL::Simple_cartesian<FT></code>	Cartesian, nonreference counted, 2D and 3D
<code>CGAL::Homogeneous<RT></code>	homogeneous, reference counted, 2D and 3D
<code>CGAL::Simple_homogeneous<RT></code>	homogeneous, nonreference counted, 2D and 3D
<code>CGAL::Cartesian_d<FT></code>	Cartesian, reference counted, <i>d</i> -dimensional
<code>CGAL::Homogeneous_d<RT></code>	homogeneous, reference counted, <i>d</i> -dimensional

The geometric objects are local types of a kernel, e.g., `Cartesian<leda::real>::Point_2` is a 2D point with Cartesian coordinates of type `leda::real`. The predicates and constructions are local function objects of a kernel. However, global functions provide a more conventional way of calling predicates and constructions.

CGAL offers three possibilities to create filtered kernels: (1) `CGAL::Lazy_exact_nt<NT>` is a filtered number type using `double` interval arithmetic and the exact number type `NT` given as template argument. It creates an expression DAG, evaluates it with interval arithmetic and, if that fails, switches to the exact number type for evaluation. This allows exact constructions. Some predicates are specialized to avoid the expression DAG construction. (2) `CGAL::Filtered_exact<CT, ET>` is a number type using the type `CT` for representation and constructions. Predicates are specialized for this number type to use interval arithmetic as filter and, if that fails, to use the exact number type `ET`. (3) `CGAL::Filtered_kernel<K>` constructs a new kernel based on the given kernel `K`. All predicates of the new kernel use the interval arithmetic as filter and, if that fails, call the predicates in the kernel `K` [BBP01].

A common misconception should be clarified here: A filter in CGAL can only be applied to predicates, not to constructions. If exact constructions are required one needs an exact number type in the kernel. For example, the Delaunay triangulation can be computed correctly with a filtered kernel, but the center of a circumcircle cannot. A kernel such as `CGAL::Homogeneous<CGAL::Gmpz>` would allow the exact construction. It should be mentioned that an approach as in the LOOK kernel [FM02] does extend filtering to constructions, but is not available in CGAL to date.

DEFAULT CHOICES FOR THE GEOMETRIC KERNEL

To ease the choice of a kernel and a suitable number type for beginners, CGAL offers three default kernels for dimension two and three that cover the most common cases of tradeoffs between speed and exactness requirements. These default choices allow initial exact constructions from `double` values and guarantee exact predicates. They vary in their capability of exact constructions and use of exact square root expressions in predicates. The names speak for themselves.

- `CGAL::Exact_predicates_inexact_constructions_kernel`
- `CGAL::Exact_predicates_exact_constructions_kernel`
- `CGAL::Exact_predicates_exact_constructions_kernel_with_sqrt`

EXAMPLE: ORIENTATION OF TRIPLE

The following example creates three points in the plane and computes their orientation. We use the `CGAL::MP_Float` number type that can represent floating-point values of arbitrary precision with the homogeneous kernel. We obtain an exact kernel that could also work correctly with constructions and with `double` input values.

```
#include <CGAL/MP_Float.h>
#include <CGAL/Homogeneous.h>

typedef CGAL::Homogeneous< CGAL::MP_Float> Kernel;
typedef Kernel::Point_2 Point_2;
```

```

int main() {
    Point_2 p(0,0), q(10,3), r(12,19);
    if (CGAL::orientation(p,q,r) == CGAL::LEFT_TURN)
        return 0;
    return 1;
}

```

The above kernel is exact, but needs more space and time than a simple `double` implementation. We want to optimize space and time in the following example using `double` coordinates in a Cartesian representation without reference counting. However, we do not want to sacrifice correctness and use the filtered kernel that has specialized predicate implementations; but note that constructions with `doubles` will be prone to rounding errors.

In the current CGAL release, we cannot use the global functions with the filtered kernels. Instead, we ask the kernel for a predicate function object, which works for all kernels.³

```

#include <CGAL/Simple_cartesian.h>
#include <CGAL/Filtered_kernel.h>

typedef CGAL::Simple_cartesian<double> Kernel;
typedef CGAL::Filtered_kernel<Kernel> Filtered_kernel;
typedef Filtered_kernel::Point_2 Point_2;
typedef Filtered_kernel::Orientation_2 Orientation;

int main()
{
    Point_2 p(0,0), q(10,3), r(12,19);
    Filtered_kernel kernel;
    Orientation orientation = kernel.orientation_2_object();
    if (orientation(p,q,r) == CGAL::LEFT_TURN)
        return 0;
    return 1;
}

```

Again, the above filtered kernel does not support exact constructions, and all the kernels used in these examples are limited to the operations on *field types*. A most flexible but also much slower alternative is the `CGAL::Cartesian<leda::real>` kernel that supports exact constructions of arbitrary depth including expressions with *k*th-roots.

65.2.2 BASIC LIBRARY

The basic library follows the design of the STL, the C++ Standard Template Library [Aus98]; generic algorithms are parameterized with iterator ranges that decouple them from data structures. In addition, CGAL invented the *circulator* concept to accommodate circular structures efficiently, such as the ring of edges around a vertex in planar subdivisions [FGK⁺00]. Essential for CGAL's flexibility is the separation of algorithms and data structures from the underlying geometric kernel with a *geometric traits class*.

³We could have written `Filtered_kernel().orientation_2_object()(p,q,r)` to have the predicate call in one line, but it is less readable to parse the C++ code this way.

GLOSSARY

Iterator: A concept for an abstraction of pointer into a linear sequence. Exists in different flavors: input, output, forward, bidirectional, and random-access iterator.

Circulator: A concept similar to iterator but for circular sequences.

Range: A pair of iterators (or circulators) describing a (sub-)sequence of items in a half-open interval notation, i.e., starting with the first item and ending before the second item.

Traits class: C++ programming technique to attach additional information to a type or function, e.g., dependent types, functions, and values.

Geometric traits: Traits classes used in CGAL to decouple the basic library from a geometric kernel. Algorithms and data structure define a geometric traits concept and the library provides various models that can be used. Often the geometric kernel itself is a valid model.

EXAMPLE OF UPPER CONVEX HULL ALGORITHM

We implement the upper convex hull algorithm following Andrew's variant of Graham's scan [Gra72, And79] with CGAL. First, we translate the implementation used in the LEDA example on page 1443 literally to STL and CGAL code for easy comparison. Therefore we use a sufficient default kernel and declare it globally. Both implementations look similar.

```
typedef CGAL::Exact_predicates_inexact_constructions_kernel Kernel;
typedef Kernel::Point_2 Point_2;
Kernel kernel; // our instantiated kernel object

std::list<Point_2> upper_hull( std::list<Point_2> L ) {
    L.sort( kernel.less_xy_2_object() );
    L.unique();
    if (L.size() <= 2)
        return L;
    Point_2 p_min = L.front(); // leftmost point
    Point_2 p_max = L.back(); // rightmost point
    std::list< Point_2> hull;
    hull.push_back(p_max); // use rightmost point as sentinel
    hull.push_back(p_min); // first hull point
    while (!L.empty() && !kernel.left_turn_2_object()(p_min,p_max,L.front()))
        L.pop_front(); // goto first point p above (p_min,p_max)
    if (L.empty())
        hull.reverse(); // fix orientation for this special case
    return hull;
}
Point_2 p = L.front(); // keep last point on current hull separately
L.pop_front();
for (std::list< Point_2>::iterator i = L.begin(); i != L.end(); ++i) {
    while (! kernel.left_turn_2_object()( hull.back(), *i, p )) {
```

```

        p = hull.back();    // remove non-extreme points from current hull
        hull.pop_back();
    }
    hull.push_back(p);    // add new extreme point to current hull
    p = *i;
}
hull.push_back(p);        // add last hull point
hull.pop_front();         // remove sentinel
return hull;
}

```

Now we rewrite the example to expose more of CGAL's flexibility and also of its own way of implementing generic code inside the library itself, which follows the conventional style of STL code with iterators and generic algorithms.

As a first obvious solution we can make the kernel exchangeable as a template parameter of the function. Before we do so, we factor out the core of the control flow—the two nested loops at the end—into its own generic function with an interface of bidirectional iterators and a single three-parameter predicate. We can eliminate the additional list data structure for the `hull` when we reuse the space that becomes available in the original sequence as our stack. So the result is returned in our original sequence starting with the iterator `first` and running to the past-the-end position that we return in the return value of the function. The interface abstraction with iterators hinders us in realizing the sentinel easily. But since the runtime difference was not measurable we go back to an explicit test for the boundary case, which accounts for the additional `break` statement, but also simplifies code later.

```

template <class Iterator, class Fct> // bidirectional iterator, function object
Iterator backtrack_remove_if_triple( Iterator first, Iterator beyond, Fct pred){ 
    if (first == beyond)
        return first;
    Iterator i = first, j = first;
    if (++j == beyond)           // i,j mark two elements on the top of the stack
        return j;
    Iterator k = j;              // k marks the next candidate value in the sequence
    while (++k != beyond) {
        while (pred( *i, *j, *k)) {
            j = i;                // remove one element from stack, part 1
            if (i == first)       // explicit test for stack underflow
                break;
            --i;                  // remove one element from stack, part 2
        }
        i = j; ++j; *j = *k; // push next candidate value from k on stack
    }
    return ++j;
}

```

Having this generic function, we can implement a variant of the upper hull algorithm that returns all points on the upper convex hull (instead of only the extreme points) in two lines. All degeneracies are handled correctly in the generic functions. This implementation requires a range of random access iterators because of the sorting. It also uses now a template parameter for a geometric traits class and extracts the suitable predicate from this traits class for the call to the new generic function. A

kernel from CGAL is a valid model for this traits parameter. Here, CGAL's design makes things fit together smoothly.

```
template <class Iterator, class Traits> // random access iterator
Iterator upper_hull( Iterator first, Iterator beyond, Traits traits) {
    std::sort( first, beyond, traits.less_xy_2_object());
    return backtrack_remove_if_triple( first, beyond,
        traits.left_turn_2_object());
}
```

We apply the upper hull function to an array filled with three points. Because the resulting hull contains only two points, the `result` value is equal to `points + 2` and the array is modified to start with these two hull points.

```
Point_2 points[3] = { Point_2(0,0), Point_2(1,-1), Point_2(2,0) };
Point_2 *result = upper_hull( points, points + 3, kernel);
```

We go back to compute the extreme points and reuse the new generic function. The code simplifies because we do not use the sentinel technique anymore. However, we need a different orientation test than the `left_turn` predicate provided by CGAL. The other orientation predicates are omitted in CGAL since they can be realized easily with the `left_turn` predicate, permuted parameters, and negating the result. We can use higher order function objects from CGAL to achieve exactly that on a generic function object level; `swap_2` exchanges the order of the second and the third parameter of the `left_turn` predicate and `negate` is inverting the return value, and none of this is costing extra runtime.

We stay with the random-access iterator-based interface, but a list-based interface would be an obvious combination with the first implementation above.

```
template <class Iterator, class Traits> // random access iterator
Iterator upper_hull( Iterator first, Iterator beyond, Traits traits) {
    std::sort( first, beyond, traits.less_xy_2_object());
    beyond = std::unique( first, beyond, traits.equal_2_object());
    return backtrack_remove_if_triple( first, beyond,
        CGAL::negate(
            CGAL::swap_2( traits.left_turn_2_object())));
}
```

EXAMPLE OF USER KERNEL

In contrast to LEDA, data structures and algorithms in CGAL can be easily adapted to work on user data with a custom geometric traits class. Let us assume we already have a point class:

```
struct Point { // our point type
    double x, y;
    Point( double xx = 0.0, double yy = 0.0) : x(xx), y(yy) {}
    bool operator==( const Point& p) const { return x == p.x && y == p.y; }
    bool operator!=( const Point& p) const { return ! (*this == p); }
};
```

We want to use this point class with one of CGAL's convex hull algorithms. The reference manual tells us for the `CGAL::ch_graham_andrew` function that we need

a type `Point_2`, and function objects `Equal_2`, `Less_xy_2`, and `Left_turn_2`. A possible geometric traits class could look like this:

```
struct Geometric_traits { // traits class for our point type
    typedef double RT;           // ring number type, for random points generator
    typedef Point Point_2;      // our point type
    struct Equal_2 {           // equality comparison
        bool operator()( const Point& p, const Point& q) {
            return (p.x == q.x) && (p.y == q.y);
        }
    };
    struct Less_xy_2 {          // lexicographic order
        bool operator()( const Point& p, const Point& q) {
            return (p.x < q.x) || ((p.x == q.x) && (p.y < q.y));
        }
    };
    struct Left_turn_2 {         // orientation test
        bool operator()( const Point& p, const Point& q, const Point& r) {
            return (q.x-p.x) * (r.y-p.y) > (q.y-p.y) * (r.x-p.x); // inexact!
        }
    };
    // member functions to access function objects, here by default construction
    Equal_2 equal_2_object() const { return Equal_2(); }
    Less_xy_2 less_xy_2_object() const { return Less_xy_2(); }
    Left_turn_2 left_turn_2_object() const { return Left_turn_2(); }
};
```

In the last step we have to let CGAL know that our traits class belongs to our point class. We specialize CGAL's kernel traits for this:

```
namespace CGAL { // specialization that links our point type with our traits class
    template <> struct Kernel_traits< ::Point> {
        typedef ::Geometric_traits Kernel;
    };
}
```

Now, we can use the `CGAL::ch_graham_andrew` function on our points. The above implementation also suffices to employ the random point generators in CGAL. Here is a complete program computing the convex hull of 20 points from a random distribution in a disk.

```
#include <CGAL/ch_graham_andrew.h>
#include <CGAL/point_generators_2.h>
#include <CGAL/copy_n.h>
#include <vector>

int main() {
    std::vector<Point> points, hull;
    CGAL::Random_points_in_disc_2<Point> rnd_pts( 1.0 );
    CGAL::copy_n( rnd_pts, 20, std::back_inserter( points ) );
    CGAL::ch_graham_andrew( points.begin(), points.end(),
                           std::back_inserter( hull ), Geometric_traits() );
    return 0;
}
```

The separation of the algorithms and data structures from the geometric kernel provides flexibility, the fingerprint of generic programming, resolved at compile time and therefore without sacrificing performance. We foster such flexibility in CGAL and design algorithms and data structures to be used in different contexts. One example is the geometric traits class `CGAL::Triangulation_euclidean_traits_xy_3` that allows to build a 2D triangulation of the projections on the xy -plane of 3D points, useful for terrain triangulations in GIS (cf. Chapter 58).

BASIC LIBRARY CONTENTS

The basic library contains data structures and algorithms. It is structured into packages that correspond to different chapters of the reference manual.

CONVEX HULL

The *2D convex hull* algorithms return the counterclockwise sequence of the extreme points. The *3D convex hull* algorithms return, in nondegenerate cases, the convex polytope of the extreme points. The *d-dimensional* convex hull algorithm returns a simplicial complex for the closure of the convex polytope. All implementations are iterator-based generic algorithms and data structures. See Table 65.2.2.

TABLE 65.2.2 Convex hull algorithms on a set of n points with h extreme points.

DIM	MODE	ALGORITHM
2	Static	Bykat, Eddy, and Jarvis march, all in $O(nh)$ time
	Static	Akl & Toussaint, and Graham-Andrew scan, both in $O(n \log n)$ time [Sch99]
	Polygon	Melkman for points of a simple polygon in $O(n)$ time
	Others	lower hull, upper hull, subsequences of the hull, extreme points, convexity test
3	Static	quickhull [BDH96]
	Incremental	randomized incremental construction [CMS93, BMS94b]
	Dynamic	by-product of the dynamic Delaunay tetrahedronization in 3D
	Test	convexity test as program checker [MNS ⁺ 99]
d	Incremental	randomized incremental constr. [CMS93, BMS94b], also as LEDA extension package

POLYGON AND NEF POLYGON

A *polygon* is a closed chain of edges. CGAL provides a container class for polygons, but all functions are generic with iterators and work on arbitrary sequences of points. The functions available are polygon area, point location, tests for simplicity and convexity of the polygon, and generation of random instances.

Polygons can also be *partitioned* into y -monotone polygons or convex polygons. The y -monotone partitioning is based on the sweep-line algorithm explained

in [dBvK⁺00]. For the convex partitioning an optimal algorithm w.r.t. number of pieces and a factor 4-approximation sweep-line algorithm are given [Gre83]. Another factor 4-approximation algorithm is based on the constrained Delaunay triangulation [HM83].

A *Nef polygon* is a point set $P \subseteq \mathbb{R}^2$ generated from a finite number of open halfspaces by set complement and set intersection operations. It is therefore closed under Boolean set operations and topological operations; intersection, union, complement, difference, closure, interior, boundary, regularization, etc. It also captures features of mixed dimension, e.g., antennas or isolated vertices, open and closed boundaries, and unbounded faces, lines, and rays. The theory of Nef polyhedra is explained in [Nef78, Bie95], and a full implementation report is available in [See01]. The potential unboundedness of Nef polygons is addressed with *infimaximal frames* and an extended kernel [MS01]. The representation is based on the *halfedge data structure* [Ket98] (see below), extended with face loops and isolated vertices.

PLANAR MAPS, SWEEP-LINE ALGORITHM, ARRANGEMENTS, AND POLYHEDRAL SURFACES

Planar maps are based on the halfedge data structure, an edge-based representation with two oppositely directed halfedges per edge [Ket98]. Planar maps extend the halfedge data structure with a geometric embedding in the plane and halfedge cycles for inner and outer loops around faces. Various basic manipulations of the map are available [FHH⁺00]. The point-location and vertical ray-shooting use an incremental randomized algorithm for a dynamic trapezoidal decomposition to achieve $O(\log n)$ expected location and update time [Mul90, Sei91].

Planar maps are generic with respect to the type of curve they allow for the embedding of the edges. Currently, segments, poly-lines, circular arcs, and general conic arcs are supported [Wei02]. It uses new techniques for handling degeneracies as the existing exact algebraic number types do not yet support all the operations required by intersecting conics, and it uses filtering techniques at the geometry level, and not only at the number type level.

Planar map curves must be non-intersecting and x -monotone. The *planar map with intersections* supports also intersecting curves. A sweep-line algorithm speeds up the construction compared to the incremental insertion. The *arrangements* extend the planar maps with intersections. They maintain the relationship between input curves and x -monotone subcurves in a flexible multi-layer curve hierarchy [HH00].

A *polyhedral surface* is a mesh data structure based on the halfedge data structure. It embeds the halfedge data structure in 3D space. The polyhedral surface provides various basic integrity-preserving operations, the “Euler operations” [Ket98].

TRIANGULATIONS, VORONOI DIAGRAMS, AND ALPHA SHAPES

The triangulations use a triangle-based data structure in 2D, and a tetrahedra-based data structure in 3D. Both are standard container classes with an iterator interface. The triangulations are built with a randomized incremental construction

and support efficient vertex removal [BDP⁺02, DT03]. The Voronoi diagram is only implicitly represented with its dual triangulation.

Point location is the walk method by default. The Delaunay hierarchy [Dev02] is available in 2D and 3D to speed up point location. It is recommended for triangulations of more than 10000 points [DPT02].

Regular triangulations are the dual of power diagrams, the Voronoi diagram of weighted points under the power-distance. Regular triangulations are available in 2D and 3D [ES96].

Apollonius graphs are the dual of the *Apollonius diagrams* that are also known as *additively weighted* Voronoi diagrams of weighted points. They are available in 2D with dynamic vertex insertion, deletion, and fast point location [KY02].

Alpha shapes are extensions of the triangulations. The simplicial subcomplex for the alpha shape of the Delaunay (or regular) triangulation can be efficiently selected for a given α parameter value. Alpha shapes are available in 2D and 3D, for unweighted and for weighted points under the power distance [EM94].

A constrained triangulation (cf. Chapter 25) accepts as input in addition to the points a set of constraining segments. These segments are required edges in the triangulation. Intersecting segments can be handled in various ways. A constrained triangulation and a constrained Delaunay triangulation are available in 2D only.

A d -dimensional Delaunay triangulation is available with the $(d+1)$ -dimensional convex hull algorithm and an adapter that implements the lifting map [BMS94b].

OPTIMIZATION

The geometric optimization algorithms in CGAL fall into three categories, *Bounding Volumes*, *Optimal Distances*, and *Advanced Techniques*; see Table 65.2.3.

TABLE 65.2.3 Geometric optimization.

DIM	ALGORITHM
2,3, d	Smallest enclosing disk/sphere of a point set [Wel91, GS98a, Gär99]
2,3, d	Smallest enclosing sphere of a set of spheres [FG03]
2	Smallest enclosing ellipse of a point set [Wel91, GS97, GS98b]
2	Smallest enclosing rectangle [Tou83], parallelogram [STV ⁺ 95], and strip [Tou83] of a point set
d	Smallest enclosing annulus [GS00]
2	Maximum (area and perimeter) inscribed k -gon of a convex polygon [AKM ⁺ 87]
2	Rectangular p -center, $2 \leq p \leq 4$ [Hof99, SW96]
d	Distance between the convex hulls of two given point sets [GS00]
3	Width of a point set
2	All furthest neighbors for the vertices of a convex polygon [AKM ⁺ 87]
d	Monotone [AKM ⁺ 87] and sorted [FJ84] matrix search

SEARCH STRUCTURES

CGAL provides generic *range trees* and *segment trees* [dBvK⁺00] that can be interchangeably nested to form higher-dimensional search trees. In addition, CGAL provides d -dimensional k -d-trees; however, they cannot be mixed with the other trees. All trees are static and provide the conventional window and enclosing queries.

More advanced queries are k -nearest and k -furthest neighbor searching, incremental nearest and incremental furthest neighbor searching [HS95]. All queries are available as exact and approximate searches. Query items can be points and other spatial objects. These queries are based on the d -dimensional k -d-trees.

Related to the segment tree is the interval skip list in CGAL, a data structure for finding all intervals that overlap a point, that is fully dynamic but works only in the one-dimensional case [Han91].

Furthermore, an interface class to the dynamic 2D Delaunay triangulation implements nearest neighbor, k -nearest neighbors, and range searching in the plane following the idea described in [MN00].

65.2.3 PROJECTS ENABLED BY CGAL

CGAL extension packages are external contributions on top of CGAL available from <<http://www.cgal.org/CEP>>. They allow more flexibility in licensing, documentation, or support questions. Currently CGAL offers one extension package for the Gale-transform of a set of points, the visibility complex data structure for planar scenes, and an adapter for the LEDA rational kernel. Others for parametric search, polygonal approximations, and shape matching are in progress.

CGAL was also successful in enabling new academic projects. The 3D Delaunay triangulation was used in surface reconstructions [DG01, AGJ00, GJ02] and in meshing [CSdVY02]. Planar maps and arrangements were used in exact Minkowski sums and motion planning [AFH02, HH02, Hal02]. They were also used to understand and experiment with an algorithmic idea on the union of geometric objects; for example, an arrangement was built from triangles with half-a-million vertices [EHS02]. The polyhedral surface was used in approximate swept volumes [Raa99, Hal02] and computing a canonical polygonal schema of an orientable triangulated surface [LPVV01]. The halfedge data structure was used in modeling pseudotriangulations [KKM⁺03].

Finally, the programming paradigm enabled a successful and practical implementation framework for parametric search [vOV02].

RELATED CHAPTERS

[Chapter 41: Robust geometric computation](#)

[Chapter 64: Software](#)

REFERENCES

- [AFH02] P.K. Agarwal, E. Flato, and D. Halperin. Polygon decomposition for efficient construction of minkowski sums. *Comp. Geom. Theory Appl.*, 21:39–61, 2002.
- [AGJ00] U. Adamy, J. Giesen, and M. John. New techniques for topologically correct surface reconstruction. In *Proc. IEEE Visualization*, pages 273–380, 2000.
- [AHU74] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, 1974.
- [AKM⁺87] A. Aggarwal, M.M. Klawe, S. Moran, P.W. Shor, and R. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2:195–208, 1987.
- [And79] A.M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Inform. Process. Lett.*, 9:216–219, 1979.
- [Aus98] M.H. Austern. *Generic Programming and the STL*. Addison-Wesley, Reading, 1998.
- [BBP01] H. Brönnimann, C. Burnikel, and S. Pion. Interval arithmetic yields efficient dynamic filters for computational geometry. *Discrete Appl. Math.*, 109:25–47, 2001.
- [BDH96] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22:469–483, 1996.
- [BDP⁺02] J.-D. Boissonnat, O. Devillers, S. Pion, M. Teillaud, and M. Yvinec. Triangulations in CGAL. *Comput. Geom. Theory Appl.*, 22:5–19, 2002.
- [BFMS00] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica*, 27:87–99, 2000.
- [Bie95] H. Bieri. Nef polyhedra: A brief introduction. *Computing Suppl. Springer-Verlag*, 10:43–60, 1995.
- [BK89] M. Blum and S. Kannan. Designing programs that check their work. In *Proc. 21th Annu. ACM Sympos. Theory Comput.*, pages 86–97, 1989.
- [BKSV00] H. Brönnimann, L. Kettner, S. Schirra, and R.C. Veltkamp. Applications of the generic programming paradigm in the design of CGAL. In M. Jazayeri, R. Loos, and D. Musser, editors, *Generic Programming—Proc. Dagstuhl Seminar*, volume 1766 of *Lecture Notes Comput. Sci.*, pages 206–217, Springer-Verlag, 2000.
- [BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proc. 22nd Annu. ACM Sympos. Theory of Computing*, pages 73–83, 1990.
- [BMS94a] C. Burnikel, K. Mehlhorn, and S. Schirra. How to compute the Voronoi diagram of line segments: Theoretical and experimental results. In *Proc. 2nd Annu. European Sympos. Algorithms*, volume 855 of *Lecture Notes Comput. Sci.*, pages 227–239. Springer-Verlag, Berlin, 1994.
- [BMS94b] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994.
- [BN02] M. Bäsken and S. Näher. Geowin—a generic tool for interactive visualization of geometric algorithms. In S. Diehl, editor, *Software Visualization*, volume 2269 of *Lecture Notes Comput. Sci.*, pages 88–100. Springer-Verlag, Berlin, 2002.
- [CLR90] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

- [CMS93] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3:185–212, 1993.
- [CSdVY02] D. Cohen-Steiner, É. Colin de Verdière, and M. Yvinec. Conforming Delaunay triangulations in 3D. In *Proc. 18th ACM Sympos. Comput. Geom.*, pages 109–208, 2002.
- [dBvK⁺00] M. de Berg, M. van Kreveld, M.H. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*, 2nd edition. Springer-Verlag, Berlin, 2000.
- [Dev02] O. Devillers. The Delaunay hierarchy. *Internat. J. Found. Comput. Sci.*, 13:163–180, 2002.
- [DG01] T.K. Dey and J. Giesen. Detecting undersampling in surface reconstruction. In *Proc. 17th ACM Sympos. Comput. Geom.*, pages 257–263, 2001.
- [DPT02] O. Devillers, S. Pion, and M. Teillaud. Walking in a triangulation. *Internat. J. Found. Comput. Sci.*, 13:181–199, 2002.
- [DT03] O. Devillers and M. Teillaud. Perturbations and vertex removal in a 3D Delaunay triangulation. In *Proc. 14th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 313–319, 2003.
- [EHS02] E. Ezra, D. Halperin, and M. Sharir. Speeding up the incremental construction of the union of geometric objects in practice. In *Proc. 10th European Sympos. Algorithms*, pages 473–484, 2002.
- [EM94] H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graph.*, 13:43–72, 1994.
- [ES96] H. Edelsbrunner and N.R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.
- [FG03] K. Fischer and B. Gärtner. The smallest enclosing ball of balls: Combinatorial structure and algorithms. In *Proc. 19th Annu. ACM Sympos. Comput. Geom.*, pages 292–301, 2003.
- [FGK⁺96] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr. The CGAL kernel: A basis for geometric computation. In M.C. Lin and D. Manocha, editors, *Proc. 1st ACM Workshop Appl. Comput. Geom.*, volume 1148 of *Lecture Notes Comput. Sci.*, pages 191–202. Springer-Verlag, Berlin, 1996.
- [FGK⁺00] A. Fabri, G.-J. Giezeman, L. Kettner, S. Schirra, and S. Schönherr. On the design of CGAL a computational geometry algorithms library. *Softw.—Pract. Exp.*, 30:1167–1202, 2000.
- [FHH⁺00] E. Flato, D. Halperin, I. Hanniel, O. Nechushtan, and E. Ezra. The design and implementation of planar maps in CGAL. *The ACM J. Experimental Algorithms*, 5, 2000. Also in *Lecture Notes Comput. Sci.*, volume 1668, Springer-Verlag, Berlin, pages 154–168.
- [FJ84] G.N. Frederickson and D.B. Johnson. Generalized selection and ranking: sorted matrices. *SIAM J. Comput.*, 13:14–30, 1984.
- [FM02] S. Funke and K. Mehlhorn. Look: A lazy object-oriented kernel for geometric computation. *Comput. Geom. Theory Appl.*, 22(1–3):99–118, 2002.
- [Gär99] B. Gärtner. Fast and robust smallest enclosing balls. In *Proc. 7th annu. European Sympos. Algorithms*, volume 1643, *Lecture Notes Comput. Sci.*, pages 325–338. Springer-Verlag, Berlin, 1999.
- [GJ02] J. Giesen and M. John. Surface reconstruction based on a dynamical system. In *Proc. Eurographics 2002*, 2002.

- [Gra72] R.L. Graham. An efficient algorithm for determining the convex hulls of a finite point set. *Inform. Process. Lett.*, 1:132–133, 1972.
- [Gra02] T. Granlund. *GNU MP, The GNU Multiple Precision Arithmetic Library*, 4.1 edition, May 2002. manual, <http://www.swox.com/gmp>.
- [Gre83] D.H. Greene. The decomposition of polygons into convex parts. In F.P. Preparata, editor, *Computational Geometry*, volume 1 of *Adv. Comput. Res.*, pages 235–259. JAI Press, Greenwich, 1983.
- [GS97] B. Gärtner and S. Schönherr. Exact primitives for smallest enclosing ellipses. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 430–432, 1997.
- [GS98a] B. Gärtner and S. Schönherr. Smallest enclosing circles—An exact and generic implementation in C++. Tech. Rep. B 98–04, Informatik, Freie Universität Berlin, Germany, 1998.
- [GS98b] B. Gärtner and S. Schönherr. Smallest enclosing ellipses—An exact and generic implementation in C++. Tech. Rep. B 98–05, Informatik, Freie Universität Berlin, Germany, 1998.
- [GS00] B. Gärtner and S. Schönherr. An efficient, exact, and generic quadratic programming solver for geometric optimization. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 110–118, 2000.
- [Hal02] D. Halperin. Robust geometric computing in motion. *Internat. J. Robotics Research*, 21:219–232, 2002.
- [Han91] E.N. Hanson. The interval skip list: a data structure for finding all intervals that overlap a point. In *Proc. 2nd Workshop Algorithms Data Struct., Lecture Notes Comput. Sci.*, volume 519, pages 153–164. Springer-Verlag, Berlin, 1991.
- [HH00] I. Hanniel and D. Halperin. Two-dimensional arrangements in CGAL and adaptive point location for parametric curves. In *Proc. 4th Workshop Algorithm Eng.*, 2000.
- [HH02] S. Hirsch and D. Halperin. Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane. In *Proc. 5th Workshop Algorithmic Found. Robot.*, pages 225–241, Nice, 2002.
- [HHK⁺01] S. Hertel, M. Hoffmann, L. Kettner, S. Pion, and M. Seel. An adaptable and extensible geometry kernel. In *Proc. Workshop Algorithm Eng.*, volume 2141 of *Lecture Notes Comput. Sci.*, pages 79–90. Springer-Verlag, Berlin, 2001.
- [HM83] S. Hertel and K. Mehlhorn. Fast triangulation of simple polygons. In *Proc. 4th Internat. Conf. Found. Comput. Theory*, volume 158 of *Lecture Notes Comput. Sci.*, pages 207–218. Springer-Verlag, Berlin, 1983.
- [HMN96] C. Hundack, K. Mehlhorn, and S. Näher. A simple linear time algorithm for identifying Kuratowski subgraphs of non-planar graphs. Unpublished, 1996.
- [Hof99] M. Hoffmann. A simple linear algorithm for computing rectangular three-centers. In *Proc. 11th Canad. Conf. Comput. Geom.*, pages 72–75, 1999.
- [HS95] G.R. Hjaltason and H. Samet. Ranking in spatial databases. In *Proc. 4th Interat. Sympos. Advances Spatial Databases, Lecture Notes Comput. Sci.*, volume 951, pages 83–95. Springer-Verlag, Berlin, 1995.
- [Ket98] L. Kettner. Designing a data structure for polyhedral surfaces. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 146–154, 1998.
- [Kin90] J.H. Kingston. *Algorithms and Data Structures*. Addison-Wesley, Reading, 1990.
- [KKM⁺03] L. Kettner, D.G. Kirkpatrick, A. Mantler, J. Snoeyink, B. Speckmann, and F. Takeuchi. Tight degree bounds for pseudo-triangulations of points. *Comput. Geom. Theory Appl.*, 25(1–2):3–12, 2003.

- [KLPY99] V. Karamcheti, C. Li, I. Pechtchanski, and C.K. Yap. A core library for robust numeric and geometric computation. In *15th ACM Sympos. Comput. Geom.*, pages 351–359, 1999.
- [KY02] M. Karavelas and M. Yvinec. Dynamic additively weighted Voronoi diagrams in 2D. In *Proc. 10th European Sympos. Algorithms*, pages 586–598, 2002.
- [LPVV01] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 80–89, 2001.
- [Meh84] K. Mehlhorn. *Data Structures and Algorithms 1,2, and 3*. Springer-Verlag, Berlin, 1984.
- [MM95] K. Mehlhorn and P. Mutzel. On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm. *Algorithmica*, 16:233–242, 1995.
- [MMN⁺98] K. Mehlhorn, M. Müller, S. Näher, S. Schirra, M. Seel, C. Uhrig, and J. Ziegler. A computational basis for higher-dimensional computational geometry and applications. *Comput. Geom. Theory Appl.*, 10:289–303, 1998.
- [MN94a] K. Mehlhorn and S. Näher. Implementation of a sweep line algorithm for the straight line segment intersection problem. Tech. Rep. MPI-I-94-160, Max-Planck-Institut für Informatik, 1994.
- [MN94b] K. Mehlhorn and S. Näher. The implementation of geometric algorithms. In *Proc. 13th IFIP World Computer Congress*, volume 1, pages 223–231. Elsevier North-Holland, Amsterdam, 1994.
- [MN00] K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, 2000.
- [MNS⁺99] K. Mehlhorn, S. Näher, M. Seel, R. Seidel, T. Schilz, S. Schirra, and C. Uhrig. Checking geometric programs or verification of geometric structures. *Comput. Geom. Theory Appl.*, 12:85–103, 1999.
- [MNSU] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. The LEDA User Manual. Tech. Rep., Max-Planck-Institut für Informatik. <http://www.mpi-sb.mpg.de/LEDA/leda.html>.
- [MS01] K. Mehlhorn and M. Seel. Infimaximal frames: A technique for making lines look like segments. In *17th European Workshop Comput. Geom.*, pages 78–81. Freie Universität Berlin, 2001.
- [Mul90] K. Mulmuley. A fast planar partition algorithm, I. *J. Symbolic Comput.*, 10(3–4):253–280, 1990.
- [Nef78] W. Nef. *Beiträge zur Theorie der Polyeder*. Herbert Lang, Bern, 1978.
- [NH93] J. Nievergelt and K. Hinrichs. *Algorithms and Data Structures*. Prentice-Hall, Englewood Cliffs, 1993.
- [O'R98] J. O'Rourke. *Computational Geometry in C*, second edition. Cambridge University Press, 1998.
- [Ove96] M.H. Overmars. Designing the Computational Geometry Algorithms Library CGAL. In *Proc. 1st ACM Workshop Appl. Comput. Geom.*, volume 1148 of *Lecture Notes Comput. Sci.*, pages 53–58. Springer-Verlag, Berlin, 1996.
- [Raa99] S. Raab. Controlled perturbation for arrangements of polyhedral surfaces with application to swept volumes. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1999.
- [Sch96] S. Schirra. Designing a computational geometry algorithms library. *Lecture Notes for Advanced School on Algorithmic Foundations of Geographic Information Systems*, CISIM, Udine, 1996.

- [Sch99] S. Schirra. A case study on the cost of geometric computing. In M.T. Goodrich and C.C. McGeoch, editors, *Algorithm Engineering and Experimentation (Proc. ALENEX '99)*, volume 1619 of *Lecture Notes Comput. Sci.*, pages 156–176. Springer-Verlag, Berlin, 1999.
- [Sed91] R. Sedgewick. *Algorithms*. Addison-Wesley, Reading, 1991.
- [See94] M. Seel. *Eine Implementierung abstrakter Voronoidiagramme*. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, 1994.
- [See01] M. Seel. Implementation of planar Nef polyhedra. Research Report MPI-I-2001-1-003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, 2001.
- [Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Comput. Geom. Theory Appl.*, 1:51–64, 1991.
- [STV⁺95] C. Schwarz, J. Teich, A. Vainshtein, E. Welzl, and B.L. Evans. Minimal enclosing parallelogram with application. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C34–C35, 1995.
- [SW96] M. Sharir and E. Welzl. Rectilinear and polygonal p -piercing and p -center problems. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 122–132, 1996.
- [Tar83] R.E. Tarjan. Data structures and network algorithms. In *CBMS-NSF Regional Conf. Series in Applied Mathematics*, volume 44, 1983.
- [Tou83] G.T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON 83*, pages A10.02/1–4, 1983.
- [van88] C.J. van Wyk. *Data Structures and C Programs*. Addison-Wesley, Reading, 1988.
- [Vel97] R.C. Veltkamp. Generic programming in CGAL, the computational geometry algorithms library. In *Proc. 6th Eurographics Workshop Programming Paradigms in Graphics*, 1997.
- [vOV02] R. van Oostrum and R.C. Veltkamp. Parametric search made practical. In *Proc. 18th ACM Sympos. Comput. Geom.*, pages 1–10, 2002.
- [Wei02] R. Wein. High level filtering for arrangements of conic arcs. In *Proc. 10th European Sympos. Algorithms*, volume 2461 of *Lecture Notes Comput. Sci.*, pages 884–895. Springer-Verlag, Rome, 2002.
- [Wel91] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes Comput. Sci.*, pages 359–370. Springer-Verlag, Berlin, 1991.
- [Woo93] D. Wood. *Data Structures, Algorithms, and Performance*. Addison-Wesley, Reading, 1993.