

George Bebis et al. (Eds.)

LNCS 8034

Advances in Visual Computing

9th International Symposium, ISVC 2013
Rethymnon, Crete, Greece, July 2013
Proceedings, Part II

2
Part II



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

TU Dortmund University, Germany

Madhu Sudan

Microsoft Research, Cambridge, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

George Bebis Richard Boyle
Bahram Parvin Darko Koracin Baoxin Li
Fatih Porikli Victor Zordan
James Klosowski Sabine Coquillart
Xun Luo Min Chen David Gotz (Eds.)

Advances in Visual Computing

9th International Symposium, ISVC 2013
Rethymnon, Crete, Greece, July 29-31, 2013
Proceedings, Part II

Volume Editors

George Bebis, E-mail: bebis@cse.unr.edu

Richard Boyle, E-mail: richard.boyle@nasa.gov

Bahram Parvin, E-mail: parvin@hpcrd.lbl.gov

Darko Koracin, E-mail: darko@dri.edu

Baoxin Li, E-mail: baoxin.li@asu.edu

Fatih Porikli, E-mail: fatih@merl.com

Victor Zordan, E-mail: vbz@cs.ucr.edu

James Klosowski, E-mail: jklosow@research.att.com

Sabine Coquillart, E-mail: sabine.coquillart@inria.fr

Xun Luo, E-mail: xun.luo@ieee.org

Min Chen, E-mail: min.chen@oerc.ox.ac.uk

David Gotz, E-mail: dgotz@us.ibm.com

ISSN 0302-9743

e-ISSN 1611-3349

ISBN 978-3-642-41938-6

e-ISBN 978-3-642-41939-3

DOI 10.1007/978-3-642-41939-3

Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013951706

CR Subject Classification (1998): I.3-5, H.5.2, I.2.10, J.3, F.2.2, I.3.5

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

© Springer-Verlag Berlin Heidelberg 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

It is with great pleasure that we welcome you all to the 9th International Symposium on Visual Computing (ISVC 2013) in Rethymnon, Crete, Greece. ISVC provides a common umbrella for the four main areas of visual computing including vision, graphics, visualization, and virtual reality. The goal is to provide a forum for researchers, scientists, engineers, and practitioners throughout the world to present their latest research findings, ideas, developments, and applications in the broader area of visual computing.

This year, the program consists of 11 oral sessions, one poster session, 6 special tracks, and 6 keynote presentations. The response to the call for papers was very good; we received over 220 submissions for the main symposium from which we accepted 63 papers for oral presentation and 35 papers for poster presentation. Special track papers were solicited separately through the Organizing and Program Committees of each track. A total of 32 papers were accepted for oral presentation in the special tracks.

All papers were reviewed with an emphasis on their potential to contribute to the state-of-the-art in the field. Selection criteria included accuracy and originality of ideas, clarity and significance of results, and presentation quality. The review process was quite rigorous, involving two - three independent blind reviews followed by several days of discussion. During the discussion period we tried to correct anomalies and errors that might have existed in the initial reviews. Despite our efforts, we recognize that some papers worthy of inclusion may not have been included in the program. We offer our sincere apologies to authors whose contributions might have been overlooked.

We wish to thank everybody who submitted their work to ISVC 2012 for review. It was because of their contributions that we succeeded in having a technical program of high scientific quality. In particular, we would like to thank the ISVC 2013 area chairs, the organizing institutions (UNR, DRI, LBNL, and NASA Ames), the industrial sponsors (BAE Systems, Intel, Ford, Hewlett Packard, Mitsubishi Electric Research Labs, Toyota, General Electric), the International Program Committee, the special track organizers and their Program Committees, the keynote speakers, the reviewers, and especially the authors that contributed their work to the symposium. In particular, we would like to express our appreciation to MERL and Dr. Fatih Porikli for their sponsorship of the “best” paper award this year.

We sincerely hope that ISVC 2013 will offer opportunities for professional growth. We wish you a pleasant time in Crete, Greece.

July 2013

George Bebis
Richard Boyle
Bahram Parvin
Darko Koracin
Baoxin Li
Fatih Porikli
Victor Zordan
James Klosowski
Sabine Coquillart
Xun Luo
Min Chen
David Gotz

Organization

ISVC 2013 Steering Committee

Bebis George	University of Nevada at Reno, USA
Boyle Richard	NASA Ames Research Center, USA
Parvin Bahram	Lawrence Berkeley National Laboratory, USA
Koracin Darko	Desert Research Institute, USA

ISVC 2013 Area Chairs

Computer Vision

Li Baoxin	Arizona State University, USA
Porikli Fatih	Mitsubishi Electric Research Labs (MERL), USA

Computer Graphics

Zordan Victor	University of California at Riverside, USA
Klosowski James	AT&T Research Labs, USA

Virtual Reality

Coquillart Sabine	Inria, France
Luo Xun	Qualcomm Research, USA

Visualization

Chen Min	University of Oxford, UK
Gotz David	IBM, USA

Publicity

Erol Ali	ASELSAN, Turkey
----------	-----------------

Local Arrangements

Zaboulis Xenophon	ICS-FORTH, Greece
-------------------	-------------------

Special Tracks

Wang Junxian Microsoft, USA

ISVC 2013 Keynote Speakers

Zorin Dennis	New York University, USA
Belongie Serge	University of California at San Diego, USA
Ertl Thomas	University of Stuttgart, Germany
Hoogs Anthony	Kitware, USA
Metaxas Dimitris	Rutgers University, USA
Slater Mel	ICREA-University of Barcelona, Spain

ISVC 2013 International Program Committee

(Area 1) Computer Vision

Abidi Besma	University of Tennessee at Knoxville, USA
Abou-Nasr Mahmoud	Ford Motor Company, USA
Aggarwal J.K.	University of Texas at Austin, USA
Albu Branzan Alexandra	University of Victoria, Canada
Amayeh Gholamreza	Foveon, USA
Ambardekar Amol	Microsoft, USA
Agouris Peggy	George Mason University, USA
Argyros Antonis	University of Crete, Greece
Asari Vijayan	University of Dayton, USA
Athitsos Vassilis	University of Texas at Arlington, USA
Basu Anup	University of Alberta, Canada
Bekris Kostas	Rutgers University, USA
Bhatia Sanjiv	University of Missouri-St. Louis, USA
Bimber Oliver	Johannes Kepler University Linz, Austria
Bourbakis Nikolaos	Wright State University, USA
Brimkov Valentin	State University of New York, USA
Campadelli Paola	University of Milan, Italy
Cavallaro Andrea	Queen Mary, University of London, UK
Charalampidis Dimitrios	University of New Orleans, USA
Chellappa Rama	University of Maryland, USA
Chen Yang	HRL Laboratories, USA
Cheng Hui	Sarnoff Corporation, USA
Cheng Shinko	HRL Labs, USA
Cremers Daniel	Technical University of Munich, Germany
Cui Jinshi	Peking University, China
Dagher Issam	University of Balamand, Lebanon
Darbon Jerome	CNRS-Ecole Normale Superieure de Cachan, France

Demirdjian David	Vecna Robotics, USA
Duan Ye	University of Missouri-Columbia, USA
Doulamis Anastasios	Technical University of Crete, Greece
El-Ansari Mohamed	Ibn Zohr University, Morocco
El-Gammal Ahmed	University of New Jersey, USA
Eng How Lung	Institute for Infocomm Research, Singapore
Erol Ali	ASELSAN, Turkey
Fan Guoliang	Oklahoma State University, USA
Fan Jialue	Northwestern University, USA
Ferri Francesc	University of Valencia, Spain
Ferryman James	University of Reading, UK
Foresti GianLuca	University of Udine, Italy
Fowlkes Charless	University of California at Irvine, USA
Fukui Kazuhiro	The University of Tsukuba, Japan
Galata Aphrodite	The University of Manchester, UK
Georgescu Bogdan	Siemens, USA
Goh Wooi-Boon	Nanyang Technological University, Singapore
Guerra-Filho Gutemberg	Intel, USA
Guevara Angel Miguel	University of Porto, Portugal
Gustafson David	Kansas State University, USA
Hammoud Riad	BAE Systems, USA
Harville Michael	Hewlett Packard Labs, USA
He Xiangjian	University of Technology, Australia
Heikkil Janne	University of Oulu, Finland
Hongbin Zha	Peking University, China
Hou Zujun	Institute for Infocomm Research, Singapore
Hua Gang	IBM T.J. Watson Research Center, USA
Imiya Atsushi	Chiba University, Japan
Jia Kevin	IGT, USA
Kamberov George	Stevens Institute of Technology, USA
Kampel Martin	Vienna University of Technology, Austria
Kamberova Gerda	Hofstra University, USA
Kakadiaris Ioannis	University of Houston, USA
Kettebekov Sanzhar	Keane inc., USA
Kimia Benjamin	Brown University, USA
Kisacanin Branislav	Texas Instruments, USA
Klette Reinhard	Auckland University, New Zealand
Kokkinos Iasonas	Ecole Centrale de Paris, France
Kollias Stefanos	National Technical University of Athens, Greece
Komodakis Nikos	Ecole Centrale de Paris, France
Kozintsev Igor	Intel, USA
Kuno Yoshinori	Saitama University, Japan
Kim Kyungnam	HRL Laboratories, USA
Latecki Longin Jan	Temple University, USA
Lee D.J.	Brigham Young University, USA

Levine Martin	McGill University, Canada
Li Chunming	Vanderbilt University, USA
Li Xiaowei	Google Inc., USA
Lim Ser N.	GE Research, USA
Lisin Dima	VidoeIQ, USA
Lee Hwee Kuan	Bioinformatics Institute A*STAR, Singapore
Lee Seong-Whan	Korea University, South Korea
Leung Valerie	MathWorks, France
Li Shuo	GE Healthcare, Canada
Lourakis Manolis	ICS-FORTH, Greece
Loss Leandro	Lawrence Berkeley National Lab, USA
Luo Gang	Harvard University, USA
Ma Yunqian	Honeywell Labs, USA
Maeder Anthony	University of Western Sydney, Australia
Makrogiannis Sokratis	NIH, USA
Maltoni Davide	University of Bologna, Italy
Maybank Steve	Birkbeck College, UK
Medioni Gerard	University of Southern California, USA
Melenchon Javier	Universitat Oberta de Catalunya, Spain
Metaxas Dimitris	Rutgers University, USA
Ming Wei	Konica Minolta Laboratory, USA
Mirmehdi Majid	Bristol University, UK
Monekosso Dorothy	University of Ulster, UK
Morris Brendan	University of Nevada at Las Vegas, USA
Mueller Klaus	Stony Brook University, USA
Muhammad Ghulam	King Saud University, Saudi Arabia
Mulligan Jeff	NASA Ames Research Center, USA
Murray Don	Point Grey Research, Canada
Nait-Charif Hammadi	Bournemouth University, UK
Nefian Ara	NASA Ames Research Center, USA
Nicolescu Mircea	University of Nevada at Reno, USA
Nixon Mark	University of Southampton, UK
Nolle Lars	The Nottingham Trent University, UK
Ntalianis Klimis	National Technical University of Athens, Greece
Or Siu Hang	The Chinese University of Hong Kong, Hong Kong
Papadourakis George	Technological Education Institute, Greece
Papanikopoulos Nikolaos	University of Minnesota, USA
Pati Peeta Basa	CoreLogic, India
Patras Ioannis	Queen Mary, University of London, UK
Pavlidis Ioannis	University of Houston, USA
Petrakis Euripides	Technical University of Crete, Greece
Peyronnet Sylvain	University Paris-Sud, France
Pinhanez Claudio	IBM Research, Brazil

Piccardi Massimo	University of Technology, Australia
Pietikainen Matti	LRDE/University of Oulu, Finland
Pitas Ioannis	Aristotle University of Thessaloniki, Greece
Porikli Fatih	Mitsubishi Electric Research Labs, USA
Prabhakar Salil	DigitalPersona Inc., USA
Prati Andrea	University IUAV of Venice, Italy
Prokhorov Danil	Toyota Research Institute, USA
Qian Gang	Arizona State University, USA
Rafopoulos Kostas	National Technical University of Athens, Greece
Regazzoni Carlo	University of Genoa, Italy
Regentova Emma	University of Nevada at Las Vegas, USA
Remagnino Paolo	Kingston University, UK
Ribeiro Eraldo	Florida Institute of Technology, USA
Robles-Kelly Antonio	National ICT Australia, Australia
Ross Arun	Michigan State University, USA
Samal Ashok	University of Nebraska, USA
Samir Tamer	Ingersoll Rand Security Technologies, USA
Sandberg Kristian	Computational Solutions, USA
Sarti Augusto	DEI Politecnico di Milano, Italy
Savakis Andreas	Rochester Institute of Technology, USA
Schaefer Gerald	Loughborough University, UK
Scalzo Fabien	University of California at Los Angeles, USA
Scharcanski Jacob	UFRGS, Brazil
Shah Mubarak	University of Central Florida, USA
Shi Pengcheng	Rochester Institute of Technology, USA
Shimada Nobutaka	Ritsumeikan University, Japan
Singh Rahul	San Francisco State University, USA
Skurikhin Alexei	Los Alamos National Laboratory, USA
Souvenir Richard	University of North Carolina at Charlotte, USA
Su Chung-Yen	National Taiwan Normal University, Taiwan (R.O.C.)
Sugihara Kokichi	University of Tokyo, Japan
Sun Zehang	Apple, USA
Syeda-Mahmood Tanveer	IBM Almaden, USA
Tan Kar Han	Hewlett Packard, USA
Tan Tieniu	Chinese Academy of Sciences, China
Tavakkoli Alireza	University of Houston at Victoria, USA
Tavares, Joao	Universidade do Porto, Portugal
Teoh Eam Khwang	Nanyang Technological University, Singapore
Thiran Jean-Philippe	Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
Tistarelli Massimo	University of Sassari, Italy
Tong Yan	University of South Carolina, USA
Tsechpenakis Gabriel	Indiana University - Purdue University Indianapolis, USA

Tsui T.J.	Chinese University of Hong Kong, Hong Kong
Trucco Emanuele	University of Dundee, UK
Tubaro Stefano	Politecnico di Milano, Italy
Uhl Andreas	Salzburg University, Austria
Velastin Sergio	Kingston University London, UK
Veropoulos Kostantinos	GE Healthcare, Greece
Verri Alessandro	University of Genoa, Italy
Wang C.L. Charlie	The Chinese University of Hong Kong, Hong Kong
Wang Junxian	Microsoft, USA
Wang Song	University of South Carolina, USA
Wang Yunhong	Beihang University, China
Webster Michael	University of Nevada at Reno, USA
Wolff Larry	Equinox Corporation, USA
Wong Kenneth	The University of Hong Kong, Hong Kong
Xiang Tao	Queen Mary, University of London, UK
Xue Xinwei	Fair Isaac Corporation, USA
Xu Meihe	University of California at Los Angeles, USA
Yang Ming-Hsuan	University of California at Merced, USA
Yang Ruigang	University of Kentucky, USA
Yi Lijun	SUNY at Binghamton, USA
Yu Ting	GE Global Research, USA
Yu Zeyun	University of Wisconsin-Milwaukee, USA
Yuan Chunrong	University of Tuebingen, Germany
Zabulis Xenophon	ICS-FORTH, Greece
Zervakis Michalis	Technical University of Crete, Greece
Zhang Yan	Delphi Corporation, USA
Ziou Djamel	University of Sherbrooke, Canada
Zhou Huiyu	Queen's University Belfast, UK
Abd Rahni Mt Piah	Universiti Sains Malaysia, Malaysia
Abram Greg	Texas Advanced Computing Center, USA
Adamo-Villani Nicoletta	Purdue University, USA
Agu Emmanuel	Worcester Polytechnic Institute, USA
Andres Eric	University of Poitiers, France
Artusi Alessandro	Universitat de Girona, Spain
Baciu George	Hong Kong PolyU, Hong Kong
Balcisoy Selim Saffet	Sabanci University, Turkey
Barneva Reneta	State University of New York, USA
Belyaev Alexander	Heriot-Watt University, UK
Benes Bedrich	Purdue University, USA
Berberich Eric	Max-Planck Institute, Germany
Bilalis Nicholas	Technical University of Crete, Greece
Bimber Oliver	Johannes Kepler University Linz, Austria
Bouatouch Kadi	University of Rennes I, France
Brimkov Valentin	State University of New York, USA

Brown Ross	Queensland University of Technology, Australia
Bruckner Stefan	Vienna University of Technology, Austria
Callahan Steven	University of Utah, USA
Capin Tolga	Bilkent University, Turkey
Chaudhuri Parag	Indian Institute of Technology Bombay, India
Chen Min	University of Oxford, UK
Cheng Irene	University of Alberta, Canada
Chiang Yi-Jen	Polytechnic Institute of New York University, USA
Choi Min-Hyung	University of Colorado at Denver, USA
Comba Joao	Univ. Fed. do Rio Grande do Sul, Brazil
Crawfis Roger	Ohio State University, USA
Cremer Jim	University of Iowa, USA
Culbertson Bruce	HP Labs, USA
Dana Kristin	Rutgers University, USA
Debattista Kurt	University of Warwick, UK
Deng Zhigang	University of Houston, USA
Dick Christian	Technical University of Munich, Germany
Dingliana John	Trinity College, Ireland
El-Sana Jihad	Ben Gurion University of The Negev, Israel
Entezari Alireza	University of Florida, USA
Fabian Nathan	Sandia National Laboratories, USA
De Floriani Leila	University of Genoa, Italy
Fuhrmann Anton	VRVis Research Center, Austria
Gaither Kelly	University of Texas at Austin, USA
Gao Chunyu	Epson Research and Development, USA
Geist Robert	Clemson University, USA
Gelb Dan	Hewlett Packard Labs, USA
Gooch Amy	University of Victoria, Canada
Gu David	Stony Brook University, USA
Guerra-Filho Gutemberg	Intel, USA
Habib Zulfiqar	COMSATS Institute of Information Technology, Pakistan
Hadwiger Markus	KAUST, Saudi Arabia
Haller Michael	Upper Austria University of Applied Sciences, Austria
Hamza-Lup Felix	Armstrong Atlantic State University, USA
Han JungHyun	Korea University, South Korea
Hand Randall	Lockheed Martin Corporation, USA
Hao Xuejun	Columbia University and NYSPI, USA
Hernandez Jose Tiberio	Universidad de los Andes, Colombia
Hou Tingbo	Google Inc., USA
Huang Jian	University of Tennessee at Knoxville, USA
Huang Mao Lin	University of Technology, Australia
Huang Zhiyong	Institute for Infocomm Research, Singapore
Hussain Muhammad	King Saud University, Saudi Arabia

Jeschke Stefan	Vienna University of Technology, Austria
Jones Michael	Brigham Young University, USA
Julier Simon J.	University College London, UK
Kakadiaris Ioannis	University of Houston, USA
Kamberov George	Stevens Institute of Technology, USA
Ko Hyeong-Seok	Seoul National University, South Korea
Kolingerova Ivana	University of West Bohemia, Czech Republic
Lai Shuhua	Virginia State University, USA
Lewis R. Robert	Washington State University, USA
Li Bo	Samsung, USA
Li Frederick	University of Durham, UK
Lindstrom Peter	Lawrence Livermore National Laboratory, USA
Linsen Lars	Jacobs University, Germany
Loviscach Joern	Fachhochschule Bielefeld, Germany
Magnor Marcus	TU Braunschweig, Germany
Martin Ralph	Cardiff University, UK
Meenakshisundaram Gopi	University of California-Irvine, USA
Mendoza Cesar	NaturalMotion Ltd., USA
Metaxas Dimitris	Rutgers University, USA
Mudur Sudhir	Concordia University, Canada
Musuvathy Suraj	Siemens, USA
Myles Ashish	University of Florida, USA
Nait-Charif Hammadi	University of Dundee, Scotland
Nasri Ahmad	American University of Beirut, Lebanon
Noh Junyong	KAIST, South Korea
Noma Tsukasa	Kyushu Institute of Technology, Japan
Okada Yoshihiro	Kyushu University, Japan
Olague Gustavo	CICESE Research Center, Mexico
Oliveira Manuel M.	Univ. Fed. do Rio Grande do Sul, Brazil
Owen Charles	Michigan State University, USA
Ostromoukhov Victor M.	University of Montreal, Canada
Pascucci Valerio	University of Utah, USA
Patchett John	Los Alamos National Lab, USA
Peters Jorg	University of Florida, USA
Pronost Nicolas	Utrecht University, The Netherlands
Qin Hong	Stony Brook University, USA
Rautek Peter	Vienna University of Technology, Austria
Razdan Anshuman	Arizona State University, USA
Rosen Paul	University of Utah, USA
Rosenbaum Rene	University of California at Davis, USA
Rudomin Isaac	Barcelona Supercomputing Center, Spain
Rushmeier Holly	Yale University, USA
Sander Pedro	The Hong Kong University of Science and Technology, Hong Kong
Sapidis Nickolas	University of Western Macedonia, Greece
Sarfraz Muhammad	Kuwait University, Kuwait

Scateni Riccardo	University of Calgiari, Italy
Schaefer Scott	Texas A&M University, USA
Sequin Carlo	University of California-Berkeley, USA
Shead Timothy	Sandia National Laboratories, USA
Sourin Alexei	Nanyang Technological University, Singapore
Stamminger Marc	REVES/Inria, France
Su Wen-Poh	Griffith University, Australia
Szumilas Lech	Research Institute for Automation and Measurements, Poland
Tan Kar Han	Hewlett Packard, USA
Tarini Marco	University dell'Insubria, Italy
Teschner Matthias	University of Freiburg, Germany
Torchelsen Rafael Piccin	Universidade Federal da Fronteira Sul, Brazil
Umlauf Georg	HTWG Constance, Germany
Vanegas Carlos	University of California at Berkeley, USA
Wald Ingo	University of Utah, USA
Walter Marcelo	UFRGS, Brazil
Wimmer Michael	Technical University of Vienna, Austria
Wylie Brian	Sandia National Laboratory, USA
Wyman Chris	University of Calgary, Canada
Wyvill Brian	University of Iowa, USA
Yang Qing-Xiong	University of Illinois at Urbana, USA
Yang Ruigang	University of Kentucky, USA
Ye Duan	University of Missouri-Columbia, USA
Yi Beifang	Salem State University, USA
Yin Lijun	Binghamton University, USA
Yoo Terry	National Institutes of Health, USA
Yuan Xiaoru	Peking University, China
Zhang Jian Jun	Bournemouth University, UK
Zeng Jianmin	Nanyang Technological University, Singapore
Zara Jiri	Czech Technical University in Prague, Czech Republic

(Area 3) Virtual Reality

Alcaiz Mariano	Technical University of Valencia, Spain
Arns Laura	Purdue University, USA
Balcisoy Selim	Sabanci University, Turkey
Behringer Reinhold	Leeds Metropolitan University, UK
Benes Bedrich	Purdue University, USA
Bilalis Nicholas	Technical University of Crete, Greece
Blach Roland	Fraunhofer Institute for Industrial Engineering, Germany
Blom Kristopher	University of Barcelona, Spain
Bogdanovych Anton	University of Western Sydney, Australia
Brady Rachael	Duke University, USA
Brega Jose Remo Ferreira	Universidade Estadual Paulista, Brazil

Brown Ross	Queensland University of Technology, Australia
Bues Matthias	Fraunhofer IAO in Stuttgart, Germany
Capin Tolga	Bilkent University, Turkey
Chen Jian	Brown University, USA
Cooper Matthew	University of Linkiping, Sweden
Coquillart Sabine	Inria, France
Craig Alan	NCSA University of Illinois at Urbana, USA
Cremer Jim	University of Iowa, USA
Edmunds Timothy	University of British Columbia, Canada
Egges Arjan	Universiteit Utrecht, The Netherlands
Encarnao L. Miguel	ACT Inc., USA
Figueroa Pablo	Universidad de los Andes, Colombia
Fox Jesse	Stanford University, USA
Friedman Doron	IDC, Israel
Fuhrmann Anton	VRVis Research Center, Austria
Gregory Michelle	Pacific Northwest National Lab, USA
Gupta Satyandra K.	University of Maryland, USA
Haller Michael	FH Hagenberg, Austria
Hamza-Lup Felix	Armstrong Atlantic State University, USA
Herbelin Bruno	EPFL, Switzerland
Hinkenjann Andre	Bonn-Rhein-Sieg University of Applied Sciences, Germany
Hollerer Tobias	University of California at Santa Barbara, USA
Huang Jian	University of Tennessee at Knoxville, USA
Huang Zhiyong	Institute for Infocomm Research, Singapore
Julier Simon J.	University College London, UK
Kaufmann Hannes	Vienna University of Technology, Austria
Kiyokawa Kiyoshi	Osaka University, Japan
Kozintsev Igor	Intel, USA
Kuhlen Torsten	RWTH Aachen University, Germany
Lee Cha	University of California at Santa Barbara, USA
Liere Robert van	CWI, The Netherlands
Malzbender Tom	Hewlett Packard Labs, USA
Mantler Stephan	VRVis Research Center, Austria
Molineros Jose	Teledyne Scientific and Imaging, USA
Muller Stefan	University of Koblenz, Germany
Owen Charles	Michigan State University, USA
Paelke Volker	Institut de Geomatica, Spain
Peli Eli	Harvard University, USA
Pettifer Steve	The University of Manchester, UK
Pronost Nicolas	Utrecht University, The Netherlands
Pugmire Dave	Los Alamos National Lab, USA
Qian Gang	Arizona State University, USA
Raffin Bruno	Inria, France
Raij Andrew	University of South Florida, USA
Richir Simon	Arts et Metiers ParisTech, France

Rodello Ildeberto	University of San Paulo, Brazil
Sandor Christian	University of South Australia, Australia
Sapidis Nickolas	University of Western Macedonia, Greece
Schulze Jurgen	University of California at San Diego, USA
Sherman Bill	Indiana University, USA
Slavik Pavel	Czech Technical University in Prague, Czech Republic
Sourin Alexei	Nanyang Technological University, Singapore
Steinicke Frank	University of Wurzburg, Germany
Suma Evan	University of Southern California, USA
Stamminger Marc	REVES/Inria, France
Srikanth Manohar	Indian Institute of Science, India
Vercher Jean-Louis	University de la Mediterranee, France
Wald Ingo	University of Utah, USA
Yu Ka Chun	Denver Museum of Nature and Science, USA
Yuan Chunrong	University of Tuebingen, Germany
Zachmann Gabriel	Clausthal University, Germany
Zara Jiri	Czech Technical University in Prague, Czech Republic
Zhang Hui	Indiana University, USA
Zhao Ye	Kent State University, USA

(Area 4) Visualization

Andrienko Gennady	Fraunhofer Institute IAIS, Germany
Avila Lisa	Kitware, USA
Apperley Mark	University of Waikato, New Zealand
Balzs Csbfalvi	Budapest University of Technology and Economics, Hungary
Brady Rachael	Duke University, USA
Benes Bedrich	Purdue University, USA
Bilalis Nicholas	Technical University of Crete, Greece
Bonneau Georges-Pierre	Grenoble University, France
Bruckner Stefan	Vienna University of Technology, Austria
Brown Ross	Queensland University of Technology, Australia
Bhler Katja	VRVis Research Center, Austria
Callahan Steven	University of Utah, USA
Chen Jian	Brown University, USA
Chen Min	University of Oxford, UK
Chiang Yi-Jen	Polytechnic Institute of New York University, USA
Cooper Matthew	University of Linkoping, Sweden
Chourasia Amit	University of California at San Diego, USA
Crossno Patricia	Sandia National Laboratories, USA
Daniels Joel	University of Utah, USA
Dick Christian	Technical University of Munich, Germany
Doleisch Helmut	SimVis GmbH, Austria

XVIII Organization

Duan Ye	University of Missouri-Columbia, USA
Dwyer Tim	Monash University, Australia
Entezari Alireza	University of Florida, USA
Ertl Thomas	University of Stuttgart, Germany
De Floriani Leila	University of Maryland, USA
Fujishiro Issei	Keio University, Japan
Geist Robert	Clemson University, USA
Gotz David	IBM, USA
Grinstein Georges	University of Massachusetts Lowell, USA
Goebel Randy	University of Alberta, Canada
Grg Carsten	University of Colorado at Denver, USA
Gregory Michelle	Pacific Northwest National Lab, USA
Hadwiger Helmut Markus	KAUST, Saudi Arabia
Hagen Hans	Technical University of Kaiserslautern, Germany
Hamza-Lup Felix	Armstrong Atlantic State University, USA
Healey Christopher	North Carolina State University at Raleigh, USA
Hochheiser Harry	University of Pittsburgh, USA
Hollerer Tobias	University of California at Santa Barbara, USA
Hong Lichan	University of Sydney, Australia
Hong Seokhee	Palo Alto Research Center, USA
Hotz Ingrid	Zuse Institute Berlin, Germany
Huang Zhiyong	Institute for Infocomm Research, Singapore
Jiang Ming	Lawrence Livermore National Laboratory, USA
Joshi Alark	Yale University, USA
Julier Simon J.	University College London, UK
Laramée Robert	Swansea University, UK
Lewis R. Robert	Washington State University, USA
Liere Robert van	CWI, The Netherlands
Lim Ik Soo	Bangor University, UK
Linsen Lars	Jacobs University, Germany
Liu Zhaping	University of Pennsylvania, USA
Ma Kwan-Liu	University of California at Davis, USA
Maeder Anthony	University of Western Sydney, Australia
Malpica Jose	Alcala University, Spain
Masutani Yoshitaka	The University of Tokyo Hospital, Japan
Matkovic Kresimir	VRVis Research Center, Austria
McCaffrey James	Microsoft Research / Volt VTE, USA
Melancon Guy	CNRS UMR 5800 LaBRI and Inria Bordeaux Sud-Ouest, France
Miksch Silvia	Vienna University of Technology, Austria
Monroe Laura	Los Alamos National Labs, USA
Morie Jacki	University of Southern California, USA
Moreland Kenneth	Sandia National Laboratories, USA
Mudur Sudhir	Concordia University, Canada

Museth Ken	Linkpings University, Sweden
Paelke Volker	Institut de Geomatica, Spain
Papka Michael	Argonne National Laboratory, USA
Peikert Ronald	Swiss Federal Institute of Technology Zurich, Switzerland
Pettifer Steve	The University of Manchester, UK
Pugmire Dave	Los Alamos National Lab, USA
Rabin Robert	University of Wisconsin at Madison, USA
Raffin Bruno	Inria, France
Razdan Anshuman	Arizona State University, USA
Rhyne Theresa-Marie	North Carolina State University, USA
Rosenbaum Rene	University of California at Davis, USA
Scheuermann Gerik	University of Leipzig, Germany
Shead Timothy	Sandia National Laboratories, USA
Shen Han-Wei	Ohio State University, USA
Sips Mike	Stanford University, USA
Slavik Pavel	Czech Technical University in Prague, Czech Republic
Sourin Alexei	Nanyang Technological University, Singapore
Thakur Sidharth	Renaissance Computing Institute (RENCI), USA
Theisel Holger	University of Magdeburg, Germany
Thiele Olaf	University of Mannheim, Germany
Toledo de Rodrigo	Petrobras PUC-RIO, Brazil
Tricoche Xavier	Purdue University, USA
Umlauf Georg	HTWG Constance, Germany
Viegas Fernanda	IBM, USA
Wald Ingo	University of Utah, USA
Wan Ming	Boeing Phantom Works, USA
Weinkauf Tino	Max-Planck-Institut fuer Informatik, Germany
Weiskopf Daniel	University of Stuttgart, Germany
Wischgoll Thomas	Wright State University, USA
Wylie Brian	Sandia National Laboratory, USA
Wu Yin	Indiana University, USA
Xu Wei	Stony Brook University, USA
Yeasin Mohammed	Memphis University, USA
Yuan Xiaoru	Peking University, China
Zachmann Gabriel	Clausthal University, Germany
Zhang Hui	Indiana University, USA
Zhao Ye	Kent State University, USA
Zheng Ziyi	Stony Brook University, USA
Zhukov Leonid	Caltech, USA

ISVC 2013 Special Tracks

1. Computational Bioimaging

Organizers:

Tavares Joo Manuel R.S.	University of Porto, Portugal
Natal Jorge Renato	University of Porto, Portugal
Cunha Alexandre	Caltech, USA

2. 3D Mapping, Modeling and Surface Reconstruction

Organizers:

Nefian Ara	Carnegie Mellon University/NASA Ames Research Center, USA
Edwards Laurence	NASA Ames Research Center, USA
Huertas Andres	NASA Jet Propulsion Lab, USA
Visentin Gianfranco	ESA European Space Research and Technology Centre, The Netherlands
Lourakis Manolis	Foundation for Research and Technology, Greece
Chliveros Georgios	Foundation for Research and Technology, Greece

3. Visual Computing in Digital Cultural Heritage

Organizers:

Doulamis Anastasios D.	Technical University of Crete, Greece
Doulamis Nikolaos D.	National Technical University of Athens, Greece
Ioannides Marinos	Cyprus University of Technology, Cyprus
Georgopoulos Andreas	National Technical University of Athens, Greece
Voulodimos Athanasios	National Technical University of Athens, Greece

4. Sparse Methods for Computer Vision, Graphics and Medical Imaging

Organizers:

Metaxas Dimitris	Rutgers University, USA
Axel Leon	New York University, USA
Zhang Shaoting	Rutgers University, USA

5. Visual Computing with Multimodal Data Streams

Organizers:

Zhang Hui	Indiana University, USA
Du Yingzi	Indiana University-Purdue University Indianapolis, USA
Boyles Mike	Indiana University, USA
Wernert Eric	Indiana University, USA

Thakur Sidharth
Ruan Guangchen

Renaissance Computing Institute, USA
Indiana University, USA

6. Intelligent Environments: Algorithms and Applications

Organizers:

Bebis George
Nicolescu Mircea
Bourbakis Nikolaos
Tavakkoli Alireza

University of Nevada at Reno, USA
University of Nevada at Reno, USA
Wright State University, USA
University of Houston at Victoria, USA

Organizing Institutions and Sponsors



imagination at work



Table of Contents – Part II

Visualization II

The Reflection Layer Extension to the Stereoscopic Highlight Technique for Node-Link Diagrams: An Empirical Study	1
<i>Ragaad AlTarauneh, Jens Bauer, Shah Rukh Humayoun, Patric Keller, and Achim Ebert</i>	
Adaptive Semantic Visualization for Bibliographic Entries	13
<i>Kawa Nazemi, Reimond Retz, Jürgen Bernard, Jörn Kohlhammer, and Dieter Fellner</i>	
A Methodology for Interactive Spatial Visualization of Automotive Function Architectures for Development and Maintenance	25
<i>Moritz Cohrs, Stefan Klimke, and Gabriel Zachmann</i>	
Navigation Recommendations for Exploring Hierarchical Graphs	36
<i>Stefan Gladisch, Heidrun Schumann, and Christian Tominski</i>	
A Tool for Visualizing Large-Scale Interactions between Turbulence and Particles in 3D Space through 2D Trajectory Visualization	48
<i>Guoyu Lu, Vincent Ly, Xiaolong Wang, Rohith M.V., Orlando Ayala, Lian-Ping Wang, and Chandra Kambhamettu</i>	

ST: Visual Computing with Multimodal Data Streams

Visual Query Specification and Interaction with Industrial Engineering Data	58
<i>Alberto Malagoli, Mariano Leva, Stephen Kimani, Alessandro Russo, Massimo Mecella, Sonia Bergamaschi, and Tiziana Catarci</i>	
Performance Anchored Score Normalization for Multi-biometric Fusion	68
<i>Naser Damer, Alexander Opel, and Alexander Nouak</i>	
Towards a Contextualized Visual Analysis of Heterogeneous Manufacturing Data	76
<i>Mario Aehnelt, Hans-Jörg Schulz, and Bodo Urban</i>	

Visual Statistics Cockpits for Information Gathering in the Policy-Making Process	86
<i>Dirk Burkhardt, Kawa Nazemi, Christian Stab, Martin Steiger, Arjan Kuijper, and Jörn Kohlhammer</i>	

ST: Visual Computing in Digital Cultural Heritage

Feature Weight Optimization and Pruning in Historical Text Recognition	98
<i>Fredrik Wahlberg and Anders Brun</i>	
A Constraint Inductive Learning- Spectral Clustering Methodology for Personalized 3D Navigation	108
<i>Nikolaos Doulamis, Christos Yiakoumettis, George Miaoulis, and Eftychios Protopapadakis</i>	
Beat Synchronous Dance Animation Based on Visual Analysis of Human Motion and Audio Analysis of Music Tempo	118
<i>Costas Panagiotakis, Andre Holzapfel, Damien Michel, and Antonis A. Argyros</i>	

Combining Unsupervised Clustering with a Non-linear Deformation Model for Efficient Petroglyph Recognition	128
<i>Vincenzo Deufemia and Luca Paolino</i>	

Analysing User Needs for a Unified 3D Metadata Recording and Exploitation of Cultural Heritage Monuments System.....	138
<i>E. Maravelakis, A. Konstantaras, A. Kritsotaki, D. Angelakis, and M. Xinogalos</i>	

Precise 3D Reconstruction of Cultural Objects Using Combined Multi-component Image Matching and Active Contours Segmentation	148
<i>Christos Stentoumis, Georgios Livanos, Anastasios Doulamis, Eftychios Protopapadakis, Lazaros Grammatikopoulos, and Michael Zervakis</i>	

ST: Intelligent Environments: Algorithms and Applications

People Tracking Based on Predictions and Graph-Cuts Segmentation ...	158
<i>Amira Soudani and Ezzeddine Zagrouba</i>	
A Framework for Quick and Accurate Access of Interesting Visual Events in Surveillance Videos	168
<i>Fei Yuan, Chu Tang, Shu Tian, and Hongwei Hao</i>	

Detecting and Tracking Unknown Number of Objects with Dirichlet Process Mixture Models and Markov Random Fields	178
<i>Ibrahim Saygin Topkaya, Hakan Erdogan, and Fatih Porikli</i>	

Grassmannian Spectral Regression for Action Recognition	189
<i>Sherif Azary and Andreas Savakis</i>	

Layered RC Circuit Model for Background Subtraction	199
<i>Karel Mozdřeň, Eduard Sojka, Radovan Fusek, and Milan Šurkala</i>	

Pairwise Kernels for Human Interaction Recognition	210
<i>Saeid Motlian, Ke Feng, Harika Bharthavarapu, Sajid Sharlemin, and Gianfranco Doretto</i>	

Applications

A Vision-Based Algorithm for Parking Lot Utilization Evaluation Using Conditional Random Fields	222
<i>Tomas Fabian</i>	

Automatic Pain Intensity Estimation with Heteroscedastic Conditional Ordinal Random Fields	234
<i>Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic</i>	

Robot Trajectory Planning Using OLP and Structured Light 3D Machine Vision	244
<i>M. Rodrigues, M. Kormann, C. Schuhler, and P. Tomek</i>	

Improving Accessibility of Virtual Worlds by Automatic Object Labeling	254
<i>Ilias Apostolopoulos, Eelke Folmer, and George Bebis</i>	

Hierarchical Image Geo-location on a World-Wide Scale	266
<i>Alexandru N. Vasile and Octavia Camps</i>	

An Image Based Approach for Content Analysis in Document Collections	278
<i>Reinhold Huber-Mörk and Alexander Schindler</i>	

Virtual Reality

Simultaneous Bidirectional Geometric Model Synchronization between CAD and VR Applications	288
<i>Dimo Chotrov and Stoyan Maleshkov</i>	

A Hand-Held 3-D Display System with Haptic Sensation	298
<i>Kai Ki Lee, Kin-Hong Wong, Michael Ming-Yuen Chang, and Ying-Kin Yu</i>	

XXVI Table of Contents – Part II

Primitive Human Action Recognition Based on Partitioned Silhouette Block Matching	308
<i>Toru Abe, Masaru Fukushi, and Daisuke Ueda</i>	
Fast and Accurate Unknown Object Segmentation for Robotic Systems	318
<i>Lazaros Nalpantidis, Bjarne Großmann, and Volker Krüger</i>	
Differential Progressive Path Tracing for High-Quality Previsualization and Relighting in Augmented Reality	328
<i>Peter Kán and Hannes Kaufmann</i>	
Projection on Suitable Sub-surface Selected in Indoor Environment	339
<i>Shafaq Mussadiq and Rehan Hafiz</i>	

Visualization III

A Framework for the Visualization of Finite-Time Continuum Mechanics Effects in Time-Varying Flow	349
<i>Alexy Agranovsky, Harald Obermaier, and Kenneth I. Joy</i>	
Visual Access to Optimization Problems in Strategic Environmental Assessment	361
<i>Tobias Ruppert, Jürgen Bernard, Alex Ulmer, Arjan Kuijper, and Jörn Kohlhammer</i>	
Mesh Generation from Layered Depth Images Using Isosurface Raycasting	373
<i>Steffen Frey, Filip Sadlo, and Thomas Ertl</i>	
FractVis: Visualizing Microseismic Events	384
<i>Ahmed E. Mostafa, Sheelagh Carpendale, Emilio Vital Brazil, David Eaton, Ehud Sharlin, and Mario Costa Sousa</i>	
Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm	396
<i>Gwenael Bothorel, Mathieu Serrurier, and Christophe Hurter</i>	

Poster

Automatically Extracting Hairstyles from 2D Images	406
<i>Chuan-Kai Yang and Chia-Ning Kuo</i>	
Evaluation of Image Forgery Detection Using Multi-scale Weber Local Descriptors	416
<i>Sahar Q. Saleh, Muhammad Hussain, Ghulam Muhammad, and George Bebis</i>	

Energy-Transfer Features for Pedestrian Detection	425
<i>Radovan Fusek, Eduard Sojka, Karel Mozdřen, and Milan Šurkala</i>	
Basic Shape Classification Using Spatially Normalised Fourier Shape Signature	435
<i>Chin Yeow Wong, Stephen Ching-Feng Lin, Guannan Jiang, and Ngai Ming Kwok</i>	
Normalized Matting of Interest Region	446
<i>Jaehwan Kim and Ilkwon Jeong</i>	
Speeding Up SURF	454
<i>Peter Abeles</i>	
Distortion Adaptive Image Classification – An Alternative to Barrel-Type Distortion Correction	465
<i>Michael Gadermayr, Andreas Uhl, and Andreas Vécsei</i>	
Moving Horizon Estimation of Pedestrian Interactions Based on Multiple Velocity Fields	475
<i>Ana Portelo, Sandra Pacheco, Mário A.T. Figueiredo, João M. Lemos, and Jorge S. Marques</i>	
Evaluating and Comparing of 3D Shape Descriptors for Object Recognition	484
<i>Alexander Ceron and Flavio Prieto</i>	
Gender Recognition Using Fusion of Local and Global Facial Features	493
<i>Anwar M. Mirza, Muhammad Hussain, Huda Almuzaini, Ghulam Muhammad, Hatim Aboalsamh, and George Bebis</i>	
Curvelet Transform and Local Texture Based Image Forgery Detection	503
<i>Muneer H. Al-Hammadi, Ghulam Muhammad, Muhammad Hussain, and George Bebis</i>	
Camera Distance from Face Images	513
<i>Arturo Flores, Eric Christiansen, David Kriegman, and Serge Belongie</i>	
Towards Robust Gait Recognition	523
<i>Tenika P. Whytock, Alexander Belyaev, and Neil M. Robertson</i>	
Direct Encoding for Sampled Color Pictures with Location Consideration	532
<i>Chulhee Lee, Jaehoon Lee, and Guiwon Seo</i>	

XXVIII Table of Contents – Part II

Real-Time Hand Gesture Recognition for Uncontrolled Environments Using Adaptive SURF Tracking and Hidden Conditional Random Fields	542
<i>Yi Yao and Chang-Tsun Li</i>	
Examination of Hybrid Image Feature Trackers	552
<i>Peter Abeles</i>	
3D Shape Estimation Based on Sparsity in Stereo Matching	562
<i>Naoto Hirose, Tatsuki Yasunobe, and Akira Kawanaka</i>	
Color Image Compression by Riemannian B-Tree Triangular Coding	572
<i>Olfa Triki and Mourad Zéraï</i>	
Human Tracking and Counting Using the KINECT Range Sensor Based on Adaboost and Kalman Filter	582
<i>Lei Zhu and Kin-Hong Wong</i>	
Hand Pose Estimation from a Single RGB-D Image	592
<i>Alina Kuznetsova and Bodo Rosenhahn</i>	
3D Human Tracking in a Top View Using Depth Information Recorded by the Xtion Pro-Live Camera	603
<i>Cyrille Mignot and Fakhreddine Ababsa</i>	
Determination of Object Directions Using Optical Flow for Crowd Monitoring	613
<i>Aravinda S. Rao, Jayavaradhana Gubbi, Slaven Marusic, Andrew Maher, and Marimuthu Palaniswami</i>	
Evolutionary Techniques for Procedural Texture Automation	623
<i>Alaa Eldin M. Ibrahim</i>	
Voxel-Based Harmonic Map for Voxel-Based Model Deformation/Manipulation	633
<i>Tomoaki Nagaoka</i>	
A Novel Approach to Retrieval of Similar Patterns in Biological Images	643
<i>Andrzej Sluzek</i>	
Variational Model for Image Segmentation	653
<i>Qiong Lou, Jialin Peng, Fa Wu, and Dexing Kong</i>	
Sky Segmentation by Fusing Clustering with Neural Networks	663
<i>Ali Pour Yazdanpanah, Emma E. Regentova, Ajay Kumar Mandava, Touqeer Ahmad, and George Bebis</i>	
An Interactive Web Based Spatio-Temporal Visualization System	673
<i>Anil Ramakrishna, Yu-Han Chang, and Rajiv Maheswaran</i>	

One-to-Two Digital Earth	681
<i>Ali Mahdavi Amiri, Faraz Bhojani, and Faramarz Samavati</i>	
Storygraph: Telling Stories from Spatio-temporal Data	693
<i>Ayush Shrestha, Ying Zhu, Ben Miller, and Yi Zhao</i>	
Organizing Visual Data in Structured Layout by Maximizing Similarity-Proximity Correlation	703
<i>Grant Strong, Rune Jensen, Minglun Gong, and Anne C. Elster</i>	
Mixing Geometrically Diverse Window Managers.....	714
<i>Anthony Savidis and Andreas Maragudakis</i>	
Classifier Comparison for Repeating Motion Based Video Classification	725
<i>Kahraman Ayyildiz and Stefan Conrad</i>	
Implementation of Source Engine for Virtual Tours in Manufacturing Factories	737
<i>Petr Horejsi and Jiri Polcar</i>	
Evaluating 3D Vision for Command and Control Applications	747
<i>Britton Wolfe, Beomjin Kim, Benjamin Aeschliman, and Robert Sedlmeyer</i>	
Author Index	757

Table of Contents – Part I

ST: Computational Bioimaging I

What Is the Role of Color Symmetry in the Detection of Melanomas?	1
<i>Margarida Ruela, Catarina Barata, and Jorge S. Marques</i>	
Automatic Quantitative Assessment of the Small Bowel Motility with Cine-MRI Sequence Analysis	11
<i>Xing Wu, Shaojian Zhuo, and Wu Zhang</i>	
Pharynx Segmentation from MRI Data for Analysis of Sleep Related Disorders	20
<i>Tatyana Ivanovska, Johannes Dober, René Laqua, Katrin Hegenscheid, and Henry Völzke</i>	
Fully Automated Brain Tumor Segmentation Using Two MRI Modalities	30
<i>Mohamed Ben Salah, Idanis Diaz, Russell Greiner, Pierre Boulanger, Bret Hoehn, and Albert Murtha</i>	
Evaluation of Color Based Keypoints and Features for the Classification of Melanomas Using the Bag-of-Features Model	40
<i>Catarina Barata, Jorge S. Marques, and Jorge Rozeira</i>	
Barrel-Type Distortion Compensated Fourier Feature Extraction	50
<i>Michael Gadermayr, Andreas Uhl, and Andreas Vécsei</i>	

Computer Graphics I

Rotation-Aware LayerPaint System	60
<i>Jiazhai Xia, Shenghui Liao, and Juncong Lin</i>	
Digital Circlism as Algorithmic Art	69
<i>Sourav De and Partha Bhowmick</i>	
Color Edge Preserving Smoothing	79
<i>Ali Alsam and Hans Jakob Rivertz</i>	
Parallel 3D 12-Subiteration Thinning Algorithms Based on Isthmuses ...	87
<i>Kálmán Palágyi</i>	
Depth Peeling Algorithm for the Distance Field Computation of Overlapping Objects	99
<i>Marcin Ryciuk and Joanna Porter-Sobieraj</i>	

Evaluation of Rendering Algorithms Using Position-Dependent Scene Properties.....	108
<i>Claudius Jähn, Benjamin Eikel, Matthias Fischer, Ralf Petring, and Friedhelm Meyer auf der Heide</i>	

Motion, Tracking, and Recognition

Improving Robustness and Precision in GEI + HOG Action Recognition	119
<i>Tenika P. Whytock, Alexander Belyaev, and Neil M. Robertson</i>	
A Unified Framework for 3D Hand Tracking	129
<i>Rudra P.K. Poudel, Jose A.S Fonseca, Jian J. Zhang, and Hammadi Nait-Charif</i>	
A Multiple Velocity Fields Approach to the Detection of Pedestrians Interactions Using HMM and Data Association Filters	140
<i>Ricardo A. Ribeiro, Jorge S. Marques, and João M. Lemos</i>	
Human Activity Recognition Using Hierarchically-Mined Feature Constellations	150
<i>Antonios Oikonomopoulos and Maja Pantic</i>	
An Active Vision Approach to Height Estimation with Optical Flow	160
<i>Sotirios Ch. Diamantaras and Prithviraj Dasgupta</i>	
Structure Descriptor for Articulated Shape Analysis	171
<i>Li Han, Jiangyue Hu, and Lin Li</i>	

Segmentation

A Machine Learning Approach to Horizon Line Detection Using Local Features	181
<i>Touqeer Ahmad, George Bebis, Emma E. Regentova, and Ara Nefian</i>	
Pose Invariant Deformable Shape Priors Using L_1 Higher Order Sparse Graphs	194
<i>Bo Xiang, Nikos Komodakis, and Nikos Paragios</i>	
Connected Components Labeling on the GPU with Generalization to Voronoi Diagrams and Signed Distance Fields	206
<i>A. Rasmusson, T.S. Sørensen, and G. Ziegler</i>	
Foreground Detection with a Moving RGBD Camera	216
<i>P. Koutlemanis, X. Zabulis, A. Ntelidakis, and Antonis A. Argyros</i>	
Image Segmentation Using Iterated Graph Cuts with Residual Graph...	228
<i>Michael Holuša and Eduard Sojka</i>	

- Pressure Based Segmentation in Volumetric Images 238
Thamer S. Alathari and Mark S. Nixon

Visualization I

- On Connectedness of Discretized Objects 246
Valentin E. Brimkov
- Visualizing 3D Time-Dependent Foam Simulation Data 255
Dan R. Lipşa, Robert S. Laramee, Simon Cox, and I. Tudur Davies
- Analyzing and Reducing DTI Tracking Uncertainty by Combining Deterministic and Stochastic Approaches 266
Khoa Tan Nguyen, Anders Ynnerman, and Timo Ropinski
- TimeExplorer: Similarity Search Time Series by Their Signatures 280
Tuan Nhon Dang and Leland Wilkinson
- A New Visual Comfort-Based Stereoscopic Image Retargeting Method 290
Sang-Hyun Cho and Hang-Bong Kang

ST: 3D Mapping, Modeling and Surface Reconstruction

- Simultaneous Color Camera and Depth Sensor Calibration with Correction of Triangulation Errors 301
Jae-Hean Kim, Jin Sung Choi, and Bon-Ki Koo
- Improving Image-Based Localization through Increasing Correct Feature Correspondences 312
Guoyu Lu, Vincent Ly, Haoquan Shen, Abhishek Kolagunda, and Chandra Kambhamettu
- Reconstructing Plants in 3D from a Single Image Using Analysis-by-Synthesis 322
Jérôme Guénard, Géraldine Morin, Frédéric Boudon, and Vincent Charvillat
- Rapid Disparity Prediction for Dynamic Scenes 333
Jun Jiang, Jun Cheng, and Baowen Chen
- A Solution to the Similarity Registration Problem of Volumetric Shapes 343
Wanmu Liu, Sasan Mahmoodi, Michael J. Bennett, and Tom Havelock
- 3D Surface Reconstruction Using Polynomial Texture Mapping 353
Mohammed Elfarargy, Amr Rizq, and Marwa Rashwan

Feature Extraction, Matching and Recognition

Keypoint Detection and Matching on High Resolution Spherical Images	363
<i>Christiano Couto Gava, Jean-Marc Hengen, Bertram Taetz, and Didier Stricker</i>	
Scene Perception and Recognition in Industrial Environments for Human-Robot Interaction	373
<i>Nikhil Soman, Emmanuel Dean-León, Caixia Cai, and Alois Knoll</i>	
Good Appearance and Shape Descriptors for Object Category Recognition	385
<i>Pedro F. Proença, Filipe Gaspar, and Miguel Sales Dias</i>	
Object Recognition for Service Robots through Verbal Interaction Based on Ontology	395
<i>Hisato Fukuda, Satoshi Mori, Yoshinori Kobayashi, Yoshinori Kuno, and Daisuke Kachi</i>	
Corner Detection in Spherical Images via the Accelerated Segment Test on a Geodesic Grid	407
<i>Hao Guan, William A.P. Smith, and Peng Ren</i>	
Object Categorization in Context Based on Probabilistic Learning of Classification Tree with Boosted Features and Co-occurrence Structure	416
<i>Masayasu Atsumi</i>	

Computer Graphics II

Reconstruction of Wire Structures from Scanned Point Clouds	427
<i>Kotaro Morioka, Yutaka Otake, and Hiromasa Suzuki</i>	
Real-Time Simulation of Vehicle Tracks on Soft Terrain	437
<i>Xiao Chen and Ying Zhu</i>	
Real-Time 3D Rendering of Heterogeneous Scenes	448
<i>Ralf Petring, Benjamin Eikel, Claudius Jähn, Matthias Fischer, and Friedhelm Meyer auf der Heide</i>	
Sketch-Based Image Warping Interface	459
<i>Jiazhi Xia and Zhi-Quan Cheng</i>	
Saliency-Guided Color Transfer between Images	468
<i>Jiazhi Xia</i>	
Memory Efficient Shortest Path Algorithms for Cactus Graphs	476
<i>Boris Brimkov</i>	

ST: Sparse Methods for Computer Vision, Graphics and Medical Imaging

Localization of Multi-pose and Occluded Facial Features via Sparse Shape Representation	486
<i>Yang Yu, Shaoting Zhang, Fei Yang, and Dimitris Metaxas</i>	
Collaborative Sparse Representation in Dissimilarity Space for Classification of Visual Information	496
<i>Ilias Theodorakopoulos, George Economou, and Spiros Fotopoulos</i>	
A Novel Technique for Space-Time-Interest Point Detection and Description for Dance Video Classification	507
<i>Soumitra Samanta and Bhabatosh Chanda</i>	
Efficient Transmission and Rendering of RGB-D Views	517
<i>Zahid Riaz, Thorsten Linder, Sven Behnke, Rainer Worst, and Hartmut Surmann</i>	
Face Processing and Recognition	
Shared Gaussian Process Latent Variable Model for Multi-view Facial Expression Recognition	527
<i>Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic</i>	
Face Verification Using Local Binary Patterns and Maximum A Posteriori Vector Quantization Model.....	539
<i>Elhocine Boutellaa, Farid Harizi, Messaoud Bengherabi, Samy Ait-Aoudia, and Abdenour Hadid</i>	
Face Box Shape and Verification	550
<i>Eric Christiansen, Iljung S. Kwak, Serge Belongie, and David Kriegman</i>	
3D Face Pose and Animation Tracking via Eigen-Decomposition Based Bayesian Approach.....	562
<i>Ngoc-Trung Tran, Fakhr-Eddine Ababsa, Maurice Charbit, Jacques Feldmar, Dijana Petrovska-Delacrétaz, and Gérard Chollet</i>	
Local Orientation Patterns for 3D Surface Texture Analysis of Normal Maps: Application to Facial Skin Condition Classification	572
<i>Alassane Seck, Hannah Dee, and Bernard Tiddeman</i>	
Author Index	583

The Reflection Layer Extension to the Stereoscopic Highlight Technique for Node-Link Diagrams: An Empirical Study

Ragaad AlTarawneh¹, Jens Bauer¹, Shah Rukh Humayoun¹,
Patric Keller², and Achim Ebert¹

¹ Computer Graphics and HCI Group, University of Kaiserslautern, Germany

² Software Engineering: Dependability Group, University of Kaiserslautern, Germany
`{tarawneh,j_bauer,humayoun,pkeller,ebert}@cs.uni-kl.de`

Abstract. The stereoscopic highlighting technique is a new emerging technique that supports using the depth cue in 3D devices to encode some attributes in general node-link diagrams. In this paper, we extend the original stereoscopic highlighting technique by adding an extra reflection layer at the bottom of the graph to help users detecting the highlighted nodes faster and more accurate. We measure the accuracy of this extension in the technique by evaluating the users' ability of reading the variation of nodes depth values using the proposed reflection layer. We carried out a controlled user-study. The results show that using the depth cue in stereoscopic devices is readable for medium data sizes with at most three or four different layers without the reflection layer, while they were able to read more complicated configurations through the support of the reflection layer with higher accuracy.

Keywords: Stereoscopic Highlighting Technique, Node-Link Diagrams, Information Visualization.

1 Introduction

Recently, the 3D display technology has been used in many daily life applications. We have witnessed common acceptance of this technology in the public. People are getting familiar with 3D movies or with 3D games. Moreover, the cost of such technologies becomes affordable with time due to the continuous decreasing of the hardware cost. Currently, the technologies behind the 3D displays are advancing at accelerated steps, resulting in high quality stereoscopic images. This gives a great opportunity to use these displays for visualizing many applications in the scientific visualization field or for encoding some aspects about the data in the information visualization field [1].

For example, visualizing a dense graph using the node-link diagram in a pleasant way is considered to be one of the challenges in the information visualization [2,3]. Due to the edge-crossings, the current state-of-the-art lacks a pleasant representation for large dense diagrams [1]. However, many heuristic solutions have been proposed to optimize this issue. One such solution is based on the

3D node-link layout [4], where the authors argued that the rotation could be used to identify the adjacency relations of certain elements. In spite of this advantage of the 3D layout, the perspective view in the 3D world could lead to wrong conclusions about the importance degree of some nodes. This happens as a result of rendering some nodes closer to the virtual viewpoint than the others, which increases the visual emphasis of closer nodes and consequently affects the viewer's perception. This problem appears mostly because of two reasons, either due to the graph layout algorithm or due to the viewer's viewpoint rather than the data set itself. Additionally, the 3D layout requires high amount of viewpoint navigation, e.g., when a node of interest is rendered at a further depth level the viewer needs to rotate, zoom in/out, or change the view-angle to get a better view of the required node. This affects on preserving the viewers mental map because the graph appears differently each time to the viewer [1].

These problems can be reduced if the graph is rendered in a stereoscopic environment, because such environment could eliminate the perspective issues and the navigation problems. In [1], Alper et al. did a comparison of using the depth cue provided by stereoscopic displays as a highlighting technique beside other static visual cues on 2D and 3D graph layouts. They proved the ability of users to read the stereoscopic depth and used it as a highlighting technique for showing the important nodes in the underlying graph. They claimed that using the stereoscopic depth as a highlighting technique helps in saving other static visual cues for other usages, especially with highly attributed data. However, it can be used only as an ordinal property while the other static visual cues can be used to highlight nominal properties. In our previous work [5], we have already used the stereoscopic depth as a cue to give a meaning to a group of nodes on the same depth value. Moreover, we used it to classify the nodes into different groups according to one attribute (e.g., the nodes weight), such that the nodes having the same weight value were in the same depth. The accuracy of the technique was evaluated through a controlled user-study with users from different backgrounds. The goal of that user study was to measure the different depth values that can be read by normal users easily. Results of that user study showed the feasibility of the technique in encoding some data attributes especially, the ordinal data attributes. However, the accuracy of a small graph size was 86% in average, which is insufficient for most of critical applications.

Utilizing the stereoscopic depth as a highlighting technique suffers from many potential drawbacks, e.g., users need more time to realize the depth variation in stereoscopic displays rather than using colors [5], which can be observed instantly. Therefore, the depth cue needs to be optimized such that users can read the depth variation faster and more accurately. In this work, we propose a solution to increase the accuracy of the stereoscopic highlighting technique by providing an extra layer, called *the Reflection Layer*. This reflection layer is achieved by projecting the set of nodes of the 3D node-link diagram onto a 2D plane on the bottom of the rendering space to increase the detection speed of the nodes in different layers, as shown in Figure 1. To prove this claim about our proposed solution, we conducted a controlled user-study, consisted of six tasks in

two categories, with and without the refection layer (see Figure 3 in Section 4.4). Each category contained three tasks with different configurations and settings. We measured the number of different depth layers that normal users from different backgrounds can detect without and with the proposed reflection layer. We were also interested in the accuracy and the average time to complete the detection.

The remainder of this paper is structured as follows. In Section 2, we summarize the related work. In Section 3, we explain our proposed reflection layer extension to the stereoscopic highlighting technique. In Section 4, we present our hypotheses and experiment settings. Moreover, we also discuss the outcomes of the conducted user-study. We conclude in Section 5.

2 Related Work

The 3D layouts have been used in many previous works. For example, the hyperbolic layout has been proposed by [6] for showing the information structure in the 3D world. In [7], Robertson et al. presented the cone tree layout method as a pure 3D layout algorithm for hierarchy structures. Beside this, space-division techniques have also been extended into 3D versions such as the Treecube technique [8], which is as an extension to the traditional Treemap algorithm [9]. Those examples were based on using the animated 3D visualization and the lightening to show the depth perception. The main drawback of using such visualization is the extra occlusions that might hinder the visibility of the layout elements. Moreover, the mentioned examples do not consider the variations in depth values while positioning the nodes in the 3D space, which effects the level of the data perception [10]. The 3D layout requires a special 3D display to perceive the correct impression. In a study evaluating the efficiency of 2D, 2.5D, and 3D layouts in physical and virtual platforms, it has been concluded that the 3D without stereoscopic effect does not help in utilizing the spatial memory of the third dimension [10]. As a result of this finding, many scientific visualization applications are getting benefit from 3D display devices because many of these applications objects inherently are 3D objects [11].

A 3D node-link visualization can reduce the edge crossing [4], which makes it easier to identify the adjacency of certain elements. This makes the task of counting nodes accessible from a particular node, quite easier. However, due to the perspective view, rendering a graph in 3D environments can lead to wrong conclusions about the importance degree for some nodes. This is because some nodes will be rendered closer to the virtual camera viewpoint than the others, which increases the visual emphasis of the closer nodes. This emphasis situation comes from the graph layout algorithm or from the viewer's viewpoint rather than from the data set. Moreover, 3D layouts require a high amount of viewpoint navigation. For example, when a node of interest is rendered at a further depth level the user needs to rotate, zoom or change the view angle to get a better view. This process affects on viewer to keep the mental map of the layout because the graph can appear differently each time [1].

Many efforts have also been done in showing the efficiency of using the 3D graph visualization with high resolution stereo displays [2]. According to their results, 3D graph visualization is recommended only when stereoscopic cues and real time rotations are supported. Many applications use stereoscopic effect as a technique to separate the set of images that share the same texture of the background. In addition to that, depth cues have been used to separate the overlapped labels in the 3D world in order to reduce the cluttering effect [12]. In [13], the variation in depth was used to reduce the cluttering of information by filtering the information. Whereas in [14], authors evaluated the preattentive visual features as a highlighting technique in virtual reality environment. They showed that the stereoscopic depth has the additional advantage of facilitating an intuitive interaction technique, as it positions the relevant information towards the user or pushes them away according to their importance degree.

Many examples have been provided by [15, 16] for showing the usability of using the third dimension in encoding different aspects of the data. This can be achieved by isolating a subset of a 2D graph representation to a separate layer, thus a 2.5D would be created. In this case, the third dimension can be used to encode aspects of the data rather than the relations among nodes [17]. For example, the third dimension in [17] has been used to show the evolution of the node-link diagram over time, where each layer encodes a 2D representation of the graph at a certain time. In another example by [16], the third dimension has been used to depict the hierarchical nesting of clusters in the graph using a set of transparent layers. However, above mentioned examples did not exploit the depth cue nor they provided a quantitative criteria to measure their approachs usability.

3 The Reflection Layer Extension

The stereoscopic highlighting technique, as proposed in [1], utilizes the stereoscopic depth for highlighting the important regions in a 2D diagram by bringing them closer to the viewer. This helps in reserving other highlighting attributes, like color or/and shape, for encoding other data properties. One of the main advantages of using this technique is the ability of achieving the *focus+context* views naturally. This technique brings the region of interests to the foreground while keeping the rest of the graph in the background; hence, it reveals fewer details. Moreover, the depth-cue has few advantages over other static visual cues, e.g., when elements in a set are associated together at the same depth level it gives a spatial relation to those associated elements. Also, based on the Gestalt theory the spatial association among the data elements is a stronger indication than the colors [18]. However, according to [1], nearly 8 ~ 10% of the population suffers from depth blindness problem. On the other hand, around 10% of men and less than 1% of women population also face some degree of color-blindness [19], which makes it also difficult for them to use the color as a cue.

In our previous work [5], we measured the accuracy of common users in detecting the number of depth layers in the stereoscopic displays. In that study,

users were unable to give highly accurate results for the nodes and their specified layers. The average accuracy was 85%, which is not sufficient for critical applications. This happened because many times users were unable to read the configurations properly. To tackle this problem, we propose a solution to extend the original stereoscopic highlighting technique via supporting more cues in order to help the users in detecting the depth values more accurately.

In [7], a set of approaches has been proposed to enhance users' perceptions in detecting the depth cue. Few examples are: the size of the node for indicating the distance of the node from the viewer, lighting cue for enhancing the depth variation such that the lighter nodes look closer to the viewer while the darker nodes look farther, the shadow for conveying information about the hierarchy of the tree. To enhance the viewer's perception in detecting the 3D impression, we propose using a *reflection layer* at the bottom of the graph to support the viewer in detecting the depth cue faster and more accurately.

Our enhancement to the stereoscopic highlighting technique is based on adding a reflection layer with additional characteristics such as *the layer line* and *the depth line*. Figure 1 shows a prototype that we implemented to evaluate our proposed solution. Below is the explanation of our extension:

- *The Reflection Layer*: The main extension in the original stereoscopic highlighting technique is the addition of a reflection layer at the bottom of the rendered graph. The purpose of this layer is to show the projection of nodes from the 3D space into a 2D plane orthogonal to the 3D rendered graph. Each node is placed on the layer according to its depth value. This reflection layer also indicates the number of the different depth layers, as it arranges the nodes according to their depth values.
- *The Layer's Line*: This is illustrated in Figure 1, where a horizontal line is drawn on the reflection layer to discern the specified layer.
- *The Depth Line*: This is an on-demand option. The viewer can use it to see the connection between the node in the 3D space and its reflection on the reflection layer. This is drawn as a dashed red line between the node and its projection.

4 The User Evaluation Study

We carried out a user-study for evaluating the accuracy of our proposed reflection layer extension to the stereoscopic highlighting technique. We performed the evaluation study through two main settings in order to check the effectiveness and the feasibility of our proposed solution on users' performance. The first setting was the standard one, i.e., without the reflection layer, as proposed in [5] while the second setting was with the reflection layer as we proposed in the previous section. In the study, users from different backgrounds evaluated the proposed solution in a controlled environment through task-based evaluation tests [19], and then provided an overall feedback through sets of open-ended and closed-ended questionnaires. The results of the study provide an indication of

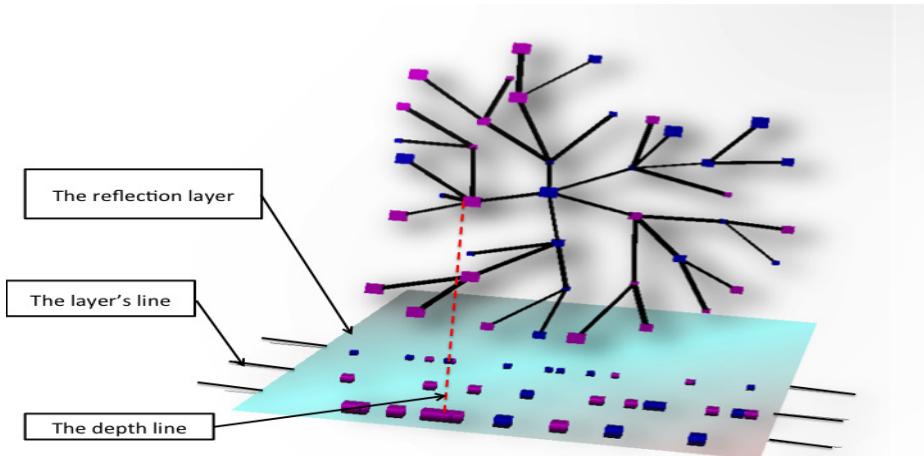


Fig. 1. The reflection layer enhancement example

an increment in the users' abilities in detecting the depth more accurately with less time using our extended technique compared to the standard one. This also indicates the usefulness of our reflection layer solution in reducing the limitations of the stereoscopic highlighting approach.

We carried out the evaluation experiment tests using a Zalman stereoscopic 3D display equipped with 3D glasses. The display size was 60 cm with 16:9 wide as a display ratio. The resolution was 1920 x 1080 pixels with 5ms response time. The frequency rate was 38 ~ 40 KHz in the horizontal and 56 ~ 75 Hz in the vertical mode. Each subject wore the polarized glasses by Zalman to view the 3D effects. These Zalman 3D glasses are designed to create stereoscopic optical 3D image with reduced eyestrain. They were seated directly in front of the 3D monitor. A pre-session was performed to insure that each user was able to see the 3D effects accurately. We allowed each user to adjust the height of the seat until he/she sees the 3D effects properly, with almost the same horizontal distance for all participants.

4.1 Goals

The goals of the user evaluation study were:

- To measure the users' abilities in perceiving the variation in depth values with and without the reflection layer.
- To know whether the depth variations can be used to encode ordinal data attributes beside other visual cues, like color or/and shape.
- To judge whether users are able to perceive the variation in depth values more accurately and quicker with passage of time after getting more

experience. Moreover, the comparison of time for each task with and without the reflection layer.

- Finally, to investigate the users abilities to interpret the depth variations into some meaningful information related to the underlying diagram.

4.2 Hypothesis

We hypothesized that using this reflection layer in the stereoscopic highlighting technique enhances the viewer’s ability to read the configuration faster and more accurately. Viewers could detect the number of different layers directly using this reflection layer and estimate the number of nodes associated with each layer faster and more accurately. Hence, the viewer’s perception is enhanced through using this reflection layer extension. Consequently, the viewer’s accuracy in detecting the depth variation is also increased. Here, we present the set of hypothesis that we diagnosed for our user study. Based on results of our previous study [5] we observed that users were able to read the variation in depth levels for small data set; so in this study we expected:

- **H1:** The accuracy value is irrelevant to the data size but it requires longer time for users to read a configuration with big data size.
- **H2:** The accuracy of detecting the variation in depth values will be significantly increased by adding an extra layer, the *reflection layer*, in the bottom that reflects the current configuration in the bottom of the 3D world.

4.3 The User Groups

The participants in our user study were from two backgrounds. The first group, the *vis-expert (visualization-expert)* group, consisted of participants having background in the visualization area. While the second group, the *non-expert* group, consisted of normal participants from other backgrounds. At the beginning of each experiment, we asked each participated user to provide some personal information including age, gender, sight/perception problems or any form of color blindness in order to analyze the results from different perspectives. We asked 8 participants (2 females, 6 males) to undergo our experiment. Out of these 8 participants, 4 were from the vis-expert group while the remaining 4 belonged to the non-expert group. The participants’ age ranged from 24 to 34 years with a mean age of 28.5 years. Four of these participants (two from each group) participated in the first evaluation study while the remaining four were participating for the first time. Moreover, half of the participants wear corrective glasses in their normal routine life.

4.4 The Experiment Layouts

Our initial 2D visualization shows a tree structure consisting of 41 nodes, as shown in Figure 2. In Figure 2a, we show the initial appearance of the layout in a 2D representation. While in Figure 2b and Figure 2c, we show the 3rd

dimension impression of the layout with different configurations. The layout is rendered using the VRUI package [20]. The color cue was used together with the depth, as in the previous study [5] we observed that different visual cues do not affect users' abilities in detecting the depth cue.

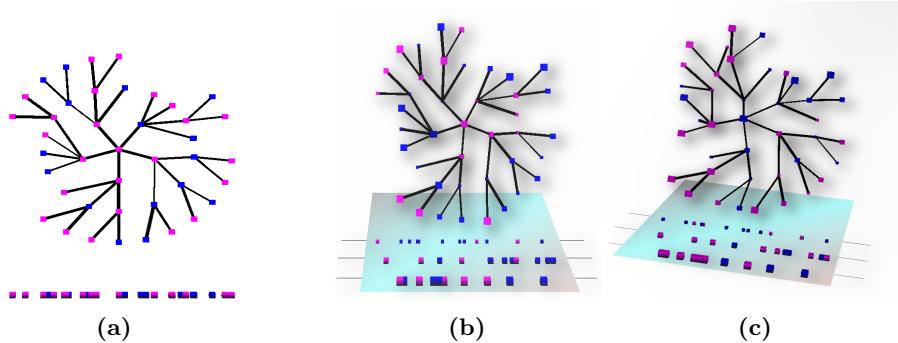


Fig. 2. An experiment's sample for the three layers configuration. In (a) we show the initial setting, in (b) we show the front view of the graph when the depth values being changed, and in (c) we show a side-view of the same configuration to visualize the 3D effect.

For each configuration we provided two possibilities, the first one was with the reflection layer while the second one was without it. To reduce the learning effect, the appearance of the reflection layer was randomly based from users' perspective. However, the order of the tasks appearance was carefully settled to ensure users abilities in reading both configurations. We kept the order of tasks same for all participants but the given sequence varied in difficulty, as we wanted to get results from an unbiased perspective. The participants had the option of skipping the task if they feel very confusing. Figure 3 provides the tasks configurations in our controlled study.

4.5 The Experiment Details and the Results

For the experiment, we added a simple animation during each test to indicate the changing process between the depth values of graph nodes in the set of configurations in real time. Each participated user started with a simple 2D node-link layout rendered in a 3D context, as it is shown in Figure 2a where all nodes have the same shape but with different colors. Then we changed the depth values for a predefined set of graph nodes to create a 2.5D representation of the configuration, as it is shown in Figure 2b. The users were able to see only the front view of the layout and were not allowed to rotate the layout. We tried to measure the accuracy of using this technique by giving a rough indication of the number of layers that a viewer can really read and can use to encode some information about the data itself. The results from the first study [5] indicated that most of the participants detected the three layers configuration

more accurately than the four or five layers configurations. We were curious whether the bottom reflection-scheme improves significantly the users' accuracy in detecting the number of layers and in picking the right number of nodes at these layers compared to the previous study, as well as the overall reduction in time for performing the tasks.

Task ID	No. of Layers	Using the Reflection
Task 1	3 layers	No
Task 2	4 layers	Yes
Task 3	4 layers	No
Task 4	5 layers	No
Task 5	3 layers	Yes
Task 6	5 layers	Yes

Fig. 3. The experiment tasks' configurations

At the end of all task-based tests, we gave each participant a closed-ended questionnaire form and an opened-ended questionnaire form in order to get their feedback and impressions regarding the overall environment and the proposed approach, especially related to the additional bottom reflection-scheme. The open-ended questionnaire form consisted of 13 questions. For each question, there were six options (*strongly agree*, *agree*, *neutral*, *disagree*, *strongly disagree*, and *don't know*) based on the likert scale. Each participant was given a maximum of 45 minutes to complete all the six tasks. This excludes the training time and the answering time to the questionnaires.

The results of this user-study provide an indication that the accuracy using the stereoscopic depth as a highlighting technique in stereoscopy devices is not affected much by the data size, although viewers need more time to find different depth levels and the number of nodes at each layer. When comparing the results of this study with the results of the previous study [5], we observe that the accuracy was 86% for the previous data size (16 nodes) while the accuracy of the current study data size (41 nodes) is 81% for the configurations without the reflection layer. We use this as an indication of the irrelevancy of the data size over the accuracy values of the detection.

The Figure 4a shows the results of all users' accuracy with two options, with and without the reflection layer. When analyzing the results of all configurations, we observe that the accuracy value is higher in the case of using the reflection layer than the tasks without using it. Overall, the average accuracy value for all those tasks that were supported by the reflection layer (i.e., task 2, task 5 and task 6) is 93% compared to 81% accuracy value for the set of tasks (i.e., task 1, task 3, and task 4) that were not supported by the reflection layer. The results

show, compared to the previous study [5] where the overall accuracy average for tasks with same color and shape was 86%, that relatively larger data set does not influence significantly to the accuracy level. In fact, the added bottom reflection layer increased the accuracy compared to the first study even though with larger data size and also compared to the tasks without bottom reflection layer support in this user study.

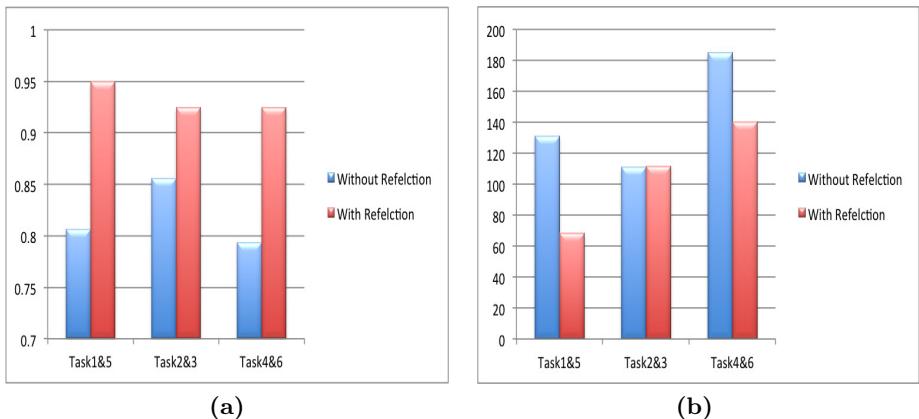


Fig. 4. (a) Average accuracy of all tasks for all user groups. (b) The average time of all tasks for all users groups.

The average time (see Figure 4b) for all tasks without the reflection layer support is 142.4 seconds (i.e., the combined mean time of task 1, task 3, and task 4), while the average time for tasks with the reflection layer support is 108.8 seconds (the combined mean time of task 2, task 5 and task 6). This indicates that the our proposed bottom reflection layer supports significantly the viewers in judging the number of layers, as through this, it is obvious for them to understand the depth levels in the underlying configuration. Moreover, it also helps the viewers in understanding and in counting the nodes at each layer faster compared to without the reflection layer support. Few of the participants mentioned in the open-ended questionnaires feedback that they used the bottom reflection layer when they were confused about the number of nodes at some particular layer. The standard deviation value for all participants' accuracy in all tasks without using the reflection layer was 0.018 while using the reflection layer was 0.007. This indicates the homogeneity in participants' answers. However, there is no much difference in respect to the deviation of the accuracy value in both options. However, there is a significant difference in the time deviation value. The standard deviation for the time in the tasks without using the reflection layer support is 16.46, which is relatively large value indicating the heterogeneity in their results. While the standard deviation for the time in the tasks with using the reflection layer support is 4.38. This indicates the high agreement level of participants' performance in all tasks.

By comparing the results of tasks with the same number of layers without and with the reflection layer support, we conclude that our proposed reflection layer solution improved significantly the participants' performance both in the accuracy and in the time performance. Overall, the results support our set of hypotheses and we can conclude that the data size does not affect the accuracy significantly. Moreover, over time after getting experienced with the proposed reflection layer support, it helps the viewers in understanding the depth levels and in picking the right component at the right depth level. In the questionnaires feedback, 7 out of 8 participants strongly agreed that our reflection layer scheme helped them in finding the number of depth levels. Moreover, 6 out of 8 participants agreed that they used this reflection layer in judging the depth levels and for picking the components at different layers. All of them said that they would prefer to work in an environment supported by our reflection layer approach.

5 Conclusion and Future Work

In this work, we provided an extended version of the stereoscopic highlighting technique for the node-link diagrams. The new extension provides a reflection layer at the bottom of the graph on which the graph nodes are projected. We evaluated the feasibility and performance of this reflection layer by conducting a user study, consisted of 6 different task-based tests and sets of open-ended and closed-ended questionnaires. However, the data size was not large enough to generalize the results. But, it gives an indication that additional cues, like the reflection layer, can increase the accuracy of the stereoscopic highlighting technique in stereoscopic devices in general.

In future, we aim to evaluate the possibility of increasing users' accuracy level in immersive environments through utilizing new interactive techniques. For this we plan to build a framework for providing an on demand depth cue, such that users could interact directly with the graph and could request for more depth cues to support the correctness of their judgment on the fly. This will give us an indication of the possibility of using the stereoscopic highlighting technique for critical information visualization applications.

Acknowledgements. This work is part of ViERforES2 project and partially funded by IRTG 1131 (DFG) and BMBF. Many thanks also go to the Software Engineering and the Dependability Group at University of Kaiserslautern for their support.

References

1. Alper, B., Hllerer, T., Kuchera-Morin, J., Forbes, A.: Stereoscopic highlighting: 2d graph visualization on stereo displays. *IEEE Trans. Vis. Comput. Graph.* 17, 2325–2333 (2011)
2. Ware, C., Franck, G.: Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graph.* 15, 121–140 (1996)

3. Ware, C., Bobrow, R.: Supporting visual queries on medium-sized nodelink diagrams. *Information Visualization* 4, 49–58 (2005)
4. Ware, C.: *Information Visualization: Perception for Design (Interactive Technologies)*, 1st edn. Morgan Kaufmann (2000)
5. AlTarawneh, R., Bauer, J., Humayoun, S.R., Keller, P., Ebert, A.: The extended stereoscopic highlighting technique for node-link diagrams: An empirical study. In: *Proceedings of the 14th IASTED International Conference on Computer Graphics and Imaging (CGIM 2013)*, Innsbruck, Austria (2013)
6. Munzner, T.: H3: laying out large directed graphs in 3d hyperbolic space. In: *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis 1997)*, p. 2. IEEE Computer Society, Washington, DC (1997)
7. Robertson, G.G., Mackinlay, J.D., Card, S.K.: Cone trees: animated 3d visualizations of hierarchical information. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1991)*, pp. 189–194. ACM, New York (1991)
8. Tanaka, Y., Okada, Y., Niijima, K.: Treecube: Visualization tool for browsing 3d multimedia data. In: *International Conference on Information Visualisation*, p. 427 (2003)
9. van Wijk, J.J., van de Wetering, H.: Cushion treemaps: Visualization of hierarchical information (1999)
10. Cockburn, A., McKenzie, B.: Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing our World, Changing Ourselves (CHI 2002)*, pp. 203–210. ACM, New York (2002)
11. Grossman, T., Balakrishnan, R.: An evaluation of depth perception on volumetric displays. In: *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2006)*, pp. 193–200. ACM, New York (2006)
12. Peterson, S.D., Axholt, M., Ellis, S.R.: Technical section: Objective and subjective assessment of stereoscopically separated labels in augmented reality. *Comput. Graph.* 33, 23–33 (2009)
13. Robertson, G.G., Mackinlay, J.D., Card, S.K.: Cone Trees: animated 3D visualizations of hierarchical information. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching Through Technology, CHI 1991*, pp. 189–194. ACM, New York (1991)
14. Deller, M., Ebert, A., Agne, S., Steffen, D.: Guiding attention in information-rich virtual environments. In: *International Association of Science and Technology for Development (IASTED)*, pp. 310–315. ACTA Press (2008)
15. Collins, C., Carpendale, S.: Carpendale s: Vislink: revealing relationships amongst visualizations. *IEEE Trans. Vis. Comput. Graph.* (2007)
16. Eades, P., Feng, Q.W.: Multilevel Visualization of Clustered Graphs. In: North, S.C. (ed.) *GD 1996 LNCS*, vol. 1190, pp. 101–112. Springer, Heidelberg (1997)
17. Brandes, U., Dwyer, T., Schreiber, F.: Visualizing related metabolic pathways in two and a half dimensions (2003)
18. Marr, D.: *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., New York (1982)
19. Dix, A., Finlay, J.E., Abowd, G.D., Beale, R.: *Human-Computer Interaction*, 3rd edn. Prentice-Hall, Inc., Upper Saddle River (2003)
20. Kreylos, O.: Vrui package (2012)
<http://idav.ucdavis.edu/~okreylos/ResDev/Vrui/index.html>

Adaptive Semantic Visualization for Bibliographic Entries

Kawa Nazemi, Reimond Retz,
Jürgen Bernard, Jörn Kohlhammer, and Dieter Fellner

Fraunhofer Institute for Computer Graphics Research (IGD),
Fraunhoferstr. 5, 64283 Darmstadt, Germany
`{kawa.nazemi,reimond.retz,juergen.bernard,
joern.kohlhammer,dieter.fellner}@igd.fraunhofer.de`

Abstract. Adaptive visualizations aim to reduce the complexity of visual representations and convey information using interactive visualizations. Although the research on adaptive visualizations grew in the last years, the existing approaches do not make use of the variety of adaptable visual variables. Further the existing approaches often premises experts, who has to model the initial visualization design. In addition, current approaches either incorporate user behavior or data types. A combination of both is not proposed to our knowledge. This paper introduces the instantiation of our previously proposed model that combines both: involving different influencing factors for and adapting various levels of visual peculiarities, on visual layout and visual presentation in a multiple visualization environment. Based on data type and users' behavior, our system adapts a set of applicable visualization types. Moreover, retinal variables of each visualization type are adapted to meet individual or canonic requirements on both, data types and users' behavior. Our system does not require an initial expert modeling.

1 Introduction

Recent research in adaptive visualizations showed significant advances in human information processing ([1], [2], [3], [4]). The adaptation techniques were in particular adopted to search and exploration tasks ([2], [1]). The evaluation results of the implemented adaptive visualizations were promising, whereas the applied methods varied enormously ([2], [3], [4]). Today's systems apply visual adaptation either on data to provide a better suited visualization type ([5], [6] or to recommend data for users based on "points-of-interest" [2]. Further some systems investigate the interaction as baseline, whereas experts model the best suited visualization for the various tasks [7]. After a review on related work and to our knowledge, none of today's systems incorporates both: adaptive visualization based on (1) data and on (2) users within the analysis process without using the expert knowledge for visualization design. Thus interactive visualizations provide a problem-solving process for various tasks, e.g. searching, locating, analyzing or exploring ([8], [9], [10]), we propose that it is more valuable to provide an adaptation on different levels of visual variables considering the human

vision perception ([11], [12], [13], [14]). An analytical task involves a more goal-directed and serial processing of vision perception [11], needs other variables to be adapted than "locating" a search result, which can be performed by providing a parallel perceived visual variable, e.g. a simple distinguishable color of one entity highlighting the from its distractors [15]. However, the combination of adapting both, type and retinal variable is not applied in today's visualization systems. Further the adaptation approaches, which involve multiple visualizations to comprehend the perspective on data, are rare. The involvement of users in the adaptation process is often performed only on individual level with limitation of experts, who has to create an initial user model. An overall improvement of the adaptation behavior, e.g. by implementing a canonical user model instead of using initial expert knowledge, is not considered in the existing approaches. Rather the individual user preferences are investigated without improving the general adaptation procedure. In this paper we introduce a novel adaptive visualization system based on our previously proposed model ([13], [14]). Our visual adaptive system focuses on a suggestion and automatic adaptation of both: best suited visualization types (Layout) and their visual representation (Presentation) ([13], [14]). Additionally, the user behavior is investigated in a "canonical user model" for providing a "learning" adaptive approach that considers the most common user beside only focusing the individual user behavior. We demonstrate our adaptive visualization system on real-world BibTeX entries of the Eurographics Association (EG) Digital Library. The application example is geared towards the field of Digital Libraries.

2 Related Work

2.1 Adaptive Visualizations

The term *adaptive visualization* is used for different levels of adapting the visual representation, filtering and recommending data to be visualized. Golemati et al. [16] introduced a context-based adaptive visualization that concerns user profiles, system configuration and the document collection (data set) to provide an adequate visualization. They state that the choice of 'one' adequate visualization from a pool of visualizations leads to a better performance. The adaptation of the visualization is based on the "context" which has to be generated manually. The rules for user classifications needs experts, who have to classify this aspect manually too [16]. An implicit interaction analysis is not performed; further the use of different visual variables is not investigated. A similar approach is proposed by Gotz et al. with the HARVEST tool [7]. HARVEST makes use of three main components: a reusable set of visualization widgets, a context-driven visualization recommendation and semantic-based approach for modeling user's analytical process. The result of the modeling component is used by the integrated visualization recommendation to choose one visualization for the analytical task and is limited to just one visualization. A further and essential limitation is the need of experts who have to define an initial design for the interaction patterns and the resulting visualization recommendation. [7] With the *APT tool* [5] and

the consecutive *Show Me* system [6], Mackinlay et al. differ from the previously described works in a metaphor of small multiple displays and an enhanced aspect of user experience in visual analytics. Although they propose an adaptive visual system, the used algebra is defined for data to provide a better mapping of data-tables to visual representations. Another approach for data-adaptive visual presentation is HiMap [17]. The system reduces the graph-layout complexity (visual density) by implementing an adaptive data-loading algorithm. Similarly, da Silva et al. investigated the reduction of complexity by adapting the data [18]. They introduced a formal model for computing the degree-of-interest based on the main concept of an ontological data structure. These approaches are limited to adapting the visual representation based on the underlying data; the user is not investigated as influencing factor for the adaptation.

The adaptation of a spatial visual presentation layer based on user preferences is proposed in the *Adaptive VIBE* system by Ahn and Brusilovsky ([1], [2]). However, their approach does not provide adaptive capabilities for various data types and is limited to one visualization type and a point-of-interest provided by the user manually. [1] The introduced examples demonstrate the upcoming popularity of adaptive visualization concepts. However, the majority of the systems use either one influence factor or adapt to one visualization or visual presentation, but the main limitation is further the involvement of either experts to model an initial visualization design or the active involvement of users' that may be obtrusive for their work. An automated visualization adaptation by adapting the different levels of visual representations (Data, Layout, Presentation) ([13], [14]) is not being proposed, further the advantages of reducing the complexity by combining visualizations, adapting their presentations and make use of data types and user behavior has not been thoroughly investigated so far.

2.2 Visualizations for Digital Libraries

The visualization of bibliographic entries is a broad area with many efficient and useful techniques, especially in the field of Digital Libraries. The BiblioVis system by Shen et al. [19], the Citation Map of *Web of Knowledge* [20] and a methodology based on *Power Graphs* [21] may serve as examples. Similar to our work, multiple-linked views and a variety of graph-based visualizations were applied to provide additional value of bibliographic entries to the user. However, adaptive visualization concepts are not provided in that context. Chou and Yang presented the PaperVis system [22]. They combined a modified version of Radial Space Filling and Bullseye View to arrange papers as a node-link graph. It further provided a semantically meaningful hierarchy for facilitating literature exploration [22]. Bergström and Atkinson [23] presented PaperCube, a web-based approach for visualizing the metadata of a CiteSeer version. Similar to our approach, the authors integrated different visualization types for exploring the metadata and the correlations within the data-set in a web-based environment. Furthermore, PaperCube enables the choice between *article* and *author* and provides various features for gathering detailed information. In contrary to our approach, their user interface neither provides multiple-linked views, nor

implements any adaptive behavior. [23] To provide a better access to digital libraries and find relevant information, adaptive visualization are required.

3 Approach

This chapter introduces the key technologies used for the proposed system. The main contribution of our application is to instantiate the visual adaptation model proposed in [13] and revised in [14] by combining several concepts and technologies. To face the proposed model various influencing factors, e.g. data and user are investigated to meet the adaptation of the visual representation on proposed levels of granularity. We chose the domain of digital libraries as an application example for the instantiation. We first give a brief introduction to the data integration and interpretation step of the system based on the chosen application example. Subsequently, we report on data-specific and user-dependent implications that influence the adaptive behavior of the visualization. Furthermore, we detail in how the adaptive capability of the system affects the automatic choice of visualization types (Layout) and visual variables (Presentation).

3.1 Data Processing and Light-Weight Semantics

The presented approach is capable for a variety of common data types. In the following, we describe the process of data integration and processing by means of the chosen application example on digital libraries. Our web-based visualization uses an API from the Eurographics Association (EG) for accessing the EG-BibTeX entries. We use simple HTTP-request to search in the EG-Library for results in *title*, *keywords* and *authors*. The result of the query is plain-text with BibTeX entry-results. We apply processing routines to improve the data quality. For a uniformed letter-comparison, a rule-based routine is applied that transforms all characters into the standard Latin character-set. Since a query on the EG database is applied on the three categories *title*, *keywords* and *authors*, the result may contain duplicates. We remove these redundant result entries by a duplicate detection routine. In a preliminary test case, we identified the need to disambiguate the author's names. To overcome this problem, we apply a rule-based algorithm that compares the last name, the first name, the first character of the first name, the coauthors and the ACM CSS. Figure 1 illustrates the system's disambiguation of the name 'Fellner' in the EG-Library. As a next step, we create a semantic schema based on the respective BibTeX metadata classes relation extraction. For example, we formalize paper authors as 'co-authors of' each other provide the relation 'written by' to papers and their corresponding authors and assign the relation 'author of', in return. This simple but efficient schema subsequently enables the adjustment of visualizations between different metadata attributes. The obtained network of metadata attribute relations is presented by graph-based visualization types. The ACM CSS data enables the creation of a lightweight hierarchy of publications. This additional information, encoded as a hierarchical data structure, is visually presented in the system

with hierarchical aggregation metaphors. Since the data processing is executed in real-time, additional persistence layers are not necessary. Rather, the system is capable of immediate changes in the EG-library.

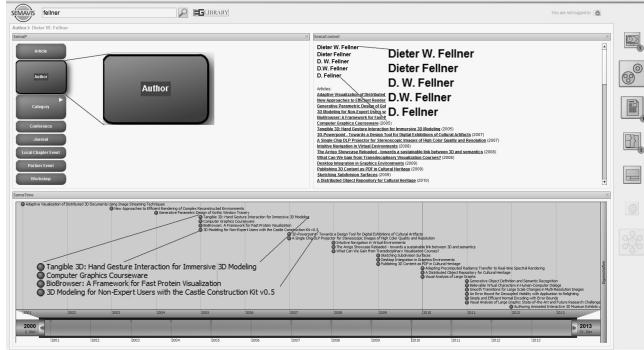


Fig. 1. Illustration of the search result for the term "fellner" in a multiple-visualization cockpit generated automatically with adaptation techniques

3.2 Adaptation of Visualizations

Our application integrates a set of seven visualization-algorithms that are responsible for the placement and arrangement of the objects on the screen. We separate the visual presentation and spatial arrangement (Layout) of the objects, based on the research outcomes of vision perception ([12], [11]), to provide a more efficient adaptability. The visualization-algorithms can be combined in a visualization cockpit with an enhanced brushing and linking metaphor [24]. We further enhance this approach by automating the selection of appropriate visualization-algorithms based on the search result. Therefore each visualization-algorithm is annotated with its visualizing capabilities. Graph-based algorithms visualize the relationships between different and within categories (authors, articles, conferences etc.) (see Figure 2 right visualization), a time-based visualization illustrates the temporal spread of the results (see Figure 2 bottom), a Treemap-similar visualization [25] provides the ability to browse within the categories (see Figure 2 left) and lists with textual information provide the content of the found results (see Figure 1 right visualization).

The capability of each visualization algorithm is one indicator to recommend and automate the selection of the most appropriate visualization algorithm. Another indicator is the users' interaction with visualizations. We enable the users to place visualizations into the user interface or to remove them and enhanced the interaction analysis and prediction algorithm proposed in [26] to investigate the users' choice of combined visualizations. The user interactions on visualizations placed on the screen and the choice of alternative visualizations or their movement from the screen are used to derive a canonic user model. [27] Our canonic user model, models the behavior of all users by analyzing the interactions with

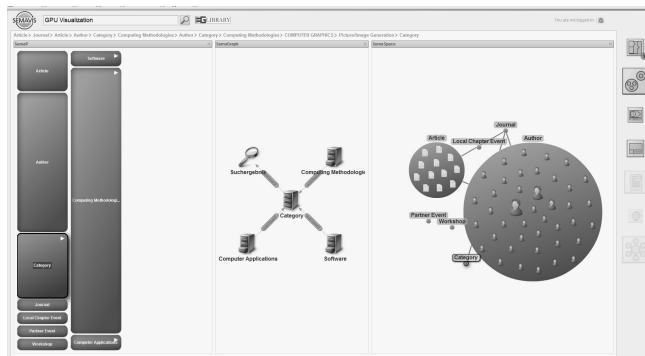


Fig. 2. Illustration of the search result of the term "GPU visualization" in a multiple-visualization cockpit generated automatically the with adaptation techniques

system based on the KO*/19 algorithm [26]. Therefore users' interactions are transformed in a numerical, internal representation and the Steady State Vector is determined as a relative measurement for the occurrence of interactions. [26] The model involves the interaction quantity with each data element, visualization element and the choice of visualizations to enable a learning system that considers the behavior of the majority of users. Further it provides general usage information of the visualizations to enable the recommendation and automatic selection of visualization-algorithms. The canonic user model does not require personal information about the user because the model itself provides a general data-dependent "initialization". To overcome an over-generalization of visualization choice, the canonic user model is counterbalanced with an additional user grouping, based on individual interactivity preferences and behavior. Thus, the system provides the capability to respond to individual users. For this, we implemented an algorithm that computes the deviation of the user interaction behavior. Therefore the user-interaction behavior of the current user is compared with the canonic user model with respect to the number of users. This enables to estimate, if the same or a similar user is interacting with the system and can be enhanced to group the users with diverting intentions. The approach provides two different modi for user-oriented adaptation. First the canonic-user model that investigates the behavior of all users and second an individualized user model. The individual user model is an instantiation of the canonic user model with certain preferences and interaction history of a certain user. If a user is interested in getting behavior-based visual adaptation, he is able to log-in as individual user. The default user model in our approach is the canonical user model. It is activated, if an individual user is not logged-in. The automatic selection and recommendation of the visualization-algorithms is one adaptation characteristic of our application. In addition the visualization layout is decoupled from the visual representation ([13], [14]). We define *visual presentation* as the sum of those visual or retinal variables, which can be perceived by human in parallel [15], e.g. color, shape, texture, size etc. of nodes, objects etc.

([13], [14]). Our approach uses the visual presentation for quantitative information of the underlying results or for specific user preferences on content. The number of results is used as an indicator for adapting the visual presentation. For example, the system highlights the authors with the most publications. If the individual user model is activated by the user, the visual presentation can be used for recommending content. Therefore, the history of her interactions is considered with a subsumption of the hierarchy of the schema she interacted with. For example if the user is more interested in user interfaces and visualizations (based on his previous interactions) and searches for the author *fellner*, our application presents the ‘categories-of-interest’ visually highlighted (see Figure 3 top illustration). In contrary the canonical user model applies the number of results as indicator for the visual variables (see Figure 3 bottom illustration). Visualizations that are not applicable for currently analyzed data types are temporarily excluded from the set of user-selectable visualization types. If the data type changes within the exploration work flow, the system automatically adapts the set of provided visualization types. For example, the user may request all publications of a specific author on demand. In this case the system automatically adapts the visualization for the specific results during the interaction with a visualization transformation. Changes in provided visualizations are performed as unobtrusively as possible in order to not confuse the user. This is achieved by automatically suggesting the most similar visualization type (e.g. aim to apply a graph-based visualization when replacing another graph-based visualization). Although the transformation phases between two visualizations have not been considered as irritating by the test users in the development phase, we aim to conduct a formal evaluation as future work to measure the obtrusiveness of a change while a user interacts with the system.

With the described approach we introduced a way how visualization can be adapted to various influencing factors, e.g. data and users. An initial design for the visual adaptation is not required anymore, thus the canonical user model provides a self-learning approach and improves the visualizations with each user. Further the adaptation output was enhanced by separating the visual variables into Layout and Presentation. [14] This enables a fine-granular adaptation of visualizations.

4 Application Scenario

In the previous section, an abstract overview of the application was presented, comprising the real-time access to a web API, a system design combining multiple types of adaptive visualizations and a user-centered recommendation system for various visualization types. We now demonstrate the applicability of the system and the system behavior. The starting point of the application is a blank user interface with a search field at the upper left, a log-in button at the upper right and icons for visualization recommendation at the right (Figure 3 right vertical bar). A search space definition (like the authors category) is not needed, thus the system recognizes automatically the search space. In Figure 4, the user started

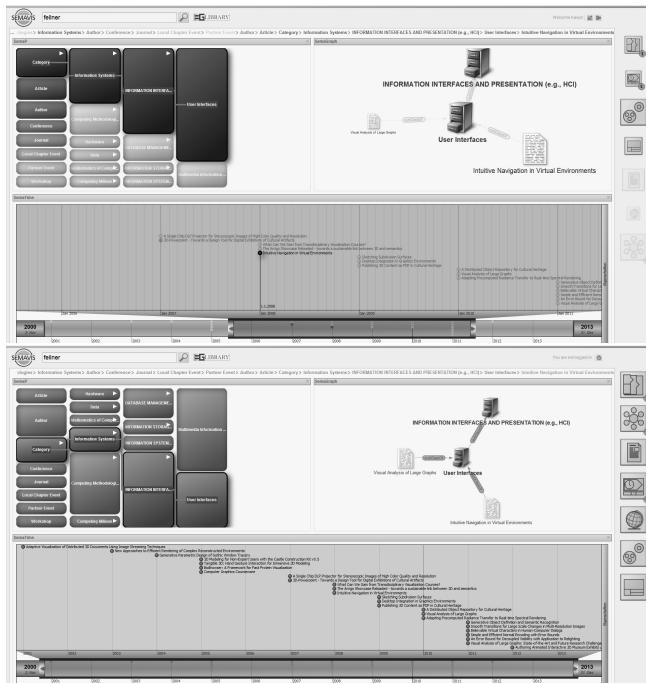


Fig. 3. Presentation Adaptation: the first visualization (top) illustrates the visual adaptation based on the behavior of an individual user, recommending relevant content; the second visualization (bottom) uses the canonic user model and adapts the visual presentation on the quantity of results.

the search with the textual query for ‘GPU visualization’. Here, the adaptive application provides two visualizations and highlights the category *article*, thus the majority of the results was found in articles. In the first very simple view he recognizes that there are a couple of articles found, the articles have relations to workshops, journals, etc. and there are more entities found in authors (as authors of the articles) than articles. If the user clicks on a category, a third visualization is added by the system to provide the navigation through the categories and the related articles. She is now able to put new visualization into the user interface, e.g. for investigating the temporal spread of the articles over time or to navigate through the results and explore the results. Figure 2 illustrates this case combined with a click on authors. The authors are represented by icons with different sizes, indicating the amount of articles related to the search. The user is now able to explore the results with the given visualization or put new visualizations into the user interface. Visualizations that are not able to illustrate the current set of chosen data are blocked and cannot be chosen. While navigating through the visualizations it may occur that new visual representations appear or the visual representation changes, e.g. by clicking on article, textual information appears on the screen. If the user searches for the term ‘fellner’, he gets another view on

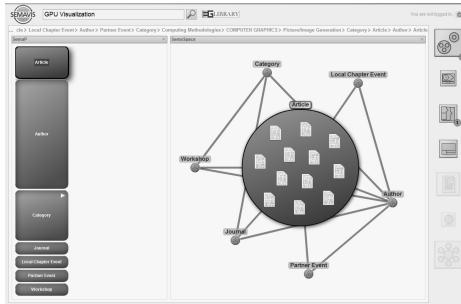


Fig. 4. Visualization of the results on the search "GPU visualization"

the same BibTeX data (Figure 1): First, the Treemap-similar visualization highlights the category ‘author’ and provides an overview navigation on the results, second a textual representation of the various (disambiguated) writings of the author’s name, the related articles and coauthors and a temporal view on his publications. If an author cannot be disambiguated, an additional visualization provides the possibility to refine the search results.

5 Preliminary User Study

We conducted a preliminary user study to evaluate the canonic user model and the adaptive behavior on the presentation level. A between-subjects-design was chosen involving 14 participants (71.43% female, average age 24.5 years). The participants were randomly distributed over two experimental conditions. A Linked-Open Data web source was chosen. The applied adaptive concept was based on the canonic user model and adapted the size, color and order of the entities. Both of the groups of participants were requested to solve identical search tasks. The independent variable of the study setup was the visual adaption capability. One of the two groups was tested with adaptive visual presentation while the other group ran the tasks without visual adaptation. As our hypothesis we expected that the users of visualizations with visual adaptation are performing more tasks in the same time. The goal of the study was to determine whether the visual adaptation of the presentation leads to a better performance with respect to search tasks and higher user acceptance of the visualizations. In order to measure the performance during the tasks the participants behavior was logged (task-completion-time, derivation from optimal path etc.). For measuring the acceptance, the INTUI questionnaire [28] was used, which measures the intuitiveness of applications. In particular, the questionnaire items for measuring the *Magical Experience* and Gut Feeling were important, thus these correspond to positive emotions. [29] A between-subjects multivariate analysis of variance was performed on six dependent variables: the number of correctly answered questions and the five sub-scales of the INTUI questionnaire. Also kolmogorov-smirnov tests (KS-tests) were performed to ensure normality with an inference statistical procedure. The effects of the condition are shown in Figure 5.

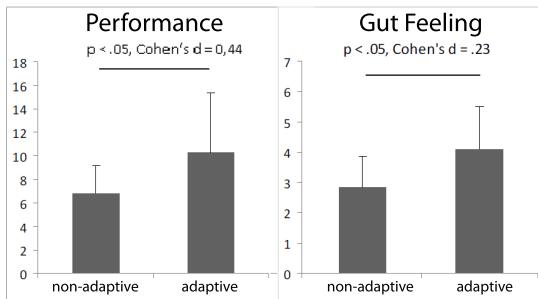


Fig. 5. Evaluation results on performance and *Gut Feeling*

Our hypothesis was confirmed by the data. The participants using the adaptive variant were able to answer more questions correctly than the participants using the visualization without adaptive behavior, $F = -4.86$, $p = .0385$. This result was expected, hence a one-sided test for significance was performed. Also participants in the condition adaptive visualization scored significantly higher on *Gut Feeling* ($F = 1.29$, $p = 0.365$), as expected, they perceived their actions more guided by gut feeling. The variables *Magical Experience*, *Verbalizability* and *Effortlessness* showed no significant differences between the conditions.

6 Conclusion

This paper introduced the instantiation of an adaptive reference model ([13], [14]) as an adaptive visualization application for bibliographic entries in digital libraries. We used a couple of existing visualization techniques, adaptation and user interaction algorithms and a technique of generating light-weight semantics to provide a novel approach for visual adaptation. The main contribution was the separation of the visual layer into layout and presentation for a fine-granular adaptation of visualizations. The integration of user models on individual and canonic level in combination with the data-attributes provided a user- and data adaptive behavior without the need of an expert to design the visual layer. The approach was exemplary adopted to bibliographic entries in a digital library. We proposed that the combination of adaptive visualizations with the generation of light-weight semantics can support users in their search and exploration tasks and improve the user experience. Even though our evaluation was preliminary, the case study led to very promising results.

Acknowledgments. We gratefully thank Stefanie Behnke from the Eurographics Association for providing us the API to the Eurographics Digital Library and for her fruitful comments. This work has been carried in the FUPOL project, partially funded by the European Commission under the grant agreement no. 287119 of the 7th Framework Programme. This work is part of the SemaVis

technology, developed by the Fraunhofer IGD (<http://www.semavis.net>). SemaVis provides a comprehensive and modular approach for visualizing heterogeneous data for various users.

References

1. Ahn, J.W., Brusilovsky, P.: Adaptive visualization of search results: bringing user models to visual analytics. *Information Visualization*, 167–179 (2009)
2. Ahn, J.W.: Adaptive Visualization for Focused Personalized Information Retrieval. PhD thesis, School of Information Sciences, University of Pittsburgh (2010)
3. Toker, D., Conati, C., Carenini, G., Haraty, M.: Towards adaptive information visualization: On the influence of user characteristics. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (eds.) UMAP 2012. LNCS, vol. 7379, pp. 274–285. Springer, Heidelberg (2012)
4. Steichen, B., Carenini, G., Conati, C.: User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In: Proc. of IUI, IUI 2013, pp. 317–328. ACM, New York (2013)
5. Mackinlay, J.: Automating the design of graphical presentations of relational information. *ACM Trans. Graph.* 5, 110–141 (1986)
6. Mackinlay, J., Hanrahan, P., Stolte, C.: Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 1137–1144 (2007)
7. Gotz, D., et al.: HARVEST: an intelligent visual analytic tool for the masses. In: Proceedings of the First International Workshop on Intelligent Visual Interfaces for Text Analysis, IVITA 2010, pp. 1–4. ACM, New York (2010)
8. Zhou, M.X., Feiner, S.K.: Visual task characterization for automated visual discourse synthesis. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1998, pp. 392–399. ACM Press/Addison-Wesley Publishing Co., New York (1998)
9. Yi, J.S., Ah Kang, Y., Stasko, J., Jacko, J.: Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics* 13, 1224–1231 (2007)
10. Pike, W.A., Stasko, J., Chang, R., O'Connell, T.A.: The science of interaction. *Information Visualization* 8, 263–274 (2009)
11. Ware, C.: *Information Visualization Perception for Design*. Morgan Kaufmann (Elsevier) (2013)
12. Rensink, R.A.: Change detection. *Annual Review of Psychology*, 245–277 (2002)
13. Nazemi, K., Stab, C., Kuijper, A.: A reference model for adaptive visualization systems. In: Jacko, J.A. (ed.) *Human-Computer Interaction, Part I*, HCII 2011. LNCS, vol. 6761, pp. 480–489. Springer, Heidelberg (2011)
14. Nazemi, K., Kohlhammer, J.: Visual variables in adaptive visualizations. In: 1st International Workshop on User-Adaptive Visualization, WUAV 2013. Extended Proceedings of UMAP 2013. CEUR Workshop Proceedings, vol. 997, pp. 1613–1673 (2013) ISSN 1613-0073
15. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. *Cognitive Psychology* 12(1), 97–136 (1980)
16. Golemati, M., Halatsis, C., Vassilakis, C., Katifori, A., Peloponnesse, U.O.: A Context-Based Adaptive Visualization Environment. In: Proceedings of the Conference on Information Visualization, IV 2006, pp. 62–67. IEEE Computer Society, Washington, DC (2006)

17. Shi, L., Cao, N., Liu, S., Qian, W., Tan, L., Wang, G., Sun, J., Lin, C.Y.: Himap: Adaptive visualization of large-scale online social networks. In: Proceedings of the 2009 IEEE Pacific Visualization Symposium, PACIFICVIS 2009, pp. 41–48. IEEE Computer Society, Washington, DC (2009)
18. da Silva, I., Santucci, G., del Sasso Freitas, C.: Ontology Visualization: One Size Does Not Fit All. In: Matkovic, K., Santucci, G. (eds.) Proceedings of EuroVA 2012: International Workshop on Visual Analytics, Eurographics Association, pp. 91–95, Vienna (2012)
19. Shen, Z., Ogawa, M., Teoh, S.T., Ma, K.L.: Bibliviz: a system for visualizing bibliography information. In: Proceedings of the 2006 Asia-Pacific Symposium on Information Visualisation, APVis 2006, vol. 60, pp. 93–102. Australian Computer Society, Inc., Darlinghurst (2006)
20. Matthews, T.: Citation Map Visualizing Citation Data in the Web of Science. Thomson Reuters (2010)
21. Tsatsaronis, G., Varlamis, I., Torge, S., Reimann, M., Nørvåg, K., Schroeder, M., Zschunke, M.: How to become a group leader? or modeling author types based on graph mining. In: Gradmann, S., Borri, F., Meghini, C., Schuldt, H. (eds.) TPDL 2011. LNCS, vol. 6966, pp. 15–26. Springer, Heidelberg (2011)
22. Chou, J.K., Yang, C.K.: PaperVis: Literature Review Made Easy. Computer Graphics Forum 30, 721–730 (2011)
23. Bergstrom, P., Atkinson, D.: Augmenting the exploration of digital libraries with web-based visualizations. In: Fourth International Conference on Digital Information Management, ICDIM 2009, pp. 1–7 (2009)
24. Nazemi, K., Breyer, M., Burkhardt, D., Fellner, D.W.: Visualization Cockpit: Orchestration of Multiple Visualizations for Knowledge-Exploration. International Journal of Advanced Corporate Learning 3, 26–34 (2010)
25. Nazemi, K., Breyer, M., Hornung, C.: SeMap: A Concept for the Visualization of Semantics as Maps. In: Stephanidis, C. (ed.) UAHCI 2009, Part III. LNCS, vol. 5616, pp. 83–91. Springer, Heidelberg (2009)
26. Nazemi, K., Stab, C., Fellner, D.W.: Interaction Analysis for Adaptive User Interfaces. In: Huang, D.-S., Zhao, Z., Bevilacqua, V., Figueroa, J.C. (eds.) ICIC 2010. LNCS, vol. 6215, pp. 362–371. Springer, Heidelberg (2010)
27. Seeberg, C.: Life Long Learning; Modulare Wissensbasen für elektronische Lernumgebungen (Modular Knowledge-bases for Electronical Learning Environments). Springer, Heidelberg (2003)
28. Ullrich, D., Diefenbach, S.: Intui. exploring the facets of intuitive interaction. In: Mensch & Computer 2010: 10. Fachübergreifende Konferenz für Interaktive und Kooperative Medien. Interaktive Kulturen, p. 251. Oldenbourg Wissenschaftsverlag (2010)
29. Ullrich, D., Diefenbach, S.: From magical experience to effortlessness: an exploration of the components of intuitive interaction. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, pp. 801–804. ACM (2010)

A Methodology for Interactive Spatial Visualization of Automotive Function Architectures for Development and Maintenance

Moritz Cohrs¹, Stefan Klimke¹, and Gabriel Zachmann²

¹ Volkswagen AG

² University of Bremen

Abstract. In this paper, the authors utilize spatial visualization of automotive function architectures to enable novel, improved methodologies and workflows for the development, validation and service of vehicle functions. The authors build upon a prior approach for consistent data integration of automotive function architectures with CAD models. They show the benefits of the proposed novel methodologies by applying them to the scenario of developing an automotive signal light system. This demonstrates the capabilities of the new methodology in making a function-oriented development much more efficient as well as supporting testing and service.

1 Introduction

Today, the automotive industry has to face the challenges of an ever increasing complexity in development, validation and service. Two main causes can be found in an increasing diversity of variants and car configurations as well as in an increasing quantity of vehicle electronics and vehicle functions. Such functions like *Park Assist*, *Dynamic Light Assist*, or *Start-Stop Automatic*, are implemented as mechatronic systems, consisting of many different components like sensors, actuators and controllers. Moreover, many automotive functions involve a considerable amount of signals and information being communicated across multiple components and networks.

In automotive development today, function architecture diagrams are used for the design and validation of automotive functions and systems. However, such diagrams do not provide any information on the spatial location and distribution of function-related components and wires within actual vehicle configurations. Especially in modern, highly networked vehicles, it is rather difficult for developing engineers, function testers and service technicians to associate function-related facts, requirements and issues to particular components and wires in actual vehicle assemblies. Moreover, a mechatronic system implementing a vehicle function can be considerably different across particular variants and configurations, even in same vehicle models.

In our work, we address the above issue by proposing and investigating a novel methodology enabling an interactive, spatial visualization of automotive function architectures. We build upon prior work in which we have proposed an approach for consistent data integration of automotive function architectures with CAD models. In this paper, we apply this approach to an automotive turn signal light (TSL) system in

order to explore that our interactive visualization can be of assistance in the early, function-oriented development as well as in the service sector.

2 Related Work

In the automotive industry, the increasing complexity and quantity of automotive systems, networks and related in-car communications present considerable challenges to many related domains like development, testing, and maintenance. Recent approaches show that one promising approach in mastering those challenges is to focus on methods of visualization to enhance quality and transparency of complex product data in related processes.

In the manufacturing industries, a typical and established application of virtual prototyping is a digital mock-up (DMU), which facilitates the utilization of CAD models for geometric investigations, such as assembly analysis or collision detection. For example, [12] show how challenges of heterogeneous, collaborative CAD assembly can be handled by DMU approaches. Moreover, approaches like [4], [7] and [8] focus on streamlining interfaces between CAD data and immersive virtual reality environments to enable high-quality rendering and improve immersive design reviews. However, such approaches focus on geometric analyses and do not incorporate function-oriented data.

A functional (digital) mock-up (FMU/FDMU) enhances traditional DMU by integrating numerical simulation models, like those created with MATLAB/SIMULINK, with CAD data to enable visual, functional simulation of product properties [3], [5]. For instance, an FMU framework has been proposed by [9] which helps to shorten development times of multi-domain systems and which allows integration tests at early stages of development. [6] use a wireless, real-time transmission to transfer simulation data to a rich 3D environment creating a comprehensible visualization of such data. Thus, their work assists in validation and presentation of simulation data, especially for non-experts.

In support of mastering the challenges of complex automotive systems, [11] provide a dual-view visualization for exploring functional dependency chains of in-car communication processes. One view focuses on hardware component dependencies using a space filling approach while the second view uses an interactive sequence chart to displays functional correlations. In addition, [10] have proposed a visual tool for exploring and communicating an automotive bus technology to support automotive engineers in the development of car communication networks. As a result, they found beneficial application in utilizing new methods for information visualization in a complex domain, in which the only access to data was textual so far. In addition, [13] developed a system for visualizing spatial sensor data to assist in the development of automotive driver-assistance systems based on environmental perception.

Summarizing, recent related work indicates the potentials of novel approaches of data and information visualization to assist in virtual prototyping. In many cases, such approaches are based on cross-system solutions and interdisciplinary interfaces. However, there is still a lack in tracing generic function architectures in actual vehicle assemblies and configurations to assist developing engineers and service technicians. Therefore, our approach contributes in these fields by utilizing a synthesis of automotive function architectures with CAD/DMU data, exploiting synergy potentials in order to enable a new methodology for spatial visualization and analysis of function-oriented data.

3 Consistent Data Integration of Automotive Function Architectures with CAD Models

A promising solution to the recent challenges of automotive complexity is the relatively new *function-oriented* development approach that addresses the interdisciplinary development of vehicle functions and which helps to handle the high complexity in automotive development. At this stage, however, a function-oriented development does not fully exploit the capabilities of *virtual technologies*, which are fairly well-established, computer-based methods for the processing of virtual product prototypes. For example, function architectures are not yet consistently integrated with CAD models. Therefore, in prior work, we have proposed an approach for consistent data integration of automotive function architectures with CAD models (see Fig. 1) to exploit potentials of virtual technologies for a function-oriented development and to enable new, beneficial methods for a spatial visualization and utilization of such data [2]. This approach provides a system-independent XML description of function architectures to enable an integration of such function-oriented data with CAD models. For the CAD data, we exploit the PLM XML data format because it enables an integration of custom metadata with CAD data in established DMU and visualization systems. In this paper, we investigate how the novel methods that are enabled by this data integration approach can be beneficially applied in the fields of function-oriented development and service. All screenshots of the visualized data (Fig. 3-7) are captured with our prototypical implementation.

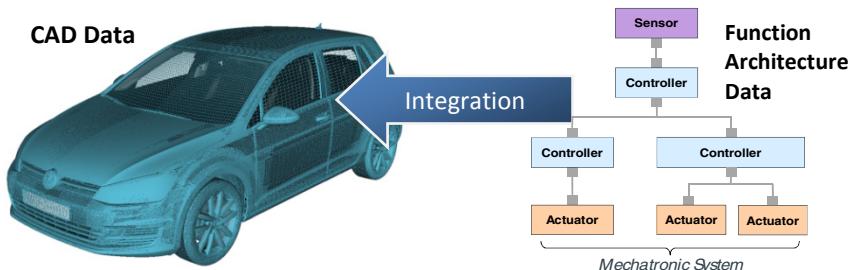


Fig. 1. We build upon our prior approach [2] of consistent data integration of function architectures with CAD models to create synergies and to enable new beneficial methods for an interactive spatial visualization and utilization of function-oriented data

Novel methods enabled by the proposed integration approach include:

- Highlighting of components and connections related to particular vehicle functions.
 - Highlighting of components and connections considering specific attributes, values and other metadata related to vehicle functions.
 - Acquisition of related function-oriented information for a given geometric part.

4 The Direction Indicator Function

In this paper, we focus on an automotive *left direction indicator* function to provide a representative and understandable use case that involves classic, discrete wires as well as network-based connections. This function is part of a turn signal light system, which is also used for different other tasks, including but not limited to indication of emergency situations, anti-theft alarm and the central-locking system. The proposed example of the direction indicator is a comparatively simple function, yet providing many beneficial use cases for our methodology. Fig. 2 illustrates a slightly simplified version of a corresponding function architecture diagram.

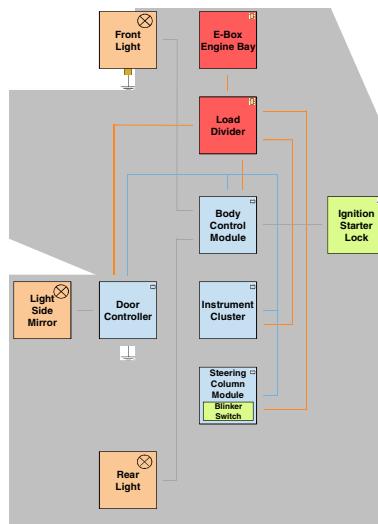


Fig. 2. The function architecture diagram of an automotive, left direction indicator function. In our approach, we aim for enabling a spatial visualization of such function architectures in actual CAD-based vehicle configurations

The above diagram shows an architecture involving different types of components and connections, including *controllers* (blue), *sensors* (green), *actuators* (orange) and *power supply units* (red). The function is triggered by a blinker switch attached to the steering column module (SCM) and the signal is communicated over a CAN bus to other controllers forwarding the signal to the particular light bulbs and control lamps in the instrument cluster. All related controllers need to be supplied with electric energy and, as well as most actuators, need a direct grounding connection for return of the electric current.

Function architecture diagrams are used in many stages of the automotive product life cycle, like in the development, validation and maintenance of vehicle functions. However, such diagrams do not provide information on the spatial distribution of function components and wires in actual car assemblies respectively in particular configurations. To fill this gap, based on our approach of consistent data integration,

we have implemented an interactive, function-oriented visualization of the left direction indicator function in an established DMU visualization system. In Fig. 3, we visualized the complete electrical system in light blue, while all wires and components related to the direction indicator function are highlighted in dark blue. In the following section, we explore the benefits of the derived methodology for a function-oriented development and service.

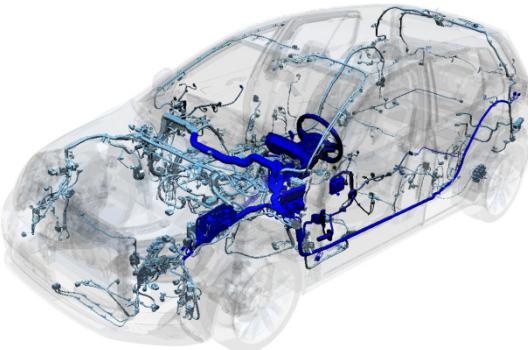


Fig. 3. A spatial visualization highlighting all function-related wires of the direction indicator function in a prototypical vehicle configuration

In modern cars, the amount of different vehicle functions can easily exceed a hundred different functions, involving many areas like safety, drivers' assistance, comfort, entertainment, etc. For example, Volkswagen differentiates more than thirty functions just for lighting applications, ranging from *Instrument Lighting*, *Parking Light* and *Headlamp Flasher* to *Dynamic Light Assist*. While most of these functions are described with architecture diagrams in a generic, vehicle-independent way, the actual implementation of a vehicle function can be different, not only across different vehicle models, but even across particular derivatives and configurations of the same vehicle model, resulting in a considerable complexity for both, development and service. The chosen example of the left direction indicator function involves a relatively clear and comprehensible architectural arrangement. However, many functions are of significantly higher complexity so that there is an increasing need for novel methodologies to enable mastery of this complexity.

5 A Novel Methodology for Automotive Function-Oriented Development

Automotive developers and service technicians have to face the challenges of an increasing number of vehicle functions implemented in complex, distributed networks. In support of mastering these challenges, in this section, we demonstrate that our visualization methodology can be of beneficial assistance, complementing usual architecture diagrams and increases transparency and quality in the development and testing of automotive functions.

5.1 Applications of Our Methodology in Testing, Maintenance and Service

In both, the development and validation of vehicle functions, as well as in repair and maintenance jobs, it is a common task to trace error causes based upon given failure symptoms. Such failures can be caused by line break (i.e. due to crash, fatigue, wear, etc.) and/or short-circuit (mass or 12V). In this use case, we assume an issue in the **power supply** to the *Steering Column Module* which is a generic cause of error with a relatively high probability. In most current cars, power supply units are critical components as they provide all systems with the necessary electrical energy. In particular, the electrical energy is created by a generator in the engine bay, temporally saved in the battery, and then distributed to the electronic components via energy dividers and/or fuse boxes.

This use case provides a fairly simple example of how our methods can assist testers and service technicians in the tracking of function-related wires. The architecture diagram in Fig. 4 provides information about the involved components and connections. Our visualization complements this diagram by providing information about the actual location and distribution of these components and connections in a real car configuration. The route of the power supply wire is clearly visible, starting from the generator/battery (E-Box) over the load divider to the steering column module, as it also could be approximately expected by the assembly locations of the involved components. In this example, the topology of the diagram is comparatively closely related to the actual assembly topology which is not necessarily true in practice.

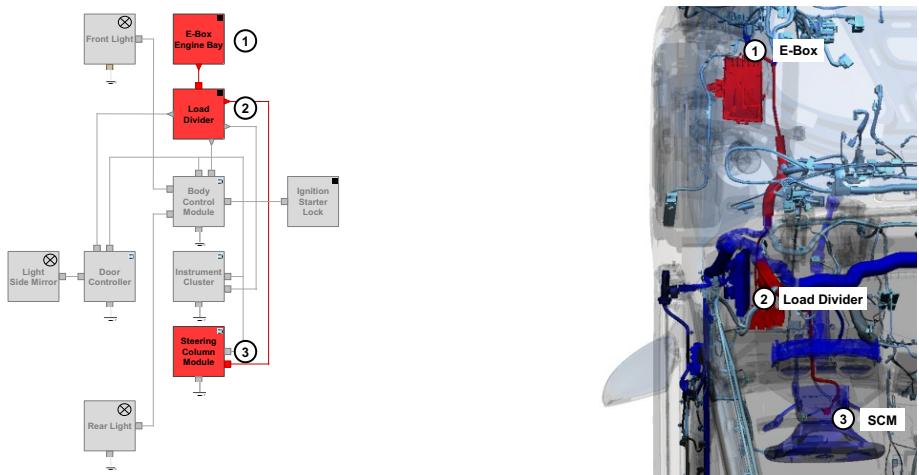


Fig. 4. On the left, all components related to the power supply of the steering column module are highlighted in the architecture diagram. On the right, we utilize a spatial visualization that highlights these components and wires in an actual car assembly

Another potential cause of error can be an issue with the **grounding connection** of the steering column module. Grounding wires connect function components to the vehicle chassis to enable a return of the electric current. This use case differs from the

previous one in that it strongly makes the limits of function architecture diagrams apparent, because, in this case, the diagram does not provide any information on the distribution of the grounding connection at all (see Fig. 5). In addition, the location of grounding bolts can be very difficult to detect in practice because they can be hidden behind caps and carpets. Moreover, dependent on vehicle platforms and decisions and needs made by the chassis fabrication, the bolts can be located in different and unexpected positions.

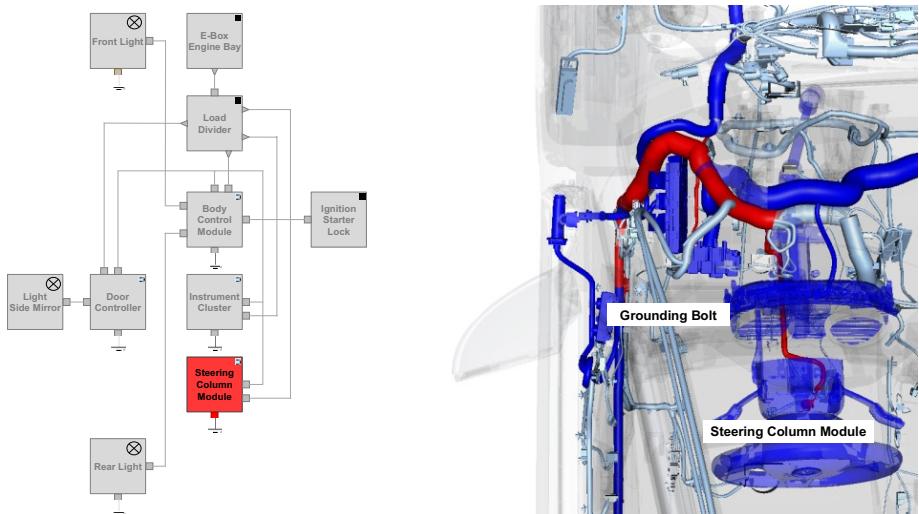


Fig. 5. Our visualization (right) provides information about the actual routing of the grounding connection which are not visible in the function architecture diagram (left)

In this example, our spatial visualization enables an efficient localization of the actual grounding connection and can beneficially assist function testers and service technicians to trace affected wires and related function-oriented information.

5.2 Applications of Our Methodology in Function-Oriented Development

The **controller area network (CAN)** is a field bus that is used for the communication between automotive systems and function components. The CAN bus is widespread in distributed embedded systems due to its electrical robustness, low price, and deterministic access delay [1]. Automotive CAN busses are frequently implemented using a star topology which has some advantages in case of failure propagation regarding particular components. For instance, if there is a communication issue due to a software error in a particular component, the other components of the bus are still able to communicate properly. However, physical failure of a component may still influence all other members of a CAN bus. Therefore, a failure of a vehicle function can still be caused by a short circuit in a system component of the bus network that is not directly related to the function at all and thus is not necessarily expected at first

sight. In this case, information about the location and distribution of the complete CAN network are necessary for further investigations. Therefore, we have extended the diagram to include all CAN members and created a corresponding visualization highlighting the full CAN network (see Fig. 6).

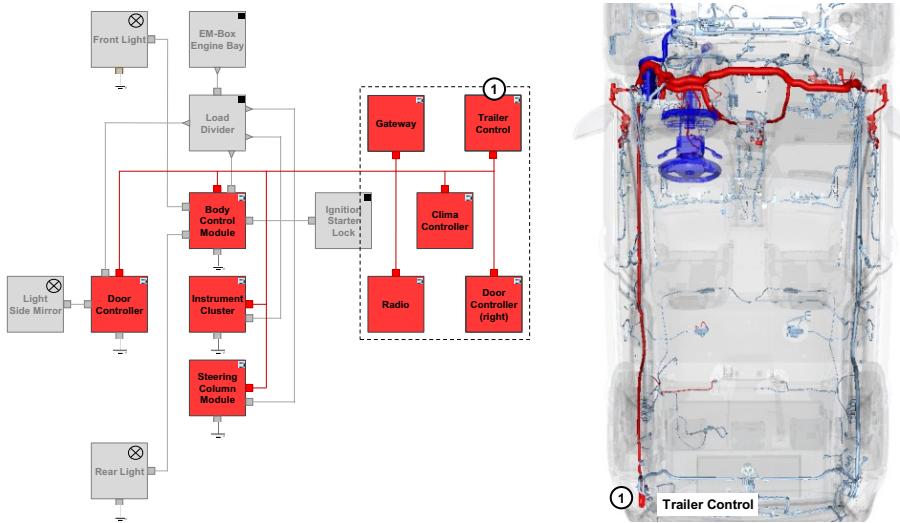


Fig. 6. We have enhanced the function architecture diagram (left) by adding all other controllers that are also connected to the involved CAN-bus network. The visualization (right) shows the complete CAN network distribution and enables statements on summed wire lengths.

While this use case can also be of benefit for service applications, at this point, we shift our focus to applications in development. An issue in a function-oriented development is that system designers of network architectures are usually not aware of the distribution of the designed networks in actual car configurations, at least not at early stages of development. Moreover, we recapture that the topology of architecture diagrams usually does not represent the actual topology of components in a car assembly because it is primarily designed after functional preferences and generically designed to be valid for many different vehicle derivatives. For example, the *Trailer Control* in the diagram in Fig. 6 is located in the upper right. As revealed by the visualization on the right, however, this controller is actually located in the left rear corner of the vehicle assembly.

Our visualization provides many benefits for the development of function architecture networks. For instance, it enables statements on the lengths of network wires and their distribution. In automotive development, for physical reasons, wire lengths in networks are limited to avoid errors in synchronicity due to long runtimes and uncontrolled communication caused by reflections and parallelism of signals. Our visualization enables developing engineers to incorporate information about the summed length and distribution into the architecture design so that potential errors can be identified and avoided at earlier steps of development. In terms of variants and configura-

tions, an approved architecture that works well in a small car does not necessarily fit in a large vehicle since wire segments can exceed the limits. Therefore, our visualization helps in applying and validating existing architectures for different vehicle configurations and variants.

In addition, our visualization enables statements on potential risks due to information access about which areas of a car may affect particular functions on crash situations. For instance, keeping on the example of the left direction indicator function, Fig. 6 illustrates the full CAN network revealing wires routed to the left rear corner including the *Trailer Control* so that it becomes visible that a rear-end collision in a parking situation can cause a short circuit in the Trailer control and thus a failure of the left direction indicator function.

5.3 Interactive Function-Oriented Data Exploration

Based upon our novel data integration approach, an engineer can now also use the geometric data or, rather, its visualization to select a particular geometric part in order to access related function-oriented architecture data. This way, he can easily obtain function-oriented information related to this geometric part, for example:

- **Functional relations:** To which function(s) does this geometric part belong to?
- **Type:** Is this geometric part a controller, sensor or actuator? Or, if it is a wire, is it a signal, grounding, etc., wire? Which other properties does it have?
- **Network relations:** If the part is a wire, is it involved in any CAN-networks or other buses?

These are just a few examples of questions that can be answered by the proposed methodology. Theoretically, any metadata that is available from the function architecture diagrams and other sources, and which has been input in the data integration, can be evaluated for specific applications. Therefore, such methods can be used as complements for digital mock-up, enhancing geometric CAD data with function-oriented information. Fig. 7 illustrates an example in which the *Body Control Module* is selected to obtain function-oriented information. A list of related functions is displayed and the *Headlamp Flasher* function is highlighted in the virtual prototype.

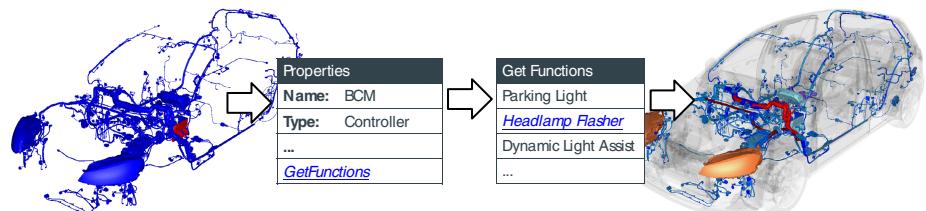


Fig. 7. Our methodology enables an interactive exploration of CAD data in order to obtain related, function-oriented information for selected geometric vehicle parts

Another use case would be in the service center. For example, a service technician might know that a particular component is defective. With our novel methodology, he

can look up this component in the visualization, select it, and thus quickly obtain related function-oriented information. As a result, he gets to know which functions may be potentially malfunctioning due to the defect component.

Finally, since our methodology is implemented in an established visualization system that is already used for conventional DMU applications, a fair user experience is ensured since engineers are able to use a well-known tool. In addition, our methodology supports all usual operations that are available in regular DMU, including object manipulation like zoom, rotation and grouping. Moreover, different filters can be used to highlight particular geometric parts in respect to different areas of interest respectively depending on specific attribute values.

6 Conclusions

In this paper, we have explored beneficial applications of interactive, spatial visualization for the development, validation and service of vehicle functions. Our methodology enables function testers and service technicians to easily locate and trace function-related components and connections in actual vehicle assemblies. Moreover, it enables reclusions about functions that may be potentially malfunctioning because of their relationship to defect components. In addition, our methods enable statements on network lengths and distribution at early development stages. We have shown that a function-oriented visualization is able to significantly assist in development and service processes and provides an appropriate solution in mastering challenges of increasing automotive complexity.

Our approach provides much potential for future work. For instance, the proposed methods can be enhanced by automatized interfaces and further improvement of end-user tools to advance a holistic integration into daily work processes. Further work can be in extending visualization tools with advanced, function-oriented functions like the evaluation of network lengths and other application-based reporting functionalities. In addition, our methodology can be integrated in hand-held devices for augmented reality applications so that function testers and service technicians get access to function-related information based upon the currently viewed vehicle assembly part.

References

1. Barranco, M., Proenza, J., Rodriguez-Navas, G., Almeida, L.: An active star topology for improving fault confinement in CAN networks. *IEEE Transactions on Industrial Informatics* 2(2), 78–85 (2006)
2. Cohrs, M., Klimke, S., Zachmann, G.: Streamlining Function-oriented Development by Consistent Integration of Automotive Function Architectures with CAD Models. *Computer-Aided Design and Applications*, vol. 11 (To appear in, 2013)
3. Enge-Rosenblatt, O., Schneider, P., Clauß, C., Schneider, A.: Functional Digital Mock – Coupling of Advanced Visualization and Functional Simulation for Mechatronic System Design. In: Proceedings of the ASIM-Treffen STS/GMMS März 2010, Ulm (2010)

4. Kim, S., Weissmann, D.: Middleware-based integration of multiple CAD and PDM systems into virtual reality environment. *Computer-Aided Design & Applications* 3(5), 547–556 (2006)
5. Krause, F., Franke, H., Gausemeier, J.: Innovationspotenziale in der Produktentwicklung. Carl-Hanser Verlag, München (2007)
6. Nybacka, M., Karlsson, T., Larsson, T.: Vehicle validation visualization. In: *Proceedings of Virtual Concepts* (2006)
7. Paillot, D., Merienne, F., Thivent, S.: CAD/CAE visualization in virtual environment for automotive industry. In: *Proceedings of the Workshop on Virtual Environments*, New York, pp. 315–316 (2003)
8. Schilling, A., Kim, S., Weissmann, D., Tang, Z., Choi, S.: CAD-VR geometry and meta data synchronization for design review applications. *J. Thejiang Univ. – Sci.* A7(9), 1482–1491 (2006)
9. Schneider, P., Clauß, C., Schneider, A., Stork, A., Bruder, T., Farkas, T.: Towards more insight with functional digital mockup. In: *European Automotive Simulation Conference* (2009)
10. Sedlmair, M., Bernhold, C., Herrscher, D., Boring, S., Butz, A.: Mostvis: An interactive visualization supporting automotive engineers in most catalog exploration. In: *13th International Conference on Information Visualization*, pp. 173–182 (2009)
11. Sedlmair, M., Hintermaier, W., Stocker, K., Buring, T., Butz, A.: A dual-view visualization of in-car communication processes. In: *12th International Conference on Information Visualization*, pp. 157–162 (2008)
12. Song, I., Chung, S.: Synthesis of the digital mock-up system for heterogeneous CAD assembly. *Computers in Industry* 60(5), 285–295 (2009)
13. Tonnis, M., Lindl, R., Walchshausl, L., Klinker, G.: Visualization of Spatial Sensor Data in the Context of Automotive Environment Perception Systems. In: *6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007*, pp. 1–9 (2007)

Navigation Recommendations for Exploring Hierarchical Graphs

Stefan Gladisch, Heidrun Schumann, and Christian Tominski

Institute for Computer Science, University of Rostock, Germany

Abstract. Navigation is a key interaction when analyzing graphs by means of interactive visualization. Particularly for unknown graphs, the user often faces situations where it is not entirely clear where to go next. For hierarchical graphs, the user may also ponder whether it is useful to look at the data at a higher or lower level of abstraction.

In this paper, we present a novel approach for recommending places in a hierarchical graph that are worth visiting next. A flexible definition of interestingness based on the notion of a degree of interest (DOI) allows us to recommend horizontal navigation in terms of the graph layout and also vertical navigation in terms of the level of abstraction. The actual recommendation is communicated to the user through unobtrusive visual cues that are embedded into the visual representation of the graph. A proof-of-concept implementation has been integrated into an existing graph visualization system.

1 Introduction

When exploring unknown graphs, users need to switch between overview and detail representations and they need to navigate to different parts of the graph. These tasks are typically supported by a zoomable representation of the graph, where the graph is hierarchically structured to provide different levels of abstraction [1]. The user can zoom & pan to visit different parts of the graph, and can expand or collapse nodes to adjust the level of abstraction. There are several existing systems that implement this strategy [2], [3], [4]. A big plus of these systems is that users can freely choose the part of the data they are interested in and the level of abstraction that suits their needs.

However, a problem is that users may be overwhelmed with the seemingly infinite number of possibilities for navigation. According to Spence [5], a key question for the user is: *Where should I go now?* Figure 1 illustrates this problem. Considering a current position arrived at during the exploration, the user does not know where interesting data could be located.

In this sense, navigating in an unknown graph to find interesting data is often a tedious trial-and-error procedure. This prompted us to investigate some kind of navigational guidance to interesting data. The aim of such a guidance is to facilitate the user's navigation decisions (i.e., recommend navigation to interesting targets) and to mitigate the trial-and-error character of navigation (i.e., minimize unconscious navigation through regions with uninteresting data).

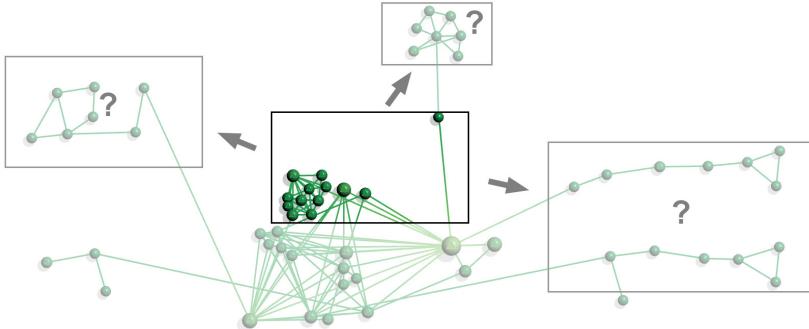


Fig. 1. The problem of navigation. Given a partial view on a graph layout (center rectangle) the user does not know where to navigate in order to find “interesting” data (rectangles with question mark).

In the following, we present a novel data-driven approach for navigation recommendations to support the exploration of hierarchical graphs. In Section 2 we describe in more detail the problem we are dealing with and briefly review existing state-of-the-art solutions. Section 3 introduces our novel approach, including means to define what the user is interested in, to compute navigation recommendations, and to communicate recommendations visually to the user. A demonstration of the proof-of-concept implementation is given in Section 4. Section 5 concludes our work and indicates directions for future work.

2 Problem Description and Related Work

Next we describe the problem addressed by our research, review existing work that is related to this problem, and identify gaps to be filled with our approach.

2.1 Problem Description

We consider hierarchical graphs as input data. A hierarchical graph is defined as a rooted tree whose leaves correspond to a graph at the finest level of granularity. Nodes and edges of the graph may be associated with data attributes. Inner nodes of the tree correspond to aggregations or abstractions of their associated child nodes [1]. We assume that a suitable layout of the graph can be computed with existing methods [6].

To allow users to explore the graph, its layout is visualized as a node-link diagram that is embedded in a zoomable space. The zoomable space enables what we call *horizontal* navigation: The user can pan to any rectangular partial view of the graph layout. A hierarchical graph allows for additional navigation on its hierarchical structure: The user can expand or collapse nodes in order to get to a lower or higher level of abstraction [7]. We call this *vertical* navigation. Figure 2 illustrates both types of navigation.

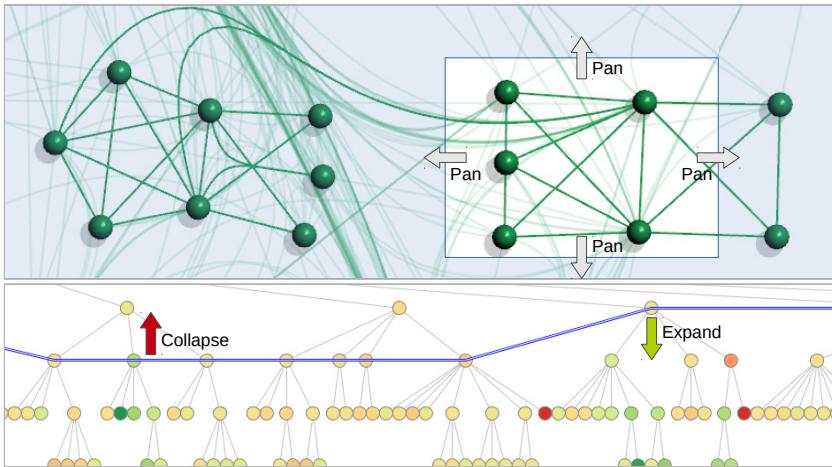


Fig. 2. Navigation in a hierarchical graph. Horizontal navigation means altering the partial view on the graph layout (e.g., by panning the view). Vertical navigation means adjusting the level of abstraction (blue line) along the graph hierarchy by expanding or collapsing individual nodes. (Colors visualize data attributes associated with nodes.)

One can easily imagine that the number of possible navigation steps is quite large. Should I pan this direction or the other to find some high-degree node? Which node should I expand to uncover a clique? Should I collapse these nodes to catch sight of the maximum-value node? In fact, the user can derive some more or less vague answers from the visual representation itself (e.g., navigate to where many edges connect). But we argue that a dedicated support to assist the user during navigation would be a promising addition to the user's analytical toolbox. With this thinking we are not alone, as documented in the next paragraphs.

2.2 Related Work

Looking at the literature one can find two categories of approaches to assist users in navigating graphs. On the one hand, there are approaches that focus on providing *orientation help* to keep users oriented. On the other hand, *navigation recommendation* approaches aim to actually suggest navigation steps to the user. In the following, we briefly review a few important examples.

Orientation help. This kind of assistance helps users to orient themselves while navigating through the data. In the context of graph exploration, May et al. [8] present a technique that computes landmarks in the vicinity (context) of the current partial view (focus). The visualization is enhanced with labeled signposts that show directions to the determined landmarks. Jusufi et al. [9] investigate orientation guidance in graphs for which a complete partition is given. The approach is based on special glyphs that provide overviews of the subgraphs connected to a focus node. Plaisant et al. [10] also use special glyphs for user

orientation. They enhance a tree visualization with preview icons that summarize the topology of subtrees. From a more general perspective, we can also consider off-screen visualization techniques (e.g., [11], [12], [13]) to be orientation help.

Navigation recommendations. The main idea of navigation recommendations is to suggest navigation steps (e.g., a specific target or a direction). Van Ham and Perer [14] describe an exploration model for graphs that includes navigation recommendations. Based on an initial focus, the approach computes and shows the most interesting contexts. Visual hints help users to decide which nodes in the context to expand in order to navigate to the additional information. Crnovrsanin et al. [15] present a technique that recommends interesting nodes based on a set of selected nodes. Interestingness of nodes depends on data attributes, graph topology, and sequences of previous user interaction. Perer and Van Ham [16] introduce *querying and browsing* as a new paradigm for graph exploration. They propose a general model that determines an initial focus and its context on the graph based on a textual query. Special icons within a node-link diagram recommend to the user where to browse the context in order to find interesting information. Additionally, the approach computes and visualizes the shortest path from a focus node to a recommended node in the context.

Open research questions. The reviewed examples from the literature demonstrate quite nicely how useful user assistance can be. A detailed look into the mechanisms behind the existing solutions reveals that most of them define a notion of a current *focus* that is associated with a *context*, where focus and context are defined exclusively on the graph structure. However, this implies that navigation recommendation can be given only for entities being connected to the focus in terms of the graph's topology. Interesting but disconnected nodes (e.g., in graphs with disconnected components) cannot be recommended, even if they are located close to the focus in the graph's layout (which is what users see on the display). Our novel solution addresses this limitation by utilizing a broader and more general notion of focus and associated context.

Another aspect common to the reviewed solutions is that they address only horizontal navigation in plain graphs. Hierarchical graphs have not been considered in connection with navigation recommendations so far. Our approach closes this gap by including vertical navigation along the axis of the level of abstraction. In other words, we address both horizontal navigation *and* vertical navigation. The next section will introduce our approach for navigation recommendations for hierarchical graphs.

3 Navigation Recommendations for Hierarchical Graphs

As described earlier, the scenario is that users explore an unknown hierarchical graph by means of a zoomable visualization that shows a layout of the graph and an encoding of associated data attributes. Our goal is to support the user in deciding which navigation step to take next to arrive at interesting data. To this end, we need to address the following key issues:

Determining Recommendation Candidates. Given a hierarchical graph and the current state of the visual exploration process we need to derive a set of recommendation *candidates*.

Selecting Interesting Recommendations. In order to compile a set of navigation *recommendations* we need to rank the candidates according to their *interestingness* and select those that are worth visiting next.

Communicating Recommendations Visually. The selected navigation recommendations need to be *communicated* to the user in an unobtrusive fashion with as little distraction from the actual visualization as possible.

Following this line of thinking, we will next describe in more detail how our approach handles these issues. But first of all, we need to define what the targets for navigation recommendations could be. In general, one could recommend navigation to any entity related to a hierarchical graph, for example, nodes, edges, connected components, cliques, or any other semantically meaningful subset of nodes and edges. For the sake of simplicity, we restrict our considerations to nodes as the targets for navigation recommendations.

3.1 Determining Recommendation Candidates

As commonly accepted, the starting point for determining candidates is the user's current focus. Based on the focus we define a context, which contains the candidates. The context must include a sufficiently large number of candidates to choose from, and it must be sufficiently small to stay focused and to avoid computations on a huge search space. The size of the context and hence the number of candidates is controlled by means of a distance measure. In summary, we use three components: (1) a set of focus nodes to start with, (2) a sufficiently sized set of context nodes – the candidates, and (3) a distance measure to control the size of the context. Figure 3 illustrates how these components can be realized.

An intuitive and often used definition of these components is based on the graph structure. A set of focus nodes is selected by the user, and the context is defined by the k -neighborhood of the focus nodes. Here k is the parameter to be adjusted to control the size of the context.

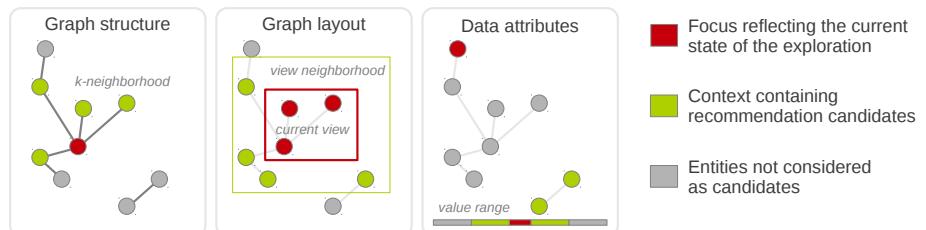


Fig. 3. Different definition of focus and context in terms of the graph structure, the graph layout, and the data attributes yield different candidates for recommendation

As already indicated, we generalize focus and context to a broader definition. So, as a second facet, we additionally consider the user's current view on the graph layout. That is, all nodes that are currently visible on the display are considered to be the focus. The context is again defined in terms of a neighborhood, but this time a neighborhood in terms of the view space. The size of the context is again controllable.

So far we have not yet taken into account the data attributes that might be associated with the nodes. Consequently, we allow for a focus in attribute space. This focus can be determined in different ways, for example by dynamic filtering sliders or by fixing the value range of what is currently visible on the display. The context can then be defined as a range of values enclosing the focus, where the range's size can be set as needed.

With this general definition we can better capture the different aspects being relevant when exploring graphs – the graph structure, the graph layout, and the data attributes. The broader definition also allows us to circumvent problems that occur when considering either of the aspects alone. An example are nodes that are close to the focus in the layout, but that are far away in terms of the graph structure. In contrast to existing solutions, our approach is able to recommend navigation to such nodes.

3.2 Selecting Interesting Recommendations

Given our definition of candidates in the context, the next step is to assign an interestingness to each candidate. As we want to recommend interesting nodes to navigate to, we need a concept that describes how interesting a recommendation candidate is. Given the unpredictability of the visual exploration process, the concept must be capable of handling varying interestingness.

An established and widely-applied concept is the *degree of interest* (*DOI*), a numerical interestingness computed by means of a *DOI* function. Originally, Furnas [17] introduced the *DOI* for trees only. Van Ham and Perer [14] generalized it for graphs. Their weighted *DOI* function considers an a priori interest of the nodes (*API*), a distance to a focus (*DIST*) and a user interest (*UI*):

$$DOI(x, F, \mathfrak{s}) = \alpha \cdot API(x) + \beta \cdot UI(x, \mathfrak{s}) + \gamma \cdot DIST(x, F)$$

where x is the node to be assigned an interestingness, F is the current focus, \mathfrak{s} are search criteria that describe what the user is currently interested in, and α, β, γ are real-valued weights. With this *DOI* definition, we already have a quite flexible mechanism to incorporate the user's interest into the recommendation computation.

Additionally, it can be important to know which data elements have already been visited (i.e., have been visible or have explicitly been marked as explored) in the course of the exploration. We propose to use an additional weighted *KNOW* component for the *DOI* function that considers the interestingness of a node according to its exploration state:

$$DOI(x, F, \mathfrak{s}) = \alpha \cdot API(x) + \beta \cdot UI(x, \mathfrak{s}) + \gamma \cdot DIST(x, F) + \delta \cdot KNOW(x)$$

By considering the exploration state of a node, we can penalize already explored data or, on the contrary, favor them. Which option to use depends on the user's goal. Visited nodes can be considered less interesting because they do not provide any new information. On the other hand, they could be particularly of interest for comparison tasks.

Given our specification of the *DOI* function, the question that remains to be answered is how to instantiate it and its components. We follow the accepted way of previous *DOI*-related approaches and provide interactive means for the user to specify and adjust the settings. To ease the specification procedure, we use template functions that can be parameterized using classic GUI elements. Which template functions to apply (i.e., what is interesting?) and how to parameterize them depends on the application domain, the use case, and the analyzed graph.

Given an appropriate *DOI* specification, we compute the interestingness of the nodes of a graph. It is worth mentioning that we do so only for the recommendation candidates in the context of the current focus. This spares us computing interestingness values for all nodes of the whole dataset.

For a hierarchical graph, we differentiate between two alternative ways of computing the interestingness. The first is that we compute interestingness for the finest level of granularity and aggregate interestingness along the hierarchy. The second alternative is to compute interestingness explicitly for each candidate irrespective of whether it is a leave node or an inner node. Again, the application scenario and the nature of the data decide on which alternative to apply.

Now that every candidate has a *DOI* value they can be sorted according to their interestingness. The result is a ranking of the recommendation candidates. Since we only want to recommend the *most interesting* nodes, we choose the first m nodes of the ranking as targets for the navigation recommendations, where m should be kept small to avoid overloading the user with too many recommendations. From our experience with test datasets, we suggest recommending $m < 10$ interesting navigation targets.

3.3 Communicating Recommendations Visually

The last step is to create an adequate visualization for the navigation recommendations. Given a potentially already visually rich graph visualization, how can we enhance it in order to communicate navigation recommendations to the users without interfering too much with the ongoing visual exploration? Our answer to this question is to embed specifically designed visual navigation cues into the existing node-link visualization. Depending on the type of navigation and on where the target of a recommendation is located, we use different visual cues. The type of navigation can be either *horizontal* or *vertical*.

Recommendation for horizontal navigation. For horizontal navigation, we distinguish navigation to nodes that are *on-screen* and nodes that are *off-screen*. Recommendations to on-screen nodes are visualized via subtle highlighting rings that encode how interesting a node is according to its *DOI* value.

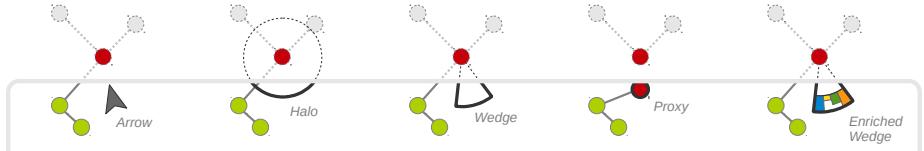


Fig. 4. Techniques for recommending navigation to off-screen nodes. Green nodes are on-screen, dashed elements are off-screen, and the red node is the recommend target.

For recommendations to off-screen nodes, we need a visual encoding that communicates at least the target’s direction and better still, the distance to the target as well. For this purpose, we consider known techniques for off-screen visualization, such as arrows, *halos* [11], *wedges* [12], or *proxies* [13]. Arrows are easy to interpret, but communicate navigation direction only. Halos and wedges have the advantage that they encode direction and distance to a recommended navigation target. Further, wedges can be arranged to reduced overlap [12]. Proxies focus not so much on target distance, but more on communicating additional information about the target by means of shape, color, or labels.

Inspired by these approaches, we designed a new solution that combines the advantages of the existing ones. What we call *enriched wedge* is a visual cue that encodes direction and distance, and also additional information about *why* the recommendation was given. This is accomplished by embedding a bar chart into a wedge. The wedge visualizes direction and distance, and each bar visualizes the partial interestingness of a recommended node according to the individual components of the *DOI* function (i.e., *API*, *UI*, *DIST*, *KNOW*). Using enriched wedges can positively influence the user in arriving at a navigation decision (i.e., choosing the “right” navigation target). Figure 4 illustrates the enriched wedge in comparison to existing off-screen techniques.

Of course, the idea of enriching the navigation recommendations with a visualization of individual *DOI* values is not restricted to wedges pointing to off-screen targets. The highlighting of on-screen targets can be enhanced in a similar manner to provide information about interestingness at a glance.

Recommendation for vertical navigation. A vertical navigation is necessary when the recommended target is not contained in the currently visualized level of abstraction. As the target is definitely not visible, we need to pick a suitable anchor to attach the navigation recommendation to. We decided to visually highlight the nodes whose expansion (or collapse) would bring the recommended target to the display. For example, if a target is below the current level of abstraction, we highlight the target’s ancestor that is contained in the current level of abstraction and whose expansion will uncover the target. If the ancestor is off-screen we can again apply one of the off-screen techniques described before.

In order to differentiate the highlighting for vertical navigation from that for horizontal navigation, and further the one for node expansion from that of node collapse, we resort to animated rings around nodes. In accordance with our goal to generate an unobtrusive visual embedding, the highlighting is designed as subtly pulsing animations with a specific direction. The animated rings appear to



Fig. 5. Snapshots of the animation that indicates nodes to be expanded to arrive at a recommended navigation target

shrink when collapse navigation is recommended and to grow for recommended expansion. Figure 5 shows snapshots of an animation indicating an expand recommendation.

3.4 Summary and Additional Concerns

With the aforementioned mechanisms, we can select interesting nodes and recommend them visually to the user as potentially worthy steps for navigation. A key issue of our approach is balancing it appropriately. Interests vary and also the visual presence of navigation cues will be perceived differently by different users in different stages of the exploration process. Therefore, it is critical to adjust the computation of recommendations and their visualization to the application scenario and to the preferences of the user. Our approach provides the required flexibility to do so. Further, we should recall the on-demand character of our approach. That is, only if users feel that they need assistance they will activate the navigation recommendations.

Two additional concerns need to be addressed in the context of navigational guidance: (1) a good initial view to start with and (2) a visual encoding of the exploration state. Both are not trivial question and we have not dealt with them in depth. Yet we give some ideas how to address them.

Ideally, a good initial view on the data provides an expressive overview of the data and offers a suitable number of options for further exploration. For determining such an initial view, different criteria can matter. For example, the number of nodes can be considered. Huang et al. [18] state that 20 to 100 nodes are suitable for an overview. Moreover, in specific applications, there may exist data elements being semantically more relevant than others. In such cases, including graph elements of higher relevance (e.g., outliers) can lead to a more appropriate initial view. One could also favor nodes with high degree as they potentially lead to more options for navigation along the graph structure. Despite these initial suggestions, creating a good initial view remains a difficult and largely context-dependent task.

The second concern regards the dependency of interestingness on the exploration state. To make this dependency clear to the user it makes sense to visualize a node's exploration state as well, because it may influence navigation decisions. When exploring hierarchical graphs the user might want to know which subtrees have already been explored. Cramming this additional information into the visualization as well is difficult. Therefore, we experimented with on-demand labeling

that classifies nodes into *unexplored*, *partially explored*, and *explored*. Such on-demand labels can help users to decide where to explore further and where no further exploration is necessary.

4 Proof-of-Concept Implementation

To test our approach, we developed a proof-of-concept implementation. As the underlying zoomable graph visualization, we use the *CGV* system [4]. We implemented a plausible default preset for the interestingness specification (including maximum attribute values and attribute outliers), which enables us to give recommendations at all times, even in cases where the user has not yet made the interests known to the system. The *DOI* function and its components can be altered interactively via a simple graphical user interface. We implemented arrow-based recommendation cues and our *enriched wedge*.

We tested the system with several hierarchical graphs. Here we illustrate its application with a graph that contains search queries as nodes and relations between the queries as edges. The graph is of moderate size with 695 nodes and 4073 edges. Figure 6 shows a partial view on the graph as the user may see it during exploration. Note that for the purpose of demonstration we use a visual encoding that might not appear as gentle and subtle as one would use it in a real application. In the figure, we can see recommendations to investigate on-screen targets, indicated by red circles around some nodes. Enriched wedges at the border of the screen recommend navigation to off-screen targets. Once the user has decided on the next navigation target, it can be visited by following the navigation recommendations manually. For example, the user can pan the view in the direction of an enriched wedge until the target falls into view. The target is then highlighted using the red circle for on-screen targets. As an alternative to manual navigation, we utilize *CGV*'s animation facilities to provide automatic animated traveling to the selected target. To this end, the user simply clicks an enriched wedge to trigger the animation.

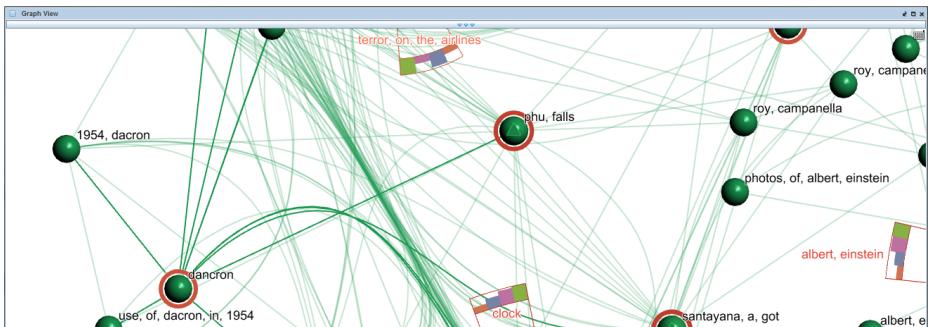


Fig. 6. Navigation recommendation in the proof-of-concept implementation. Red circles indicate on-screen targets worth investigating next. Enriched wedges indicate off-screen targets that might be of interest to the user as well.

During the exploration, the recommendations are constantly updated according to the current focus and the specification of the user's interest. The system also keeps track of which nodes have already been visited, where we rely on users explicitly marking a node as explored.

In summary, provided that users are able to express their current interest, our implementation can help in locating interesting nodes in the local context quickly.

5 Conclusion

In this work, we developed a data-driven approach for navigation recommendations to interesting information. Our solution is based on three basic steps: (1) collection of a set of recommendation candidates based on a compound focus, (2) selection of navigation recommendation based on user interests, and (3) visualization of navigation recommendations via visual cues embedded into an existing graph visualization.

Our solution extends the body of existing work in several aspects. We addressed hierarchical graphs, which require both horizontal and vertical navigation, an issue not studied in previous work. In terms of computing navigation recommendations, we generalized the notion of focus and context to incorporate the graph structure, the graph layout, and the data attributes. Further, we extended the widely-accepted *DOI* concept by the component *KNOW*, which captures the exploration state of data elements. For communicating navigation recommendations, we suggest several visual encodings, including the novel *enriched wedge*. A proof-of-concept implementation has been developed.

The mechanisms behind our concept work quite well in the proof-of-concept implementation. However, in the future, we need to develop a better interface for specifying the users' interests. Our current approach with classic GUI elements needs to be revised in order to make the overall solution more accessible for users. Further it makes sense to keep a history of what users have already marked as interesting. This would allow us to find better starting points for exploration.

A pressing issue is that the degree to which our solution reduces the trial-and-error character of visual exploration has not yet been quantified. And unfortunately we believe that it will be hard to do so due to the many influencing factors. Therefore, we invite evaluation experts to contact us and we will gladly collaborate and provide our implementation for in depth usability studies.

Acknowledgements. This research has been supported by the German Research Foundation (DFG) in the context of the project GEMS – graph exploration and manipulation on interactive surfaces.

References

1. Herman, I., Melançon, G., Marshall, M.S.: Graph Visualization and Navigation in Information Visualization: a Survey. *IEEE Transactions on Visualization and Computer Graphics* 6, 24–43 (2000)

2. Auber, D., Archambault, D., Bourqui, R., Lambert, A., Mathiaut, M., Mary, P., Delest, M., Dubois, J., Melançon, G.: The Tulip 3 Framework: A Scalable Software Library for Information Visualization Applications Based on Relational Data. Research Report RR-7860, INRIA (2012)
3. Mathieu, B., Heymann, S., Jacomy, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks. In: Proceedings of the International Conference on Weblogs and Social Media (ICWSM), pp. 361–362. Association for the Advancement of Artificial Intelligence (2009)
4. Tominski, C., Abello, J., Schumann, H.: CGV – An Interactive Graph Visualization System. *Computers & Graphics* 33, 660–678 (2009)
5. Spence, R.: *Information Visualization: Design for Interaction*, 2nd edn. Prentice-Hall (2007)
6. Abello, J., van Ham, F., Krishnan, N.: ASK-GraphView: A Large Scale Graph Visualization System. *IEEE Transactions on Visualization and Computer Graphics* 12, 669–676 (2006)
7. Elmqvist, N., Fekete, J.D.: Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics* 16, 439–454 (2010)
8. May, T., Steiger, M., Kohlhammer, J.D.J.: Using Signposts for Navigation in Large Graphs. *Computer Graphics Forum* 31, 985–994 (2012)
9. Jusufi, I., Klukas, C., Kerren, A., Schreiber, F.: Guiding the Interactive Exploration of Metabolic Pathway Interconnections. *Information Visualization* 11, 136–150 (2012)
10. Plaisant, C., Grosjean, J., Bederson, B.: SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. In: Proceedings of the IEEE Symposium on Information Visualization (InfoVis), pp. 57–64. IEEE Computer Society (2002)
11. Baudisch, P., Rosenholtz, R.: Halo: A Technique for Visualizing Off-Screen Objects. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), pp. 481–488. ACM Press (2003)
12. Gustafson, S., Baudisch, P., Gutwin, C., Irani, P.: Wedge: Clutter-Free Visualization of Off-Screen Locations. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI), pp. 787–796. ACM Press (2008)
13. Frisch, M., Dachselt, R.: Visualizing offscreen elements of node-link diagrams. *Information Visualization* 12, 133–162 (2013)
14. van Ham, F., Perer, A.: Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics* 15, 953–960 (2009)
15. Crnovrsanin, T., Liao, I., Wu, Y., Ma, K.L.: Visual Recommendations for Network Navigation. *Computer Graphics Forum* 30, 1081–1090 (2011)
16. Perer, A., van Ham, F.: Integrating Querying and Browsing in Partial Graph Visualizations. Technical report, IBM Research (2011)
17. Furnas, G.W.: Generalized Fisheye Views. *ACM SIGCHI Bulletin* 17, 16–23 (1986)
18. Huang, M.L., Eades, P., Wang, J.: On-line Animated Visualization of Huge Graphs Using a Modified Spring Algorithm. *Journal of Visual Languages and Computing* 9, 623–645 (1998)

A Tool for Visualizing Large-Scale Interactions between Turbulence and Particles in 3D Space through 2D Trajectory Visualization*

Guoyu Lu¹, Vincent Ly¹, Xiaolong Wang¹, Rohith M.V.¹, Orlando Ayala²,
Lian-Ping Wang², and Chandra Kambhamettu¹

¹ Video/Image Modeling and Synthesis (VIMS) Lab,
Department of Computer and Information Sciences

² Department of Mechanical Engineering,
University of Delaware, Newark, DE, USA

Abstract. Particle-laden turbulent flows are relevant to many industrial, biological, and environmental applications. In these flows, there are many interactions that occur between the carrier-fluid turbulence and the dispersed particles inside; visualization tools aid in understanding the complex physical phenomena at various spatial and time scales in such flows. For a large scale 3D particle-laden turbulence, visualizing the vector data in a dynamic environment makes the computation expensive. Furthermore, the interaction between the turbulence and particles is difficult to observe in 3D. The particle-turbulence interaction is easy to observe using 2D visualization, yet 2D visualization cannot properly capture the dynamics of the interactions. In this paper, we explore a method which can visualize the 3D effect of particle-turbulence interaction as a 2D trajectory visualization. We divide the 3D particle trajectory into several segments and, for each, we compute the plane crossing the line segment. As the turbulence vectors across the plane have a significant impact on the particle motion, we only visualize the planes where the particles are localized. Since we select the salient planes to visualize instead of the whole 3D vector cube, considerable improvements can be achieved with regards to the memory and time performance. For each plane, we use Line Integral Convolution to visualize the turbulence. Originally, Line Integral Convolution only visualizes the vectors' direction. Instead, we not only visualize the vector directions but also the magnitude of the flow by utilizing a more meaningful texture, which generates a more compelling visualization.

1 Introduction

Understanding particle-turbulence interactions in particle-laden turbulent flows is essential to many applications in engineering and environmental science. Examples include sediment transport, cloud physics, air-sea interaction, fluidized bed reactor, food processing, mining, and composite manufacturing. Air turbulence in atmospheric clouds has been known to affect the collision-coalescence rate of cloud droplets and as such can impact warm rain initiation and cloud dynamics [5]. Due to the complex multiscale physics phenomenon in turbulence, researchers try to visualize the turbulence and

* Supported by NSF OCI-0904534

the particles within to observe patterns and dynamic interactions in the flow. As the turbulence and the particles are described by vectors in each 3D position and vectors change dynamically for each time point, it is difficult to clearly visualize the interaction. Meanwhile, the memory requirement is large and the computation cost (time) for visualizing the whole 3D environment is expensive, which would then require visualization via GPU [15]. For these reasons, most visualization research on the particle-turbulence interaction is based on 2D visualizations or just visualizing either the turbulence or the particle movement in 3D environment. As the particles and the turbulence are evolving in the 3D space, 2D visualization will result in incomplete information for observing the particle movement in 3D space, as well as the flow evolution. Just visualizing either particles or the turbulence cannot reveal the interactions between the particles and turbulence. Furthermore, it is difficult to observe the turbulence from within the 3D domain; where excessive turbulence and particle information become visual impediments to observing the trajectory of interest.

In this paper, We introduce a tool which can visualize the interaction between the turbulence and particles in 3D space through a 2D visualization trajectory. Analyzing the path of a particle in the 3D space, we find that the particle path usually lies on several different planes. The particle's movement is influenced by the surrounding flow vectors. For this reason, the flow vectors across the plane where the particle localizes has the most significant impact on the particle's movement and also give us an indication as to how the particle responds to the turbulence in a certain path interval. We divide the point path into several segments. For each segment, we compute the plane containing the segment in the 3D space. Then each plane can be visualized in 2D images. Connecting the 2D visualization planes together forms a 2D trajectory, allowing the particle movement in 3D space to be visualized clearly in 2D space.

Flow vectors crossing the plane slices form the turbulence on the plane. We use Line Integral Convolution (LIC) to visualize the turbulence. LIC visualizes the flow based on the vectors' orientation, but ignores the magnitude information. Incorporating with LIC, we introduce a method that the color represents the turbulence direction and the color saturation indicates the strength of the turbulence. Thus, the visualization can contain more information on particle-turbulence interactions.

2 Related Work

Turbulence visualization is of interest for both the visualization and the fluid mechanics communities. In an early work, Hin et al. [6] used particle motion animation to visualize a three dimensional turbulent flow. The turbulence is decomposed into consecutive motions based on the Reynolds decomposition. The visualization effect is basically a set of particle path lines in a container. Bec et al.[2] and Coleman et al. [4] visualize the particle density on a given 2D plane based on clustering the particles less than a threshold together. Yang et al. [18] simulated and visualized the instantaneous velocity maps superimposed on the corresponding vorticity maps for the case of fan-stirred near-isotropic turbulence whose velocity maps is described by the vector space. Toschi and Bodenschatz [15] visualized the particle trajectory in a 3D place. However, the turbulence information is missing in the visualization. Jin, He and Wang [15] visualized

the preferential concentration of particles with different Stokes numbers in a slice with the background of vorticity contour.

Direct Numerical Simulation (DNS) is often used in the above studies for turbulence simulation. Toschi & Bodenschatz [15] showed the DNS visualization result of the migration and the distortion of a continuous stream of grid turbulence passing through a representative high-pressure turbine cascade. MV et al. [10] explores particle collision in turbulent fluid in DNS studies. [16] develops a system which can explore large turbulent flow datasets on desktop PCs. Second eigenvalue, vorticity magnitude and entropy are used for visualizing the turbulence, whose original vector data is compressed by a wavelet-based scheme [17] on GPU. The visualization effect shows high quality rendering with an order of magnitude acceleration. However, this system puts concentration on turbulence visualization with particle information omitted.

Line Integral Convolution [3] is used for representing the motion of the objects making use of continuous motion filters. Stalling & Hege [13] proposed to reduce the computation costs using box filter kernels with the minimization of stream line. Sundquist [14] designed one new convolution algorithm – Dynamic Line Integral Convolution (DLIC) to visualize the evolution of vector field, which is based on the dependence of time along with the evolution of streamlines. Many advances in LIC have taken advantage of color to highlight information such as in [11].

For large-scale data, many tools have been developed for visualization. To avoid the dependence on lightweight kernels, Moreland et al. [9] implemented ParaView on a Cray XT3 supercomputer. Shi et al. [12] proposed a system called HiMap, which could visualize social networks using clustered graph based on the hierarchical structure. Leigh et al. [8] introduced a work that is specifically relevant for the visualization on the large scale dataset and proposed one scheme to deal with the key data rendering. Authors in [7] demonstrated that the layout based on QAPgrid algorithm could be used to represent the relationships between objects and clusters in the given dataset. However, how it can be used to clearly and efficiently visualize information like particles and the turbulence in 3D environment is unclear.

3 Methodology

Before explaining our method for visualizing the flow and the particles, we first introduce our data for the visualization. The data is obtained by a DNS simulation on the Bluefire computer at NCAR. The simulation includes the water droplets with different radii suspended in the turbulence. The turbulent flows were simulated by solving Navier-Stokes equations in the spectral domain. The particle movement was influenced by viscous drag, gravity, inertia and the turbulence. It is assumed that the particles do not affect the turbulent flow. The simulations were conducted on a grid of 256 points in X, Y and Z directions. We select the central 21 points in each direction for making the visualization clearer. The vector space with a particle trajectory is shown on Fig. 1. The simulation details can be found in [1].

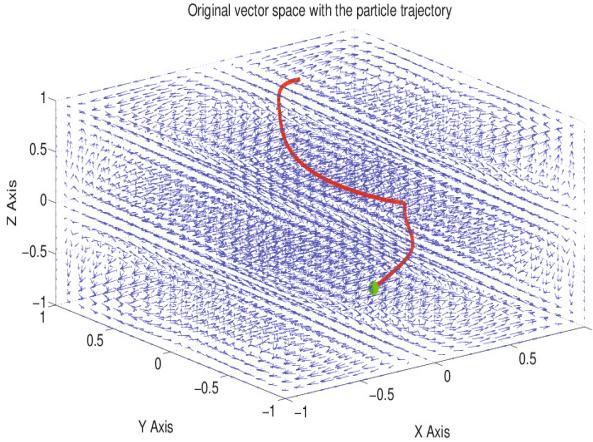


Fig. 1. Original vector space with a particle trajectory outlined in red

3.1 Particle Path Segmentation

In order to visualize the particle path such that the resulting images share similar space, the particle path must be cut in a way such that the size of the planes are similar. To accomplish this, we divide the particle path into several segments. From the starting point of the trajectory, we find the nearest point in the trajectory to the euclidean distance of d , which satisfies Eq. 1.

$$P_{end} = \min_{P \in \mathbb{Q}} \left| \sqrt{P_{start} \cdot P^T} - d \right| \quad (1)$$

where P_{start} is the position vector of the starting point of the trajectory. T is the transpose operator. \mathbb{Q} is the set of points from P_{start} till the end of the trajectory. P_{end} is the ending point for the segment, which will also be the starting point for the next segment.

3.2 Plane Fitting

For each particle trajectory segment, we compute the covariance matrix, which gives the variance between every variable, as in Eq. 2.

$$\sum = \mathbb{E}_{(P, P') \in \mathbb{P}} \{(P - P')(P - P')^T\} \quad (2)$$

\sum is the covariance matrix of all the points. E is the expectation operation. (P, P') contains the vectors of the point pair P and P' which belongs to the point set of the current segment \mathbb{P} .

We perform eigen decomposition of the covariance matrix \sum . The eigenvector corresponding to the smallest eigenvalue will be selected as the plane normal N . The plane equation is shown in Eq. 3. P_{mid} is the midpoint of the starting and ending point of the segment, as indicated in Eq. 4.

$$(P - P_{mid}) \cdot N = 0 \quad (3)$$

$$P_{mid} = (P_{start} + P_{end})/2 \quad (4)$$

3.3 Plane Transformation

After processing the plane equation, we then judge flow vectors across the plane and project the vectors onto the plane for further visualization. To facilitate this goal, we transform the fitted plane to the plane with a normal aligned to the Z axis and align the plane center to $(0, 0, 0)$. This step is accomplished by a translation followed by two rotations.

Transformation. Aligning the plane center to coordinate origin, all the points will be transformed by the following transformation matrix in homogeneous coordinate.

$$\begin{vmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5)$$

$$\begin{aligned} Tx &= -P_{midX} \\ Ty &= -P_{midY} \\ Tz &= -P_{midZ} \end{aligned} \quad (6)$$

P_{midX} , P_{midY} and P_{midZ} are the X , Y , and Z coordinate of the original plane center, which we computed in the plane fitting part. The new normal of the plane after translation is as shown in Eq. 7 below.

$$N' = \begin{vmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot N \quad (7)$$

Rotation. First we align the plane to $Z = 0$, we compute the rotation based on Rodrigues rotation formula.

$$V_{rot} = V \cos\theta + (K \times V) \sin\theta + K(K \cdot V)(1 - \cos\theta) \quad (8)$$

$$\theta = |0, 0, 1| \cdot N' \quad (9)$$

$$K = \frac{|0, 0, 1| \times N'}{\sqrt{|0, 0, 1| \times N'}} \quad (10)$$

K is the normalized rotation axis and θ is the rotation angle. V_{rot} is the output of the point position vector and V is the original point position vector input. After aligning the plane to the $Z = 0$ plane, we align the plane to the image plane where the starting point of the line segment lies on the top and the ending point lies on the bottom, which

result in the line between starting point and ending point lying on the Y axis. Similarly, the rotation axis and rotation angle is calculated as follows:

$$\theta' = |0, 1, 0| \cdot V_{up} \quad (11)$$

$$K' = \frac{|0, 1, 0| \times V_{up}}{\sqrt{|0, 1, 0| \times V_{up}}} \quad (12)$$

$$V_{up} = V_{starting} - V_{ending} \quad (13)$$

$V_{starting}$ and V_{ending} are the starting and ending point positions of the line segment. K' and θ' are the rotation axis and the rotation angle.

3.4 Vector Field Uniform Sampling

In order to visualize the vector volume using 2D tools such as line integral convolution, the 3D vector field must be projected to a 2D plane. This is accomplished, using the rotations and translations stated above, to align the plane of interest to the image plane. An additional scaling and translation is applied to obtain a sub-section of the plane. All the vectors will also be transformed by the plane transformation matrix. After the transformation, the vectors with different sign of Z value on the vectors' starting and ending point are the vectors crossing the plane. The X and Y value of the vectors passing the plane can be directly used as the coordinate value of the projected vector on the 2D plane. The flow vectors projected on the 2D plane is shown in Figure 4(a).

However, simple projection would result in an erroneous vector field that does not consider the distance from the plane and is non-uniformly sampled. As a non-uniformly sampled vector field is not conducive to producing quality visualizations, the vector volume must be uniformly sampled. To uniformly sample the vector field, we estimate the vector $v_{x,y}$ for point x, y on the plane by computing as shown below.

$$v_{x,y} = \frac{\sum_{i=1}^N w_i v_i}{\sum_{i=1}^N w_i}. \quad (14)$$

where w_i is the attenuation of the point x, y on the plane to the vector v_i at x_i, y_i as in the following:

$$w_i = \frac{1}{a * d^2 + b * d + c} \quad (15)$$

$$d = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \quad (16)$$

In Equation 15, a , b , and c are coefficients controlling how much attenuation to apply. Increasing the coefficients for higher order terms results in sharper attenuation, which is useful for minimizing the influence of distant vectors. Experimentally, we set a to 1, b to 0.5, and c to 1 through visual inspection. These parameters produce sufficient results in our experiments as can be seen in Figure 4.

3.5 Line Integral Convolution

The Line Integral Convolution (LIC) algorithm is a powerful visualization tool for 2D vector fields, allowing the overall motion to be understood easily at a glance [3]. LIC uses an input texture map and a vector field to output a scalar field by blurring pixels along the vector field's general motion. This is accomplished by applying a low-pass filter on the input texture image directed by the vector field as in

$$L_{x,y} = \frac{\sum_{i=0}^l P_i h_i + \sum_{i=0}^{l'} P'_i h'_i}{\sum_{i=1}^l h_i + \sum_{i=1}^{l'} h'_i}, \quad (17)$$

$$h_i = \int_{s_i}^{s_i + \Delta s_i} \kappa(w) dw. \quad (18)$$

In other words, the output LIC pixel $L_{x,y}$ is generated by convolving along the line of length l along the positive direction a pixel P_i on the line by a weight h_i . The same is performed along the negative direction, of a line of length l' with pixel P'_i and weights h'_i . The weights are generated by the exact integral of the convolution kernel $\kappa(w)$ between the point s_i and $s_i + \Delta s_i$ along the streamline, where Δs_i is the arclength between the points s_i and s_{i+1} .

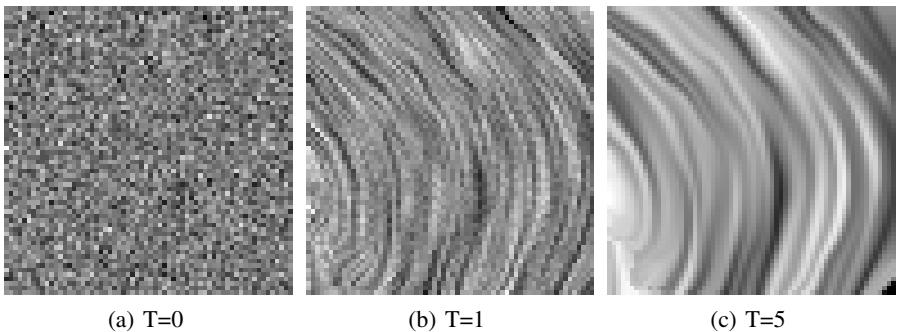


Fig. 2. Convolution of an Input Texture Using LIC

While LIC is useful for visualizing the overall motion, it does not visualize the motion magnitude or direction. It is possible to adjust the input texture, however. Instead of an image generated by random noise, the vector field is encoded into an HSV (hue, saturation, and value) image. The direction is encoded into the hue and the magnitude is encoded into the saturation. This image is then converted to RGB colorspace and given to LIC to produce the final result as in Figure 5.

4 Experiments

While we conducted a number of experiments, we describe one example here. We obtained data by a DNS simulation on the Bluefire computer at NCAR and tracked a water

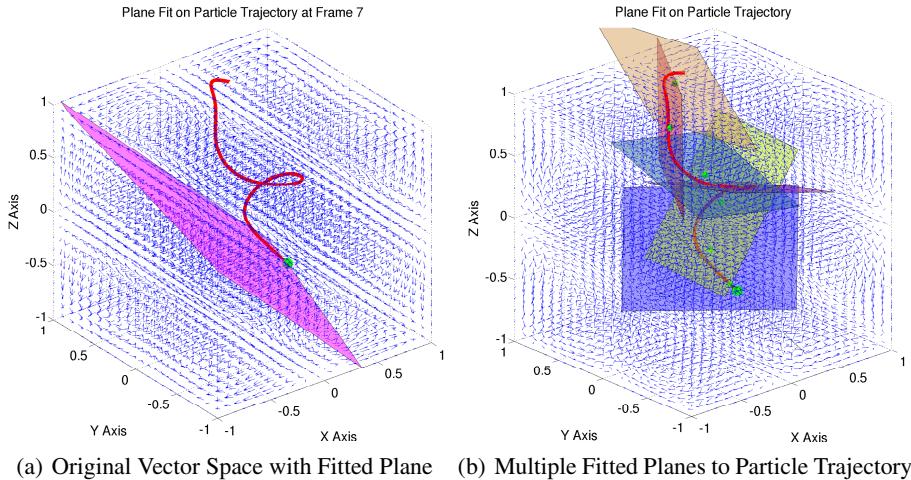


Fig. 3. Multiple Fitted Planes to Particle Trajectory

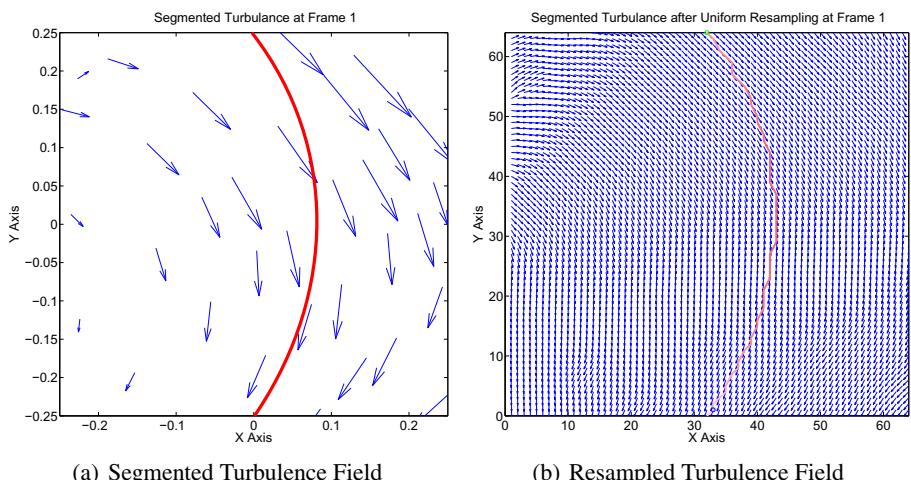


Fig. 4. Resampling the segmented turbulence field

droplet as it progressed through the turbulence. Our visualization is then performed on a computer running Ubuntu 10.04 and Intel(R)Core(TM)2 CPU 6600 with a frequency of 2.4 GHz. We applied plane fitting to the droplet trajectory as shown in Figure 3.

After determining the planes and segmenting the droplet trajectory, the turbulence field can be resampled to form a dense vector field. In Figure 4(a), the turbulence vectors closest to the plane is shown. Figure 4(b) is the turbulence resampled on the image plane.

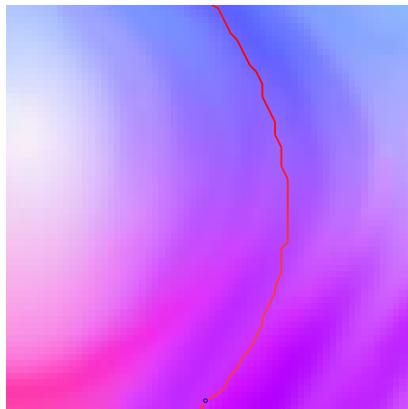


Fig. 5. After Replacing Random Noise with Image Encoded Vector Field



Fig. 6. Motion Trajectory Segmented into Multiple Planes

Once a dense, uniformly-sampled vector field has been obtained, LIC can be computed which can be seen in Figure 2. Instead of using the standard texture, we used the encoded vector field as input to LIC for convolution. Further, we then project the particle trajectory onto the image plane. The result can be seen in Figure 5.

For visualizing each plane with one LIC iteration, our unoptimized code just need 0.2 seconds in Matlab which means generating a very smooth turbulence visualization (4 iterations) needs only 0.8 seconds.

5 Conclusion

In this paper, we developed a tool for visualizing the interaction of turbulence and particles in 3D space by a 2D trajectory visualization. From each 2D plane slice, we can clearly observe the particle and turbulence behavior. Instead of just visualizing the turbulence motion, our LIC method can visualize the turbulence magnitude, using the color saturation, and the flow direction, by different hue. Rather than visualizing the whole 3D plane, we only consider the flow vectors which play important role in influencing

the particle motion. The visualization shifts from 3D to 2D allowing for more concise observations. We have used this tool on multiple flows; an example of the output of this tool can be seen in Figure 6. This idea can be extended into many other large-scale stereo visualization tasks to conduct 3D visualization using 2D images. Future work includes visualizing larger simulated data with consideration to the particle.

References

1. Ayala, O., Grabowski, W., Wang, L.P.: A hybrid approach for simulating turbulent collisions of hydrodynamically-interacting particles. *J. Comput. Phys.* 225 (2007)
2. Bec, J., Biferale, L., Cencini, M., Lanotte, A., Musacchio, S., Toschi, F.: Heavy particle concentration in turbulence at dissipative and inertial scales. *Phys. Rev. Lett.* 98 (2007)
3. Cabral, B., Leedom, L.C.: Imaging vector fields using line integral convolution. In: *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (1993)
4. Coleman, S., Vassilicos, J.: A unified sweep-stick mechanism to explain particle clustering in two-and three-dimensional homogeneous, isotropic turbulence. *Physics of Fluids* 21 (2009)
5. Grabowski, W., Wang, L.P.: Growth of cloud droplets in a turbulent environment. *Annual Review of Fluid Mechanics* 45 (2013)
6. Hin, A., Post, F.: Visualization of turbulent flow with particles. In: *Proceedings of the 4th Conference on Visualization* 1993 (1993)
7. Inostroza, M., Berretta, R., Moscato, P.: Qapgrid: A two level qap-based approach for large-scale data analysis and visualization. *PloS One* 6 (2011)
8. Leigh, J., Johnson, A., Renambot, L., Vishwanath, V., Peterka, T., Schwarz, N.: Visualization of large-scale distributed data. *Data Intensive Distributed Computing: Challenges and Solutions for Large-Scale Information Management* (2010)
9. Moreland, K., Rogers, D., Greenfield, J., Geveci, B., Marion, P., Neundorf, A., Eschenberg, K.: Large scale visualization on the cray xt3 using paraview. *Cray User Group* (2008)
10. MV, R., Parishani, H., Ayala, O., Wang, L.-P., Kambhamettu, C.: CollisionExplorer: A tool for visualizing droplet collisions in a turbulent flow. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Wang, S., Kyungnam, K., Benes, B., Moreland, K., Borst, C., DiVerdi, S., Yi-Jen, C., Ming, J. (eds.) *ISVC 2011, Part II. LNCS*, vol. 6939, pp. 669–680. Springer, Heidelberg (2011)
11. Shen, H.W., Johnson, C.R., Ma, K.L.: Visualizing vector fields using line integral convolution and dye advection. In: *Proceedings of the 1996 Symposium on Volume Visualization* (1996)
12. Shi, L., Cao, N., Liu, S., Qian, W., Tan, L., Wang, G., Sun, J., Lin, C.: Himap: Adaptive visualization of large-scale online social networks. In: *2009 IEEE Pacific Visualization Symposium* (2009)
13. Stalling, D., Hege, H.-C.: Fast and resolution independent line integral convolution. In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995)
14. Sundquist, A.: Dynamic line integral convolution for visualizing streamline evolution. *IEEE Transactions on Visualization and Computer Graphics* 9 (2003)
15. Toschi, F., Bodenschatz, E.: Lagrangian properties of particles in turbulence. *Annual Review of Fluid Mechanics* 41 (2009)
16. Treib, M., Burger, K., Reichl, F., Meneveau, C., Szalay, A., Westermann, R.: Turbulence visualization at the terascale on desktop pcs. *IEEE Transactions on Visualization and Computer Graphics* 18 (2012)
17. Treib, M., Reichl, F., Auer, S., Westermann, R.: Interactive editing of gigasample terrain fields. *Computer Graphics Forum* 31 (2012)
18. Yang, T.S., Shy, S.S.: Two-way interaction between solid particles and homogeneous air turbulence: particle settling rate and turbulence modification measurements. *Journal of Fluid Mechanics* 526 (2005)

Visual Query Specification and Interaction with Industrial Engineering Data

Alberto Malagoli¹, Mariano Leva², Stephen Kimani³, Alessandro Russo²,
Massimo Mecella², Sonia Bergamaschi¹, and Tiziana Catarci²

¹ Università di Modena e Reggio Emilia, Italy

albemala@gmail.com, sonia.bergamaschi@unimore.it

² Sapienza Università di Roma, Italy

{leva, arusso, mecella, catarci}@dis.uniroma1.it

³ Jomo Kenyatta University (JKUAT), Kenya

stephenkimani@gmail.com

Abstract. Nowadays, industrial engineering environments are typically characterized by sensors which stream massive amounts of different types of data. It is often difficult for industrial engineers to query, interact with, and interpret the data. In order to process many different kinds of distributed data stream sources originating from different kinds of data sources, a distributed federated data stream management system (FDSMS) is necessary. Although there exist some research efforts aimed at providing visual interfaces for querying temporal and real time data, there is virtually no existing work that provides a visual query specification and interaction interface that directly corresponds to a distributed federated data stream management system. This paper describes a visual environment that supports users in visually specifying queries and interacting with industrial engineering data. The visual environment comprises: a visual query specification and interaction environment, and a corresponding visual query language that runs on top of a distributed FDSMS.

Keywords: Visual query language, interaction, user, usability evaluation, data streams, industrial engineering.

1 Introduction

Nowadays, industrial engineering environments are characterized by a wide variety of equipment including sensors which continuously stream raw data. Such data is multidimensional and massive. In particular, the arrays of sensors in the distributed industrial engineering settings continuously send in huge volumes of data. Different sensors also transmit different types of data. It is also often the case that sensor networks in the target environment will continue to expand, add new variables to the infrastructure, etc. Moreover, such data is temporal in nature. Such data may in fact be transient, and arriving in a continuous and unbounded fashion as a data stream.

The foregoing characteristics make it difficult for industrial engineers and data analysts to query, interact with, and interpret the data. In order to process many different

kinds of distributed data stream sources originating from different kinds of equipment that typically characterize industrial engineering settings, a distributed federated data stream management system (FDSMS) becomes necessary. Although there exist some research efforts aimed at providing visual query languages for supporting querying and interaction with temporal and real time data (such as Haigh et al. in [1], Dionisio and Cardenas in [2]), to the best of our knowledge there is no existing research work that provides a visual query specification and interaction environment that directly corresponds to a distributed federated data stream management system.

This paper proposes and describes a visual environment that supports industrial engineers in visually specifying queries and interacting with different kinds of distributed streams of data originating from different kinds of equipment. The visual environment in particular comprises: a web-based visual query specification and interaction environment, and a corresponding visual query language that runs on top of a distributed FDSMS named Super Computer Stream Query processor (SCSQ) [3]. SCSQ supports queries over data streams and also over data stored in classical databases or files. The visual environment has been designed within the EU SmartVortex project [4].

The rest of the paper is organized as follows. Section 2 describes the state-of-the-art in the visual querying of and interaction with data streams. Section 3 describes the proposed SmartVortex web-based environment for visual query specification and interaction with industrial engineering data. Section 4 concludes the paper and points out future research work.

2 State-of-the-Art in the Visual Querying of Data Streams

Visual Query Systems (VQSS) are systems that use visual representations to depict the domain of interest and express related database requests. One of the earliest visual query languages was introduced in the mid-1970s, namely Query By Example (QBE) [5]. A wide range of implementations were built using the QBE concepts and there are several tools using this paradigm today. A general overview and classification of VQSSs can be found in [6].

Most research on visual query languages (VQLs) addresses canonical Database Management Systems (DBMSs), but querying a data stream usually requires different constraints. A large number of academic and commercial continuous queries languages and their corresponding Data Stream Management Systems (DSMSs) have been devised. Although there is not yet a standard proposal for a data stream query language, two main models and semantics exist namely: the Stanford Stream Data Manager (STREAM) [7] and StreamBase [8].

There have been some attempts to develop a visual query language able to interact with temporal databases recording multiple versions of objects. In [1], Haigh et al. present a technology for locating time series patterns in historical or real time data whilst in [2] Dionisio and Cardenas illustrate a visual query language for multimedia, timeline and simulation data using a single set of related query constructs. However,

there are no examples of visual query languages directly corresponding to or reproducing continuous query languages.

In our research, many different kinds of data stream sources originating from different kinds of equipment need to be processed, which leads to the requirement for a distributed FDSMS where continuous queries can be specified that combine data streams in different formats from different distributed data stream sources. This is enabled through SCSQ [3], where both streaming and regular/static data sources can be incorporated and where continuous queries are expressed using a textual data stream query language called SCSQL. Through the proposed SmartVortex visual environment, the industrial engineer can thus visually specify continuous queries in the same manner the user builds classical static queries, hiding the complexity related to the data stream.

3 Visual Query Specification and Interaction with Industrial Engineering Data

The development of the proposed SmartVortex web-based environment for visual query specification and interaction with industrial engineering data has been based on the user-centered design (UCD) approach. UCD places users at the core of the design [10]. We started by collecting user requirements from our industrial engineering partners. We then came up with real world usage scenarios informed by the industrial engineering partners. After that we developed the initial design of our proposed visual environment. The visual environment comprises: a web-based visual query specification and interaction environment, and a corresponding visual query language. We then conducted a usability evaluation of the visual environment. This section gives a detailed description of the design process that was followed and the visual environment.

3.1 Preliminary User Requirements and Domain Tasks

We recently conducted a requirements analysis study with the industrial engineering partners in the SmartVortex project. The study found that some of the tasks that are key to industrial engineering operators include:

- Analyze and predict system/product performance (including its maintenance, fault detection, fault prevention, fault diagnosis).
- Analyze and predict the usage of system/product/service.
- Analyze and predict customer service/product needs.

The above findings echo an observation by Uraikul et al. in [11] that operators often need to monitor systems, processes and products, assessing their current state, and detect and diagnose any abnormal behavior.

It is worth pointing out that the proposed visual environment provides a user oriented visual interface to SCSQL. The proposed visual environment supports the

aforementioned industrial engineering domain tasks by supporting the visual specification of queries corresponding to numerical models such as: fault detection, prediction, etc. Such models/algorithms are called as foreign functions in the underlying SCSQL. This is possible due to the extensible nature of SCSQL.

The queries filter, transform, and combine the streams from the distributed equipment. Queries over incoming raw data streams from equipment produce substantially reduced derived data streams that contain information about only malfunctioning equipment; when the equipment works as expected the derived data stream is idle. Since the total stream flow rate can be very high, SCSQ supports parallelization functions with which customized massively parallel continuous queries can be specified in SCSQL. This is important for obtaining timely results from expensive stream analyses.

An important consideration in the proposed visual environment is to validate whether data streaming from industrial equipment behaves as expected. When unexpected behavior of data streaming from some equipment is detected by a continuous query, a stream of error messages (alert streams) and other data streams characterizing the faulty behavior is emitted. In many cases the processed data streams can be stored persistently on disk and in particular when the continuous queries reduce the data volume. For this, SCSQ provides facilities for storing outgoing streams as log files that can be analyzed later by visually specifying them as queries in our visual environment.

3.2 SmartVortex Visual Query Language

The proposed visual query language provides industrial engineers with a visual mapping with SCSQL's textual query language. The main visual query elements can be seen in Fig. 1.

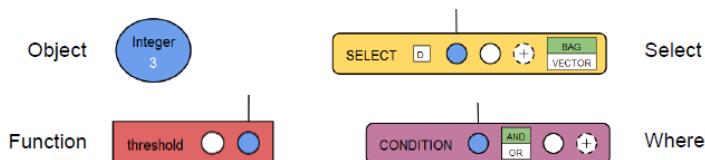


Fig. 1. Visual query elements

An industrial engineer can use those visual query elements in order to visually specify a query. The visual query language also provides visual constructs for specifying logical connections between visual query elements. For instance, to indicate that a certain visual query element is utilized as a parameter for a certain function. When the user is in the process of specifying a connection, the visual environment uses visual indicators to alert the user whether the intended connection is valid (green color) or invalid (red color) as seen in Fig. 2. The user can therefore immediately understand if something is wrong, potentially preventing execution errors from occurring.

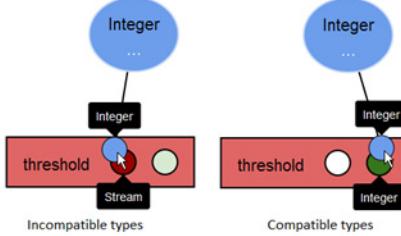


Fig. 2. The two possible connection outcomes between a function and a potential parameter

The left-hand side of Fig. 3 shows an example of how the visually constructed query looks. The visual query corresponds to the textual query on the right-hand side of the same figure. The user is not required to know all the meanings of the existing SCSQL functions. On mouse hover in fact a tooltip appears over the function describing it and providing its signature.

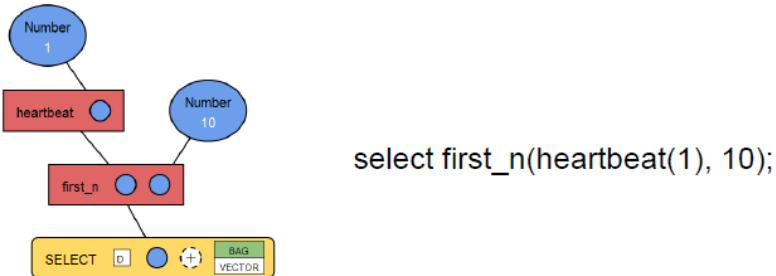


Fig. 3. Example of the visual representation of a query

3.3 SmartVortex Visual Query Specification and Interaction Interface

The visual query specification and interaction interface consists of three main components as seen in Fig. 4:

- *Canvas*: It is used to hold the visual query. Here visual query elements can be added, removed, and connected in order to compose the final query. Elements can also be freely arranged on the canvas.
- *Elements toolbox*: It contains all the visual query elements that can be used to construct a visual query. These elements are grouped by: Types, Functions, Data Sources and Instructions.
- *Toolbar*: It is placed over the canvas. It contains the “Set object value” and “Remove/Clear all elements” buttons. The first one is used to specify a value for an object, whilst the second one is used to remove an element or clear all the elements from the canvas.

Specific visual query elements are color coded in order to make their recognition easier. For instance: purple for a condition statement, yellow for a select statement or

a return statement, blue for objects, and green for basic procedural instructions; functions have a red header to represent the function name, white left-aligned rows for parameters types, and blue right-aligned rows for the result type. An example can be seen in Fig. 4. Tooltips are also available on data sources.

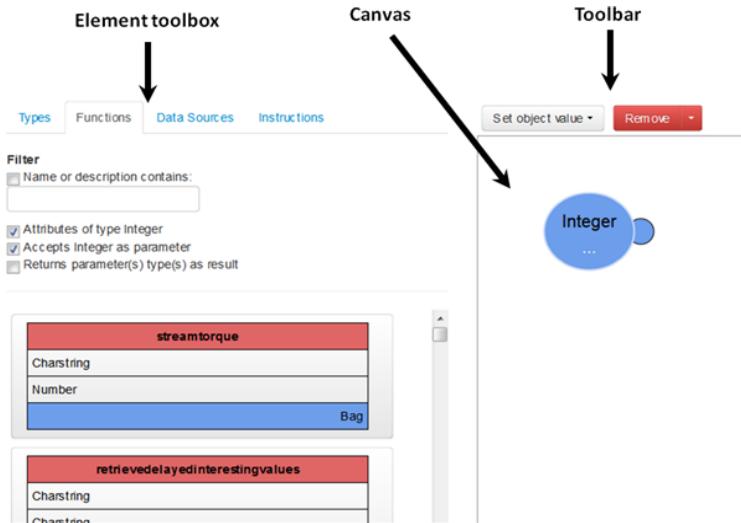


Fig. 4. The main components of the visual interface

In order to visually specify a query and interact with industrial engineering data, the user typically performs the following:

- **Picking the right elements.** To graphically compose this query, the user needs a set of types and functions, in addition to the condition statement to represent the WHERE clause.
- **Using filters.** Given the large number of existing types and functions, the visual environment provides a set of filters to speed retrieval. Each filter is activated by selecting a checkbox.
- **Setting object value.** Once the different elements are retrieved, the user can connect them, passing parameters to the functions, choosing items to include in the FROM clause, etc.
- **Translating the visual query.** This enables the user to instruct the application to show the textual query corresponding to the visual query that s/he has visually constructed. The translation is done through a button named “Translate Visual to Textual Query”.
- **Configuring the display of query results.** After visually specifying a query, the user can configure the visualization of the query result, as seen in the top part of Fig. 5.



Fig. 5. Configuring the display of query results

3.4 Usability Evaluation

The initial design of the visual query specification and interaction environment was subjected the visual environment to a usability evaluation. As is typical in UCD, initial designs are often evaluated by Human-Computer Interaction (HCI) experts.

The usability evaluation involved three HCI experts who also have database knowledge. The profile of the chosen experts therefore enabled us to get:

- HCI expert advice and feedback regarding the visual environment. In this case, the experts were required to conduct heuristic evaluation whose main purpose is to identify usability problems in a design [12].
- User-relevant feedback because the HCI experts had database knowledge. The experts were requested to assess the visual environment in terms of ease of use of the application, understandability/interpretability of the visual representations in the VQL and user interface, and satisfaction.

The experts first attended an interactive seminar where the application was introduced and demonstrated. They were then individually required to conduct the usability evaluation of the application. They were each provided with:

- The SmartVortex application.
- A brief description of the application, including the corresponding task analysis.
- Real-world usage scenarios for the application as provided by our industrial engineering partners.
- Jakob Nielsen's ten usability heuristics for user interface design [12].

- An evaluation form which had three parts:
 - Part 1: for supporting the experts in conducting heuristic evaluation.
 - Part 2: for the experts to indicate what they considered would be the user's assessment of the ease of use of the application, understandability/interpretability of the visual representations in the VQL and user interface.
 - Part 3: for the experts to indicate any existing functionalities and features they considered users would like about the application; and any other comments.
- An evaluation protocol which guided the experts regarding the evaluation.
 - The experts were in particular requested to interact with the application with reference to the foregoing materials, assess the application, and then complete the evaluation form.

The findings from the resultant individual completed evaluation forms were appropriately aggregated in order to realize usability evaluation findings which are summarized below.

On a positive note, the usability evaluation found that with the SmartVortex application it is easy for the user to:

- Visually create queries based on stored log files.
- Correctly interpret/understand the visual representation of the display of query results.

Moreover, the following functionalities/features were liked about the SmartVortex application:

- Visual validation and/or error prevention such as through visual validation indicators when one is intending to make connections between visual query elements e.g. green indicator to show that the intended connection is valid.
- Visual mapping of query elements/objects to colors (i.e. element toolbox vs. canvas).
- Zoom feature in the display of query results.
- Direct translation between visual and textual representation of the query.

On a negative note, the usability evaluation highlighted the following as critical usability issues with the SmartVortex visual environment:

- It is not easy for the user to correctly interpret/understand the visual representation of the visual elements of the visual query.
- The application does not have Help functionality/feature.
- The zoom feature in the display of query results is not visible i.e. the user interface does not have "affordances" corresponding to the feature.
- When query elements obscure each other, it is difficult to know which elements are connected unless one rearranges/drägs the elements. The user may be forced to remember/recall, and especially when observing old queries.
- While in a particular dashboard/page, the user cannot directly access/move to (or navigate) the other dashboards/pages.

The visual environment is currently being refined in line with the foregoing usability evaluation results. After that, the improved design of the visual environment will be subjected to an evaluation involving the intended users.

4 Conclusions and Future Work

This paper has highlighted data challenges associated with industrial engineering settings. It has also described the state-of-the-art in supporting the visual querying of data streams. This paper has also proposed and described a visual environment that can support users to visually specify queries and interact with industrial engineering data.

Once the current refinements are completed, we will carry out a user-based evaluation on the improved design. It is also worth pointing that even though in this paper we have primarily focused on visual query specification and access to industrial engineering data, we are also working on the implementation of interactive visualizations of query results [13].

We also intend to extend the SmartVortex application to support the visual creation of queries with tuple/time-based windows. Moreover, we plan to extend the application to be able to evaluate the cost-benefit ratio behind the introduction of complex statements like partitioned windows.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. *Int. J. Digit. Libr.* 1, 108–121 (1997)
2. Haigh, K.Z., Foslien, W., Guralnik, V.: Visual Query Language: Finding Patterns in and Relationships Among Time Series Data. In: Proc. 7th Workshop on Mining Scientific and Engineering Datasets (2004)
3. Dionisio, J.D.N., Cárdenas, A.F.: MQuery: A Visual Query Language for Multimedia, Timeline and Simulation Data. *Journal of Visual Languages & Computing* 7(4), 377–401 (1996)
4. Zeitler, E., Risch, T.: Massive Scale-out of Expensive Continuous Queries. In: Proc. VLDB (2011)
5. SmartVortex Project, <http://www.smartvortex.eu/>
6. Zloof, M.M.: Query by Example. In: Proc. National Computer Conference, vol. 44, pp. 431–438. AFIPS Press (1975)
7. Catarci, T., Costabile, M.F., Levialdi, S., Batini, C.: Visual Query Systems for Databases: A Survey. *Journal of Visual Languages and Computing* 8(2), 215–260 (1997)
8. Arasu, A., Babcock, B., Babu, S., Cieslewicz, J., Datar, M., Ito, K., Motwani, R., Srivastava, U., Widom, J.: STREAM: The Stanford Data Stream Management System. Tech. Report. Stanford InfoLab, <http://infolab.stanford.edu/~usriv/papers/streambook.pdf>
9. StreamBase home page, <http://www.streambase.com/>
10. Abadi, D.J., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., Zdonik, S.: Aurora: a new model and architecture for data stream management. *VLDB Journal* 12(2), 120–139 (2003)

11. ISO/IEC: Human-Centred Design Processes for Interactive Systems (1999)
12. Uraikul, V., Chan, C.W., Tontiwachwunthikul, P.: An Integrated Framework for Intelligent Monitoring, Diagnosis and Supervisory Control of Processes. In: IEEE Canadian Conference on Electrical and Computer Engineering, pp. 653–656 (2005)
13. Nielsen, J.: Heuristic evaluation. In: Nielsen, J., Mack, R.L. (eds.) Usability Inspection Methods. John Wiley & Sons, New York (1994)
14. Kimani, S., Leva, M., Mecella, M., Catarci, T.: Visualization of Multidimensional Sensor Data in Industrial Engineering. In: Proc. Information Visualisation 2013. IEEE (to appear, 2013)

Performance Anchored Score Normalization for Multi-biometric Fusion

Naser Damer, Alexander Opel, and Alexander Nouak

Fraunhofer Institute for Computer Graphics Research IGD,

Fraunhoferstr. 5, 64283 Darmstadt, Germany

{naser.damer, alexander.opel, alexander.nouak}@igd.fraunhofer.de

<http://www.igd.fraunhofer.de/en>

Abstract. This work presents a family of novel normalization techniques for score-level multi-biometric fusion. The proposed normalization is not only concerned to bring comparison scores to a common range and scale, it also focuses in bringing certain operational performance points in the distribution into alignment. The Performance Anchored Normalization (PAN) algorithms discussed here were tested on the extended Multi Modal Verification for Teleservices and Security applications database (XM2VTS) and proved to outperform conventional score normalization techniques in most tests. The tests were performed with combination fusion rules and presented as biometric verification performance measures.

1 Introduction

Normalization adjusts values or measures produced by different sources to a common scale. In multi-biometric systems, the values to be normalized are the comparison scores that describe the similarity between a captured biometric characteristic and a stored reference. Similarity scores between captured biometrics and a certain identity reference can be a result of different types of comparisons. Those comparisons can be based on different biometric characteristics, different algorithms, different captures, different sensors, or different instances of the same characteristic.

Multi-biometrics tries to use multiple biometric information sources to deal with the short comings of conventional uni-modal biometrics. Such short comings are noisy data, distinctiveness, intra-user variation, non-universality of biometric characteristics, and vulnerability to spoof attacks. Multi-biometrics aims also at enhancing the overall performance of biometric solutions.

To build a unified recognition decision from mulit-biometric sources, a fusion process is performed. The fusion can be applied on different levels of the biometric recognition work flow. It can be applied on data, feature, score, or rank-level [1].

In this work, score-level fusion is considered as it provides more flexibility to use a wide range of biometric characteristics, sensors and algorithms (with respect to data and feature-level fusion). Score-level fusion also provides more indepth information when compared to rank-level fusion.

Score-level fusion produces a biometric decision based on the comparison scores produced by different biometric sources. Those scores must be shifted and scaled to become comparable and suitable for fusion, this is performed by score normalization. Score normalization techniques largely affects the performance of multi-biometric systems [2] and thus are an active research field within the study of multi-biometrics.

Jain et al. presented a comparison between different score normalization methods for multi-biometric fusion [2]. Other works deeply discussed the effect of normalization on the biometric verification performance with a focus on methods such as zero-mean normalization (z-score), min-max, Median Absolute Deviation (MAD), TanH and decimal scaling normalization [3,4]. Some of these normalization methods will be discussed in more detail in Section 2.

In this work, a family of normalization techniques are presented and tested on the extended Multi Modal Verification for Teleservices and Security applications (XM2VTS) Score-level Fusion Benchmark Database [5]. The proposed Performance Anchored Normalization (PAN) proved to outperform conventional normalization algorithms and resulted in more accurate multi-biometric system decisions.

In the next Section 2, baseline normalization techniques are discussed and the proposed family of normalization algorithms are presented. In Section 3, the database, evaluation experiment procedure, and the acquired performance results are presented. Finally, in Section 4, a conclusion and an outlook for future work are drawn.

2 Performance Anchored Score Normalization

2.1 Baseline Normalization

As mentioned before, comparison scores processed by the fusion algorithm are usually not homogeneous as they are produced by different sources. Therefore, those scores have to be normalized before fusing. Some of the most common normalization techniques are min-max normalization, z-score normalization, tanh-estimator, and median absolute normalization [2].

The parameters that define the normalization process are usually determined based on the statistical properties of the training data. The performances of normalization techniques are not directly comparable as they depend on the overall multi-biometric system. Therefore, the normalization performance is measured based on the performance of the multi-biometric system. In the following, more details are presented on the baseline normalization techniques used in the later discussed experiment (Section 3).

Given the development comparison scores set S_k , $k = 1, 2, \dots, N$, for each biometric source, the normalized score is defined as a function of the score $f(S)$. The min-max normalization does not depend on the distribution properties and it considers only the range of the scores. This aims at mapping the raw scores into the range of [0,1]. The normalized score by min-max normalization is given by:

$$f_{min-max}(S) = \frac{S - \min\{S_k\}}{\max\{S_k\} - \min\{S_k\}} \quad (1)$$

The min-max normalization scheme is highly sensitive to outliers [2] as it depends on single minimum and maximum values.

Z-score normalization uses the arithmetic mean (μ) and the standard deviation (σ) of the development score set as parameters. This method assumes a Gaussian distribution of the score values. The z-score normalization showed good performance in many studies [6,7]. The normalized score by z-score normalization is given by:

$$f_{z-score}(S) = \frac{S - \mu}{\sigma} \quad (2)$$

The median absolute deviation normalization (MAD) method uses the median and median absolute deviation instead of the mean and standard deviation used in z-score normalization. This method also assumes a near Gaussian distribution of the comparison score values but is more robust to outliers. The median absolute deviation normalization is given by:

$$f_{MAD}(S) = \frac{(S - median)}{MAD} \quad (3)$$

$$MAD = median(|\{S_k\} - median|) \quad (4)$$

The last normalization baseline algorithm used here is the TanH normalization [2]. The TanH normalization is formulated as:

$$f_{TanH}(S) = 0.5\{\tanh(0.01(\frac{S - \mu_G}{\sigma_G})) + 1\} \quad (5)$$

where μ_G and σ_G are the mean and standard deviation of the genuine scores distribution (of training data).

The selection of a proper normalization method is a tradeoff between efficiency and robustness. Here, only normalization methods that require no special parameter tuning were considered. Methods like double sigmoid normalization [8] and TanH normalization based on Hampel estimators [9] require the fine tuning of certain parameters, while in this work the focus is on normalization methods that depends only on parameters that can be simply acquired from the development data statistics.

2.2 Performance Anchored Score Normalization

The proposed normalization techniques do not only consider the range and scale of comparison scores, they also align the scores of different biometric sources with respect to a certain performance operation point. The score value that aligns the score distributions is called the anchor, hence the notion of performance anchored normalization (PAN). This anchor score value is obtained based on

the statistics of the development data. The anchor value considered in this work is the score threshold value at the equal error rate (*EER*) operation point, and is noted here by TH_{EER} . Applying this threshold value to separate genuine and imposter scores produces an equal false acceptance rate (*FAR*) and false rejection rate (*FRR*).

This operating point alignment aims at the alienation of the undesired weight-like effect embedded in score value distributions, even after conventional normalization (e.g. min-max normalization). For example, if two comparison score sources have score distributions in the same range. However, one of the sources have a distribution that is more shifted to the high value side of the range, the scores of this source will have a higher weight when used with combination fusion rules. This higher weight effect might be an incorrect assumption by the system and leads to a lower performance.

The first PAN technique presented here is the PAN-min-max normalization. Here, the min-max normalization is extended by anchoring the middle point of the score range at the EER operation point TH_{EER} . The PAN-min-max normalized score $f(S)$ is given by:

$$f_{PAN-min-max}(S) = \begin{cases} \frac{S - \min\{S_k\}}{2(TH_{EER}\{S_k\} - \min\{S_k\})} & \text{if } S \leq TH_{EER} \\ 0.5 + \frac{S - TH_{EER}\{S_k\}}{\max\{S_k\} - TH_{EER}\{S_k\}} & \text{if } S > TH_{EER} \end{cases} \quad (6)$$

A modified MAD normalization is also presented here. The PAN-MAD normalization considers the anchor value (TH_{EER}) as pivot value instead of the median of development scores used in the conventional MAD normalization. the PAN-MAD normalization is given by:

$$f_{PAN-MAD}(S) = \frac{(S - TH_{EER})}{MAD_{PAN}} \quad (7)$$

where the MAD_{PAN} is formulated as:

$$MAD_{PAN} = \text{median}(|\{S_k\} - TH_{EER}|) \quad (8)$$

As for the PAN-MAD normalization, a performance anchored version of TanH normalization is presented. The PAN-TanH normalization here considers the TH_{EER} as anchore score value and is given by:

$$f_{PAN-TanH}(S) = 0.5 \{ \tanh(0.01(\frac{S - TH_{EER}}{\sigma_G})) + 1 \} \quad (9)$$

where σ_G is the standard deviation of the genuine scores distribution.

In Figure 1, the distribution of normalized genuine and imposter evaluation comparison scores resulted from different normalizers are shown. Comparisons from two sources are visualized, the face matcher based on Discrete Cosine Transform and Gaussian Mixture Model (DCTb-GMM) and the voice matcher based on the Linear Filter-bank Cepstral Coefficient and Gaussian Mixture Model (LFCC-GMM) matchers of the XM2VTS LP1 database [5]. It can be noticed

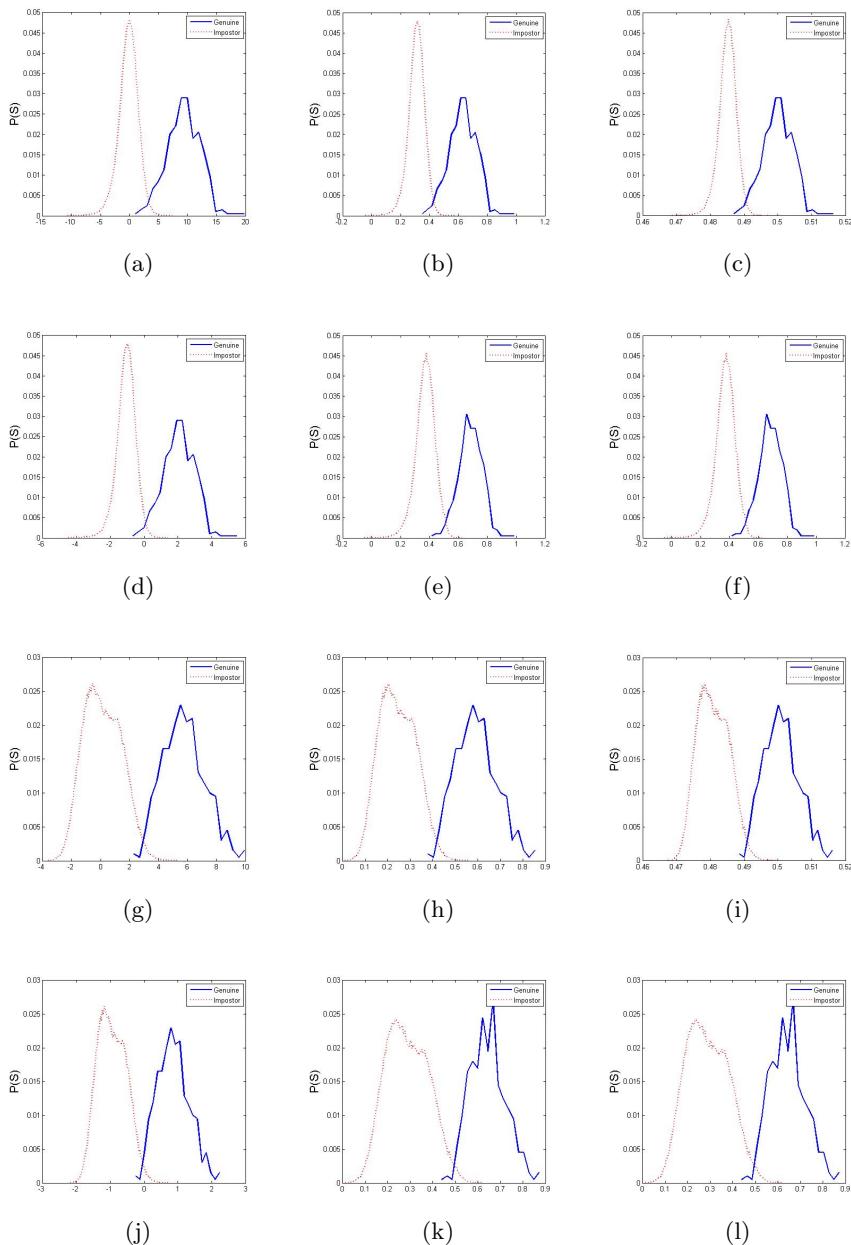


Fig. 1. Genuine and Imposter score distribution of normalized evaluation scores for face (DCTb-GMM) and voice (LFCC-GMM) [5] using different normalizers. a) face - MAD, b) face - Min-Max, c) face - TanH, d) face - PAN-MAD, e) face - PAN-Min-Max, f) face - PAN-TanH, g) voice - MAD, h) voice - Min-Max, i) voice - TanH, j) voice - PAN-MAD, k) voice - PAN-Min-Max, l) voice - PAN-TanH.

that the PAN normalization techniques did align the score distribution of both biometric sources at the anchor value TH_{EER} , the equal error rate occurs inside the area of overlap between genuine and imposter distributions. This alignment can be noticed in the face-voice distribution pairs normalized by PAN methods (d and j, e and k, f and l) in Figure 1.

3 Experiments and Results

For the development and evaluation of the proposed solution, two parts of the XM2VTS multi-biometric database were used, LP1 and LP2 [5]. LP1 and LP2 contain comparison scores by different face and voice baseline experts. The score sets are split into evaluation and development sets. LP1 contains eight score sources (5 face experts and 3 voice experts) while LP2 contains five sources (2 face experts and 3 voice experts). The major difference is that LP1 used three training captures per client while LP2 used four. For more details about the XM2VTS score database, one can refer to the work of Poh et al. [5].

Score normalization parameters for each biometric source and for both LP1 and LP2 parts of the database were obtained from the development data. Normalization parameters were acquired for the four baseline normalization (z-score, min-max, TanH and MAD) and the three proposed PAN normalization techniques (PAN-min-max, PAN-MAD, PAN-TanH). Normalization was performed on the evaluation data using all the seven techniques producing seven normalized evaluation sets.

After normalization, combination fusion rules were used to fuse the normalized comparison similarity scores. Three combination rules were used to evaluate the different normalization algorithms. The three combination rules used are the min rule, max rule, and sum rule. Assuming the combined score $C(S)$ is a function of the N biometric scores, the combination rules can be formulated as follows:

$$\text{The sum rule: } C(S) = \sum_{i=1}^N S_i \quad (10)$$

$$\text{The min rule: } C(S) = \min_i S_i \quad (11)$$

$$\text{The max rule: } C(S) = \max_i S_i \quad (12)$$

Many works discussed the performance of different combination rules, with the Sum rule usually producing higher performance [10]. However, the performance of combination rules depends on the normalization used. Different pairing between combination rules and normalization techniques produces varying results [2].

The performance of the normalization techniques were compared by considering the multi-biometric fusion results under verification scenario. The performance was reported as equal error rate (EER) values for each normalization and fusion combination rule pairing. Equal error rate is the common value of

Table 1. Equal Error Rates (EER), given as percentage, achieved by the different normalization and combination rules applied on the XM2VTS - LP1 evaluation database scores

XM2VTS - LP1			
	Min	Max	SUM
MAD	1.7797	1.7856	1.7484
MinMax	2.2538	1.0081	0.5425
Zscore	2.2994	1.3419	0.4973
TanH	1.7471	0.9996	0.5022
PAN-TanH	2.0031	1.2502	0.4960
PAN-MinMax	2.7359	1.7256	0.5031
PAN-Mad	1.8012	1.3137	0.7019

Table 2. Equal Error Rates (EER), given as percentage, achieved by the different normalization and combination rules applied on the XM2VTS - LP2 evaluation database scores

XM2VTS - LP2			
	Min	Max	SUM
MAD	1.4472	1.7341	1.6925
MinMax	1.6992	0.7502	0.2525
Zscore	1.4897	0.6930	0.2511
TanH	1.7279	0.5004	0.1952
PAN-TanH	1.2556	0.7153	0.1943
PAN-MinMax	1.2516	0.5483	0.2511
PAN-Mad	1.2663	0.7507	0.2610

the false acceptance rate (*FAR*) and the false rejection rate (*FRR*) at the operation point where *FAR* equals *FRR*. Lower EER values points out better verification performance. The EER values achieved on the evaluation sets of the XM2VTS LP1 and LP2 databases are show in Tables 1 and 2.

From Tables 1 and 2 it can be noticed that the sum combination rule outperformed the min and max combination rules in most tests. The PAN-MAD normalization achieved significantly higher performance when compared to conventional MAD normalization. Under sum rule, the PAN-MAD normalization achieved 0.70% and 0.26% EER with respect to 1.75% and 1.69% EER achieved by MAD normalization for the LP1 and LP2 parts of the database.

The best EER values for min-max and PAN-min-max normalization were achieved under the sum combination rule with the PAN-min-max normalization outperforming the conventional min-max normalization on both databases. The proposed PAN-TanH normalization achieved the highest performance on both databases with lower EER values than the conventional TanH normalization.

4 Conclusion and Future Work

This work presented novel score normalization techniques for multi-biometric score-level fusion. The techniques presented here focus on the alignment of certain operation point in the score distribution of different biometric sources.

The discussed performance anchored normalization methods proved to outperform their conventional counterparts. The evaluation of the normalization techniques was performed under multi-biometric verification scenario and was based on the XM2VTS multi-biometric database. The performance was presented in the form of equal error rate values of different normalization-fusion combinations.

Planned future work should investigate the effect of the PAN on the performance of more sophisticated fusion techniques as well as the multi-biometric identification scenario.

Acknowledgments. The work leading to these results has received funding from the European Community's Framework Programme (FP7/2007-2013) under grant agreement n° 284862.

References

1. Damer, N., Opel, A., Shahverdyan, A.: An Overview on Multi-biometric Score-level Fusion: Verification and Identification. In: Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods, pp. 647–653. SciTePress, Lynnfield (2013)
2. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. *Pattern Recognition* 38, 2270–2285 (2005)
3. Snelick, R., Uludag, U., Mink, A., Indovina, M., Jain, A.: Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 450–455 (2005)
4. Singh, Y.N., Gupta, P.: Quantitative evaluation of normalization techniques of matching scores in multimodal biometric systems. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 574–583. Springer, Heidelberg (2007)
5. Poh, N., Bengio, S.: Database, protocol and tools for evaluating score-level fusion algorithms in biometric authentication. *Idiap-RR* Idiap-RR-44-2004, IDIAP (2004)
6. Srinivas, N., Veeramachaneni, K., Osadciw, L.: Fusing correlated data from multiple classifiers for improved biometric verification. In: 12th International Conference on Information Fusion, FUSION 2009, pp. 1504–1511 (2009)
7. Vajaria, H., Islam, T., Mohanty, P., Sarkar, S., Sankar, R., Kasturi, R.: Evaluation and analysis of a face and voice outdoor multi-biometric system. *Pattern Recogn. Lett.* 28, 1572–1580 (2007)
8. Cappelli, R., Maio, D., Maltoni, D.: Combining fingerprint classifiers. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 351–361. Springer, Heidelberg (2000)
9. Hampel, F., Ronchetti, E., Rousseeuw, P., Stahel, W.: Robust statistics: the approach based on influence functions. Wiley series in probability and mathematical statistics. Wiley (2005)
10. Chang, K.I., Bowyer, K.W., Flynn, P.J., Chen, X.: Multi-biometrics using facial appearance, shape and temperature. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, FGR 2004, pp. 43–48. IEEE Computer Society, Washington, DC (2004)

Towards a Contextualized Visual Analysis of Heterogeneous Manufacturing Data

Mario Aehnelt¹, Hans-Jörg Schulz², and Bodo Urban¹

¹ Fraunhofer IGD Rostock, Germany

{mario.aehnelt,bodo.urban}@igd-r.fraunhofer.de

² University of Rostock, Germany

hjschulz@informatik.uni-rostock.de

Abstract. Visual analysis spanning multiple data sources usually requires the integration of multiple specialized applications to handle their heterogeneity. This is also true in manufacturing, where data about orders, personnel, workloads, maintenance, etc. must be analyzed together to make well-founded management decisions. Yet, the orchestration of multiple data sources and applications poses challenges to the software infrastructure and to the analyst. We present a three-tiered approach to cope with these challenges. In a first step, we establish a domain-dependent workflow as the mental model of the analyst. Based on the novel concept of contextualization, we then align the different applications with this model for their meaningful integration. In a third step, we incorporate the data according to its use in the aligned applications by means of a service-based architecture. By starting the integration on the user level, we are able to pragmatically target and streamline the required integration to a degree that is technically achievable and interactively manageable. We exemplify our approach with the Plant@Hand system for integrating manufacturing data and applications.

1 Introduction

Heterogeneous data, meaning data that stems from various data sources and that comes in a multiplicity of data formats, is far from a purely academic research challenge. Instead, it is an equally exciting and frustrating reality in many application domains. The excitement about heterogeneous data stems from the fact that bringing together diverse data sources for an integrated analysis and decision making yields more reliable and comprehensive insights than investigating individual data sources alone. Whereas the frustration stems from the realization that it is extremely hard to actually do just that – to “bring together” diverse data sources. One of the reasons for this dilemma is that there exist highly specialized domain-dependent visual and non-visual analysis tools that allow users to perform a certain set of operations for a particular kind of data. The domain of manufacturing is no exception to that, as for example, *enterprise resource planning systems* (ERP) are used to manage information about orders and personnel, while *manufacturing execution systems* (MES) are

employed to collect and evaluate data about the production process. For an integrated analysis, not only the different data sources must be used in combination, but also the various applications that are used to access them. This results not only in a technical challenge, but also in a challenge to the human analyst who must interactively manage this diversity of data and tools to pursue a cross-dataset/cross-application analysis.

Thus, our work aims to combine multiple such applications in a way that allows for a fluid integrated analysis across applications and thus across the different data sources they operate on. To achieve this with reasonable effort on the technical level and on the user level, we concentrate the integration on only the necessary aspects by following a three-step approach:

1. We investigate the domain context and establish a **general workflow**, which can be assumed as a mental model of an analyst in this field.
2. We link the applications to the workflow through **contextualization**, which anchors the tools in the spot where they are used – e.g., the spreadsheet with machine data right on top of the CAD drawing of the machine itself.
3. We perform the **data integration** using *enterprise service bus* (ESB) technologies according to the contextualization of the applications, which gives us the knowledge about how they feed data into each other.

The remainder of this paper details the related work in Section 2. Based on that, it introduces our 3-step approach of workflow generation, application contextualization, and data integration in Section 3. This approach is then illustrated by the *Plant@Hand* system, a domain-specific implementation of our concepts for the concrete case of manufacturing in Section 4. Section 5 concludes this paper and states our ideas for future work.

2 Related Work

For the challenge of integrating heterogeneous data, various solutions already exist. They focus on different aspects, addressing either the user level, the application level, or the data level.

On the **user level**, a fundamental problem of the multitude of data sources is the *information overload* challenging the user's cognitive capabilities. This problem aggravates the more data sources and thus the more applications handling them need to be integrated. Only recently, Landesberger et al. [1] identified the lack of *consistency* as one of the reasons for this, making it a crucial design problem for integrated analysis applications. On the one hand, this problem can be addressed by using the same basic visualization and interaction means, e.g., from a standard library, if the heterogeneity of data allows for it. On the other hand, consistency is also a question of the user's perception. Here we learn from research in *computer-supported cooperative work* (CSCW) that the user's mental model plays a growing role in designing interactive applications [2]. Each user has an individual small-scale model of certain aspects of reality, which influences his interaction with an application. This concept of *mental models* reaches back

to 1943, when Kenneth Craik discussed the influence of human thinking on the perception of reality [3]. A common way to align multiple analysis applications and datasets with a mental model is to use the analysis workflow to be performed on the various datasets with the different analysis tools. For example, Streit et al. [4] align their visual representation of data sources with such a workflow to clearly convey what the users are looking at and how it relates to other data.

On the **application level**, bringing together different data sources by combining the visualizations and Visual Analytics (VA) applications, which are used to show and analyze them, is so far an unsolved engineering problem. The most prominent endeavor in this direction is the *Obvious* framework [5], which aims to define an interface standard for VA applications. Other approaches range from highly centralized architectures to highly flexible ones. An example for the former is the *Universal Visualization Platform* [6] that provides one core framework to which all applications are hooked via a plug-in mechanism. An example for the latter is the *Metadata Mapper* [7] that realizes a loose coupling of applications through a common communication bus. Furthermore, it is also possible to integrate applications based on their user interfaces, rather than their data or metadata. Various such approaches exist, such as, *virtualization* described by Besacier and Vernier [8], the *mash-ups* by Aehnelt [9], or *customized graphical user interfaces* introduced by Lee et al. [10].

On the **data level**, the integration of various data sources is a longstanding challenge in database research. Over the years, many different solutions have been developed to achieve such integration. Depending on the degree of integration, the developed solutions range from *federated databases* [11] for a rather loose linkage of independent data sources to *information fusion* [12] and *data warehouses* [13]. In addition, there exist also a variety of ways to perform the actual linking of data – e.g., based on *conceptual schemas* [14], based on logic [15], or based on ontologies [16].

It is noteworthy, that none of these approaches targets the integration of data sources on all three levels at the same time. Yet, this would be required to achieve a fluent integration that is easily understood and managed by a human analyst through the various applications he uses. This is where our solution comes in, which considers each of these levels in one of its steps, as it is described in the following section.

3 Integrating Heterogeneous Data and Analysis Tools

As opposed to other approaches for integrating heterogeneous data, we do not start on a technical data or application level, but on the user level to gain an understanding of the actually required integration before performing it. This permits us to anticipate and preserve domain-specific relations between data sources, while omitting others that are not relevant for the case at hand. Being confronted only with the amount of integration complexity that is necessary makes the integration manageable and understandable by the analyst and at the same time also technically achievable from an architectural standpoint. On top

of that, the firm grounding of the application and data integration in the domain creates an extra level of application consistency, as data and applications behave according to the analyst's own understanding of his work reality. The three steps of our integration approach are detailed in the following.

3.1 Understanding the General Workflow of the Application Domain

Initially, we have to understand the domain-specific application context and based on this the likely mental model of a data analyst. One possible context to establish is the analysis procedure that the analyst follows. Although, individual procedures can deviate in detail, a generalized workflow model can be used to capture it abstractly. Figure 1 shows such a generalized workflow that aims to model the data analysis process in manufacturing. In essence, the monitoring and control of manufacturing processes are mainly a periodical review of planned and reported figures. Deviations are to be further investigated in order to identify upcoming problems and implement adequate solutions promptly. On management level, *key performance indicators* (e.g., overall equipment effectiveness, cycle-time-ratio, delivery delay) give an overall impression of manufacturing performance. On operational level, the detailed production or assembly status in relation to production and work orders as well as resources are crucial for further decision making.

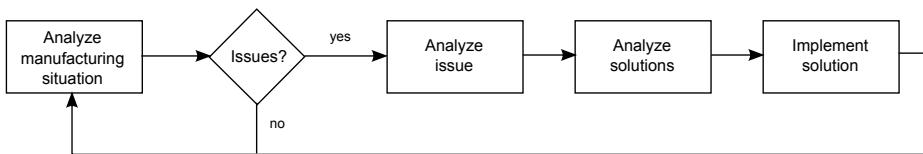


Fig. 1. Workflow model of data analysis and decision making in manufacturing monitoring and control

The production supervisor – being the analyst of various manufacturing data – is responsible for a timely and quality-conscious delivery of products. In each analysis step, he works with different data and varying analysis tools. When analyzing the manufacturing situation, he compares planned figures against reported ones, e.g., the sequence of *production orders* (times, delays, outputs), the allocation of *resources* (material, machines, staff), or the *workload* on production work places (efficiency of machines and staff).

The data and their visualizations are in this case provided by ERP systems and MES. They are rarely connected, which complicates the joint analysis of data from both systems. When it comes to unplanned interruptions, e.g., machine breakdowns, missing part orders, or industrial injuries, again other tools are needed to view and analyze this particular data. Here, a growing amount of unstructured data (sales order details, supplier part catalogs, part specifications,

machine manuals) has to be reviewed in close combination with structured data (contact details, production orders).

When using domain-specific analysis tools, users are used to familiar products and their representation of data. The manufacturing domain works with standard business applications that form the user's understanding and interpretation of data, e.g., *SAP Business One PPS*, *HYDRA-MES*, and *product data management* (PDM) products, such as *Autodesk AutoCAD* or *Dassault SolidWorks*. This often leads to a dual encoding of information that influences the user's own mental model of data, e.g., *parts* have a recognizable material number in ERP and MES, and at the same time a visual representative in form of a geometric CAD model. Thus, the production supervisor has both visual and abstract representatives of data in mind when analyzing bills of material. But, he only recognizes that the same part has been used multiple times in a single product based on its spatial location in the model and not by its serial number. This context knowledge facilitates the integration of the different applications to seamlessly perform the outlined general analysis workflow across application boundaries.

3.2 Contextualizing Visualization and VA Applications

From reviewing the application domain, we learned about the goals and structure of analysis work steps, as well as the data sources and the applications through which they are accessed and analyzed. Using a process called *contextualization*, this diverse information is then combined on the application level to form an ensemble of tools that reflects the assumed mental map of the analyst. Contextualization is a novel approach in Visual Analytics that enriches purely technical entities, such as software applications and data sources, with the context knowledge about their relations with each other, as they are derived from the application domain in the first step. In its simplest form, it can be used to capture the identity relations of the aforementioned dual encodings and thus to link their respective applications.

Yet, context knowledge can also be understood as a hierarchical construct which nests smaller sub-contexts (e.g., a particular machine) in a broader super-context (e.g., a factory building). The more confined the current context is (e.g., analysis of a single machine instead of the entire factory), the more specific is our knowledge and thus the easier it gets to derive a suitable application ensemble. Yet, as the context is reduced, the fewer data sources and data types still relate to that context. For example, work progress data and workforce data cannot be linked contextually to an individual machine, even though a drop in work progress may have a causal relation to a single broken machine. The same issue occurs for abstract data with no relation to the current sub-context at all. In these cases, we have to look on higher contextual levels for relations between such data. Thus, for example, workforce data (e.g., age or skills) refer to manufacturing teams responsible for specific tasks on a construction part. The abstract data of one sub-context can then be visualized close to the visualization

of related data of another sub-context utilizing the relationship between both on a higher contextual level.

Arranging the application ensemble in this way aids the analyst's recognition and anticipation of known correlations, patterns, and representatives of reality. Basically, the data is shown in its associated applications where it is expected (e.g., a machine data visualization on top of the CAD drawing of that machine) and where it is needed according to the workflow. For this, Aehnelt et al. [2] already identified *context objects, metaphors and analogies*, as well as *interaction scripts* as suitable means to encode information and interact with it according to the user's mental model. This allows the user a clear interpretation and improves the efficiency of data analysis. For example, when the production supervisor is analyzing the planned and reported figures, he aims to identify deviations in time, quantity, and quality. In this example, the analysis application needs to visualize and highlight critical deviations using a visual encoding that matches the user's mental model. To also make the underlying context accessible and visible, the initial ensemble of overlaid views can be interactively rearranged by the user at any time during the analysis. In order to not overburden the application ensemble with views, *semantic and visual linking* [17] can be employed to shift loosely related applications to the periphery while at the same time maintaining and communicating their relations.

This approach of contextualization allows us to visually combine the familiar ERP, MES, and CAD representations embedded into a consistent user interface which adopts the virtualization or composition of analysis tools within a more generic framework. This framework applies the contextualization rules automatically, thus influencing the presentation and interaction with data according to the given domain knowledge, such as the currently pursued workflow and the available data sources.

3.3 Data Integration with Enterprise Service Bus Architectures

Having established the application ensemble, we now know which data sources must be integrated to facilitate the analysis across application boundaries. In particular, data that is connected via identity relations (dual encodings) and data that is connected by referring to the same context (i.e., the same machine) are likely to be used in concert and should thus be linked on data level as well. To this end, we use ESB technologies that allow a flexible configuration and transfer of data between heterogeneous interfaces, data models, and applications. Within the ESB, an integrated data model derived from our workflow model (Step 1) maps the domain specific data types onto an abstract inter-application model (Step 2) of data. *Contextualized network graphs* [18] help us to automatically find relations and fill the integrated model with heterogeneous data. Thus, having streamlined the data integration to the concrete case at hand, we are able to concentrate on linking only the necessary data: work orders and resource allocations from ERP, time reports from MES, and 2D/3D product models from PDM with unstructured data from document sources.

4 Plant@Hand

Our approach of contextualizing the visual analysis of manufacturing data was implemented in the *Plant@Hand* system. It specifically addresses the monitoring and control of manufacturing and assembly works in shipbuilding industries. The system integrates different data analysis functionalities as they are usually provided by ERP, MES, and PDM systems individually within a consistent multi-touch user interface. *Plant@Hand* supports multiple users working collaboratively with different contextualized data representations. The three steps of our approach are realized in this system as follows.

4.1 Analysis Workflow in Plant@Hand

The monitoring and control workflow (see Figure 1) addresses a supervisor's procedure of analyzing progress and figures at a shipbuilding site. Thus, we adopt his own mental model to design our integrated analysis application accordingly.

The analysis application *Plant@Hand* starts with an overview of the manufacturing and assembly situation combining the data from all required sub-contexts in representative views. Issues are immediately highlighted to simplify the first workflow step. The supervisor and his team can explore all available data for each issue individually using data specific views, e.g., work reports, part models, time schedules. Through interacting with these views, the next workflow step of analyzing adequate solutions is supported. By this means, the team can modify workplan data and get instant feedback on possible resource gaps or time conflicts. Once the supervisor has decided for a solution, it is implemented, meaning that data modifications are passed on to the responsible manufacturing software (ERP, MES, PDM) and its solution-dependent business logic.

4.2 Contextualized Data Representation and Interaction

The main contextualization is based on the monitoring and control workflow as well. All required data is provided in separated views using familiar and established representatives for data.

The application is built on top of a visual representation of the main construction plan containing a 2D drawing of ship sections and installation details. Each user is then visually represented by an individual task view containing his own work orders and details. Further views visualize planned work orders and schedules, reported work results and issues, 3D construction models, and assembly tutorials. In our application, we additionally use a *spatial* and *temporal* contextualization of all available information to give any data a visual relation to product models and time schedules. Thus, analyzing a specific work order highlights its visual representatives in other views, e.g., the corresponding geometric section in the construction plan, the work orders' time schedule and staff allocation, or related work reports and issues (see Figure 2). This linking references the user's mental model of where the work is located, when it is due, which people are assigned to it, and which results were already reported.



Fig. 2. An assembly team working collaboratively with Plant@Hand on a multi-touch table (left). Spatial and temporal contextualization of work orders in Plant@Hand (right). An underlying CAD drawing is used to provide context to the individual views, which are positioned at the place they refer to, e.g., a particular machine.

Interaction with Plant@Hand bases on multi-touch gestures. Here we allow for data specific interactions that are based on the workflow. A re-planning of resources to solve work-related issues can be done visually by moving work orders with drag gestures between work order view and personalized task views. In a similar way the application supports an interactive visual data analysis for exploring work situations or reported problems. This requires a contextualized interaction with 2D and 3D construction models, as well as with work report documents. Geometric models can be moved using drag gestures, zoomed with pinch and spread gestures, or rotated with a two finger rotation.

To ensure consistency between views, all views are provided with additional controls and interaction mechanisms from a central library (cf. Section 2). These controls are dependent on the data type, e.g., volume and play controls for video/audio media, navigation shortcuts for time schedule view (see Figure 3), or model manipulation controls for 3D object models.

4.3 Integration of Manufacturing Data

Plant@Hand combines data from ERP, MES, and PDM systems. With an open ESB infrastructure as described in [17], we establish an integrated data layer between such applications and Plant@Hand which filters and connects required data for our own analysis purpose. Data modifications in Plant@Hand are then passed on to the ESB that translates them into application-specific data updates and requests for underlying ERP, MES, or PDM systems. The advantage of using an ESB as transparent data layer between analysis application and heterogeneous data sources becomes evident through its highly configurable infrastructure. It allows us to flexibly model the dependencies between data and data related information flows and thus to configure the data integration to closely resemble the data flow between the applications according to the workflow.

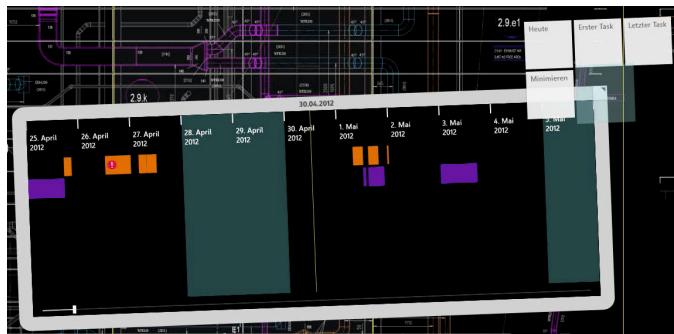


Fig. 3. Contextualized controls of time schedule view

5 Conclusion and Future Work

A first qualitative evaluation of Plant@Hand under field conditions showed positive impact of using contextualization on both efficiency and usability. We brought our Plant@Hand on-site into a manufacturing company for cooling devices and evaluated it there with assembly staff. After a short learning phase of basic touch gestures, the assembly team worked autonomously with the application. With respect to efficiency the main benefit was seen in the information integration avoiding time consuming search in paper documents or opening different applications to collect required data. The evaluation also showed a fast adoption of our provided contextualized visualizations. After working for a while with the application, the team was requesting a construction plan manipulation on design level which shows a seamless transition from monitoring and controlling towards re-planning. However, evaluating the overall benefit and performance increase in data analysis will require further long-term studies on-site.

This preliminary evaluation indicates that contextualization is a suitable methodology to design visual data analysis applications close to the user's mental model in order to improve the usability and efficiency of working with heterogeneous data. With Plant@Hand, we have illustrated that our proposed approach can be adopted to any domain that has to deal with heterogeneous data and thus various applications to access and analyze them. In our concrete case of manufacturing, the resulting integrated application ensemble exceeds the available individual analysis tools as it is able to handle data from ERP, MES, and PDM sources through a single seamless analysis interface. While Plant@Hand is already actively promoted within the industrial context on fairs and trade shows, it is subject to further research. With its successor Plant@Hand3D, we are already experimenting with the value of more realistic data and context visualizations. This includes the possibility to move individual views from the ensemble onto smaller tablet devices with which users can move around freely. It allows them to pursue individual tasks on their own at the time and place these need to be pursued, and later reintegrate separately collected data back into the ensemble.

Acknowledgements. Work on this research has been funded in part through an individual grant by the German Research Foundation (DFG).

References

1. Landesberger, T., Schreck, T., Fellner, D.W., Kohlhammer, J.: Visual search and analysis in complex information spaces. In: Expanding the Frontiers of Visual Analytics and Visualization, pp. 45–67. Springer (2012)
2. Aehnelt, M., Peter, C., Müsebeck, P.: A discussion of using mental models in assistive environments. In: Proc. of PETRA 2012 (2012)
3. Craik, K.J.W.: The nature of explanation. Cambridge University Press (1943)
4. Streit, M., Schulz, H.J., Lex, A., Schmalstieg, D., Schumann, H.: Model-driven design for the visual analysis of heterogeneous data. IEEE TVCG 18, 998–1010 (2012)
5. Fekete, J., Hemery, P., Baudel, T., Wood, J.: Obvious: A meta-toolkit to encapsulate information visualization toolkits – one toolkit to bind them all. In: Proc. of VAST 2011, pp. 91–100 (2011)
6. Gee, A.G., Li, H., Yu, M., Smrtic, M.B., Cvek, U., Goodell, H., Gupta, V., Lawrence, C., Zhou, J., Chiang, C.H., Grinstein, G.G.: Universal visualization platform. In: Proc. of VDA 2005, pp. 274–283. SPIE (2005)
7. Rogowitz, B.E., Matasci, N.: Metadata mapper: A web service for mapping data between independent visual analysis components. In: Proc. of HVEI 2011 (2011)
8. Besacier, G., Vernier, F.: Toward user interface virtualization: Legacy applications and innovative interaction systems. In: Proc. of EICS 2009, pp. 157–166. ACM (2009)
9. Aehnelt, M.: Contextualized knowledge exchange with mash-up technologies. In: Proc. of eLBa 2010, pp. 143–153. Fraunhofer IRB (2010)
10. Lee, L.C., Lutteroth, C., Weber, G.: Improving end-user GUI customization with transclusion. In: Proc. of ACSC 2010, pp. 163–172. Australian Computer Society (2010)
11. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys 22, 183–236 (1990)
12. Blasch, E., Bossé, E., Lambert, D.A. (eds.): High-Level Information Fusion Management and Systems Design. Artech House (2012)
13. Chaudhuri, S., Dayal, U.: An overview of data warehousing and olap technology. SIGMOD Record 26, 65–74 (1997)
14. Cali, A., Calvanese, D., De Giacomo, G., Lenzerini, M.: Accessing data integration systems through conceptual schemas. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 270–284. Springer, Heidelberg (2001)
15. Levy, A.Y.: Logic-based techniques in data integration. In: Minker, J. (ed.) Logic-Based Artificial Intelligence, pp. 575–595. Springer (2000)
16. Lenzerini, M.: Ontology-based data management. In: Proc. of AMW 2012. CEUR Workshop Proceedings, pp. 12–15 (2012)
17. Aehnelt, M., Bader, S., Ruscher, G., Krüger, F., Urban, B., Kirste, T.: Situation aware interaction with multi-modal business applications in smart environments. In: Yamamoto, S. (ed.) HCI 2013, Part III. LNCS, vol. 8018, pp. 413–422. Springer, Heidelberg (2013)
18. Ceglowski, M., Coburn, A., Cuadrado, J.: Semantic search of unstructured data using contextual network graphs. Preliminary white paper. National Institute for Technology and Liberal Education, Middlebury College, Middlebury, Vermont, USA (2003)

Visual Statistics Cockpits for Information Gathering in the Policy-Making Process

Dirk Burkhardt, Kawa Nazemi, Christian Stab, Martin Steiger,
Arjan Kuijper, and Jörn Kohlhammer

Fraunhofer Institute for Computer Graphics Research (IGD),
Fraunhoferstr. 5, 64283 Darmstadt, Germany

{dirk.burkhardt, kawa.nazemi, christian.stab, martin.steiger,
arjan.kuijper, joern.kohlhammer}@igd.fraunhofer.de

Abstract. A major step in ICT-driven policy making is information gathering. During this phase, analysts and experts have to deal with a high number of statistical data which they use as a basis to identify problems and find appropriate solutions. This paper introduces a statistical data model to support these analysts and experts. It allows for handling the complexity (i.e. the dimensions) of the data for the visualizations. In particular, it helps to use the same data for two-dimensional, but also multi-dimensional statistics visualizations. Based on this statistic data model we introduce an interactive approach of visual statistics cockpits. This results in highly interactive statistics visualization cockpits that enable both analysts and experts to improve problem assessment and solution finding.

1 Introduction

Nowadays, policy making is a process that is primarily performed offline, so that most of the already existing benefits of Information and Communication Technologies (ICT) are not yet considered. The biggest advantage today is that most countries are working on open-data portals to improve transparency of the government's work for the citizens. These open-data portals allow citizens to analyze the government's work through large list of indicators that are measured by public authorities. Therefore, interested individuals and organizations are developing more and more tools and sophisticated visualizations to allow a facilitated analysis, which allows citizens to make use of published data. Unfortunately, these open-data portals are not yet established in the policy making process of public authorities. Especially municipalities consider ICT and open-data today in a very limited way. Over the past years, the number of municipalities and politicians who understand the necessity to incorporate ICT in the policy making process has increased. They started to consider how ICT could be included to improve policy making process in their daily lives.

In order to develop ICT that help public authorities in the policy making, it is necessary to develop tools that help both analysts and experts to grasp the meaning of the available data. This can primarily supported through the use of adequate visualization

tools that allow for showing data in an adequate manner and enable users to get an understanding of a given problem. Furthermore, these tools need to be included in the existing policy making cycles of the public authorities. Considering adequate visualizations as well as inclusion into the given policy making processes allows for an effective policy creation. An approach that directly addresses visualization in the policy making process was published by Kohlhammer et al. [17]. They defined the most relevant visualization tasks for policy making: (1) information foraging, (2) policy design, and (3) impact analysis. They also addressed information foraging as the major task for visualization, because it includes both the problem identification and the solution finding.

Today, the general aspects of statistical data visualization are well known; there are also a couple of approaches using dashboards to allow for an improved overview of the data. A major challenge in the domain of visual analysis is the interactivity of statistics visualizations. In this paper we address this issue and introduce a novel design of a technical data model for statistical data, which is primarily designed for modern open-data portals. The main idea is to specify a data model that allows for usage in multiple types of statistics visualizations. In particular, it aims to bridge the gap of coupling two- to multi-dimensional statistics visualization with only one data model that holds all available data. Based on this data model, we outline an approach of visual statistics cockpits for interactive analysis. The main benefit of the technical statistics data model and the statistics cockpit approach is the support of analysts and domain experts in the policy making process. This is essential in the information gathering phase to identify problems and solutions just through analyzing the data.

2 Related Work

Interaction with statistical visualizations has been investigated since a couple of decades. It is one of the best researched topics in information visualizations sciences. The focus was mostly on how statistical data should be communicated in a graphical way, and how it will then be easier understood by the target users. For these target users a couple of simple but also expert visualization were developed.

The major challenge of using statistical visualizations in the domain of policy modeling is not the use of statistical visualizations in general, but it is the practical integration of appropriate visualizations with respect to the task, the data, and the user [18]. Especially the user and his level of expertise define whether the visualization is useful or rather confusing. Therefore, we give an overview of existing visualization approaches in the first part of this section and define for which user-type they will be adequate. In the second part of this section we introduce existing approaches of dashboards, which follow the idea of coupling multiple visualizations to provide a better overview. The disadvantage of single visualizations is that they are not appropriate for all analysis works and therefore different sets of visualizations can be beneficial for the user. In this section we introduce some examples of dashboards and describe their benefits for the analysis.

2.1 Statistics Visualization Approaches

Today, a large number of statistical visualizations exist, but most of them can be categorized into only four groups [4]. Visualizations offer their advantages in regard to the expected use-case. Consequently, there is no visualization that will be beneficial for all kinds of analysis work. Any visualization provides a faceted view on the data. Statistical data can be considered as lists or tables of nominal or ordinal data [4]. Additionally, the benefit of visualizations also depends on the user's behavior and his expertise in using visualization systems. An analyst can use visualizations which can show multidimensional data within complex visualizations, i.e. Pixel Matrix Displays [1] or Parallel Coordinates [2, 3]. In fact, this means that any visualization has to be selected in dependence on the use-case and the tasks that should be solved with it.

In the first category, the *Point-based methods* [4], visualizations make use of points, marks, or other symbolic signs for representing a data record. According to certain attributes, the visual representations of each record are placed on the screen to derive a visual representation of the whole data set. Depending on the selected attributes and the chosen layout method, point-based techniques are suitable to compare certain data characteristics, to identify outliers or irregularities in the data, to recognize relationships among data entities, and to identify unexpected or previously unknown clusters and patterns. Usually, each data record is projected from its n-dimensional space to a (lower) k-dimensional space (usually two- or three-dimensional) and the visual representation of the record is represented at the k-dimensional point on the screen. Example visualization representatives are (Parallel) Scatterplots [5] and (vectorized) RadViz [6]. Point-based visualizations, especially the more abstracted visualizations like the RadViz, are most suitable for advanced users. Next to the placement of the point in the coordinate system, also the shape, size, or color imply information and this metaphor is usually not easy to understand by non-advanced users.

Line-based techniques form the second category for visualizing statistical and multivariate data. They are often used to analyze financial or temporal data. Due to the high familiarity of the respective users, line-based techniques are an effective tool for common users to analyze and explore statistical data. In contrast to point-based methods that represent each data record as a symbolic representation, line-based visualizations are based on the idea of representing the values of a data record or dimension linked together in a straight or curved line. Thus, each line in a line-based visualization represents perceivable features of the given record. Consequently, these techniques are well-suited for identifying slopes, curvatures, and even crossings among multiple records. Typical representatives of line-based visualizations for statistical data and their different modes and goals are presented in the following subsections. The most established and best known representative for this kind of visualization are Line Charts. Line Charts are adequate for casual as well as advanced and expert users. They present two-dimensional data in an easily understandable form. However, other visualizations are rather designed for experts, like Parallel Coordinates [2,3] and Radial Visualization [7].

The third category, named *Region-based visualization techniques*, is used for multivariate data. They are often used to analyze financial data, to compare actual and target status, or to explore differences in experimental data. The basic idea of region-based visualization techniques is to represent data records as filled polygons or regions on the screen. Usually instantiations of region-based techniques incorporate different properties of the given data into the visual design of the polygons to convey additional values and data characteristics to offer the possibility of comparing different features of the data. For instance, the size, the shape, or the color of the visual representation of a data record can be utilized for visually representing additional dimensions of the data set. Due to the ability of the human perception – which enables an effective differentiation of the length or the size of presented polygons – region-based visualizations are successfully applied for representing and analyzing different quantitative information encoded in the data. Typical representatives of region-based visualizations include bar charts, tabular displays like heat-maps [8], and table lenses [9]. These kinds of visualization are primarily designed for advanced users. Implementations that can be used by casual users or – for data with a higher level of complexity – by expert users, exist as well.

The forth category is the *Combination of Techniques*. Point-based, line-based, as well as region based techniques utilize a specific graphical metaphor to visualize statistical or multivariate data sets. Hybrid visualization techniques incorporate several techniques of the methods presented in previous sections for obtaining a meaningful representation of the given data. Typical representatives of hybrid visualization techniques for multivariate data are Dense Pixel Displays[10] and Theme River [11]. Because of the combination of different approaches, most of the visualizations become so complex that the target users are usually advanced and expert users.

2.2 Statistics Dashboards

The use of dashboards is a common approach to reduce the problem of insufficient visualization for a certain task. Because of the combination of different visualizations, it can be ensured that at least one adequate visualization is shown to the user. Furthermore, the use of multiple visualizations allows showing the data from different perspectives, and thus providing an adequate overview.

The policy making process, especially the analysis phases, counts as a very heterogeneous task. Depending on the goal of the analysis the number of data elements that has to be analyzed varies. This comes along with strong deviations in the data dimensionality, and the resulting complexity in the visualizations. The more indicators need be conducted, the more complex the visualizations become. The use of dashboards can help to reduce the complexity, as next to detailed visualization also a very abstract overview visualization can be provided. This allows a top-down analysis (similar to Shneiderman's information seeking mantra [12]) and acts as a low-barrier entrance into the data analysis.

In the policy modeling domain two well-known dashboard approaches exist that make use of a combined set of visualizations. The *OECD eXplorer*¹ [13, 19] allows for analyzing various statistical data from OECD countries. It therefore provides a set of visualizations to show indicators, such as the GDP, population development, and education. For this purpose, the OECD Explorer uses next to traditional chart visualization also some modern visualization techniques like scatter plots, parallel coordinates, and colored geographical visualizations. The dashboard consists of an orchestration of visualizations. Only the statistical visualization can be replaced with a single other visualization. Thus, the OECD Explorer provides a first interactive approach to analyze statistical data, but it is very restricted in the number and the type of available visualizations.

The second dashboard is the *EZB Inflation Dashboard*², which allows for an analysis of inflations of different European countries. It provides a dashboard consisting of three common chart visualizations and a map visualization of Europe. With this dashboard the user can analyze and compare the inflations in the European countries in an interactive manner. To make detailed analysis, any visualization can be switched to full screen.

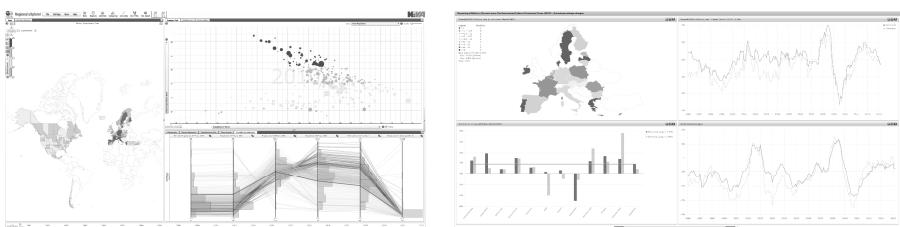


Fig. 1. On the left a screenshot of the OECD eXplorer is shown with its different visualizations. On the right a screenshot of the EZB Inflation Dashboard is shown, with its four fix-defined visualizations orchestration.

To provide interactivity it is necessary to provide the user visualizations that allow for an in-depth analysis, but also to allow interactivity through a dashboard metaphor. Traditional chart visualizations are well-known and use simple metaphors, but they are limited in single-use scenarios regarding interactivity. Modern visualizations, like parallel coordinates, are more complex, but allow for an improved in-depth analysis. Consequently, it seems beneficial to provide an analysis visualization system that contains a mixed set of traditional chart visualization, as well as modern analysis visualizations. To enhance this general approach, it also seems to be beneficial, when the visualizations are presented with a dashboard metaphor. Here the user can choose the appropriate visualization for his work. A major benefit of a dashboard with coupled visualizations is the higher interactivity. This metaphor should be used and

¹ Available on: <http://stats.oecd.org/OECDregionalstatistics/> (accessed on 29/04/2013)

² Available on: <http://www.ecb.europa.eu/stats/prices/hicp/html/inflation.en.html> (accessed on 29/04/2013)

extended through a higher flexibility. The described approaches use static orchestrated visualizations. From the interactivity perspective it seems beneficial, when the cockpit can be completely orchestrated by the user depending on his behavior and work task. Therefore however, a concept is required that allows for the unification of the existing two- to multidimensional data in a single technical data model. If all kinds of visualization are able to handle the same technical data model with the statistic information, a flexible cockpit can be designed.

3 Concept for Visual Statistics Cockpits

The design of a statistics visualization dashboard that should include various types of static visualizations, consisting of two- and multidimensional visualizations needs to introduce a concept to handle the data. Hereby we have to deal with the existing open-data portals and how this data can be used in these two- to multidimensional visualizations.

Existing open-data portals, e.g. EuroStat, GOVDATA, Data.gov and Data.gov.uk, mostly structure the data into a hierarchy of topics. For each topic a couple of so-called indicators are aligned. Furthermore, each indicator is defined by:

- The name of the indicator, e.g. GDP, public growth, or public density.
- An assignment to a geographical region, i.e. a country, state/province, municipality, or city.
- A time-based data table, which consist of the indicator value by the measured time.
- Optional additional meta-information about the indicator, such as a description of influencing indicators or the used unit.

Based on these given structures, a technical data model that is adequate for the different two- and multidimensional statistics visualizations needs to be designed. Additionally, the data model needs to support interactive approaches, such as the option to link visualizations and to allow a further exploration through the data.

3.1 Concept for a Statistic Data Model

The technical data model is a basis that allows an interactive analysis. Therefore, it must be ensured that it provides the ability to navigate through the data with all kinds of visual statistic representations.

Based on the already mentioned structure of the existing open-data platform, we identified the relevant parameters that can be changed in the visualization process to explore the data. A change of the parameters is required when the user wants to select parts of the entire data set or when the visualization is limited, e.g. on just two dimensions (see also Fig. 2).

The major grouping option is defined by the indicator name, the geographic location, and time. These properties are basic elements that every indicator comprises.

It allows for using 2-dimensional visualization on multidimensional data, just by reduction through one of the grouping options.

Additionally, an advanced grouping option is also considered. This procedure is only available on complex data sources which provide such additional meta-information. An example is EuroStat over the SDMX API. SDMX³ is one of the most established sharing formats for statistical data and meta-information. The structure of meta-information is similar to Linked-Open-Data [14] and can be used to provide an enhanced navigation through the statistical data. In this paper we just consider the major grouping options, because most public data sources are currently in the progress to include additional meta-information. Furthermore, if they have such metadata, they often use very heterogeneous structures which make it difficult to combine them from multiple data sources.

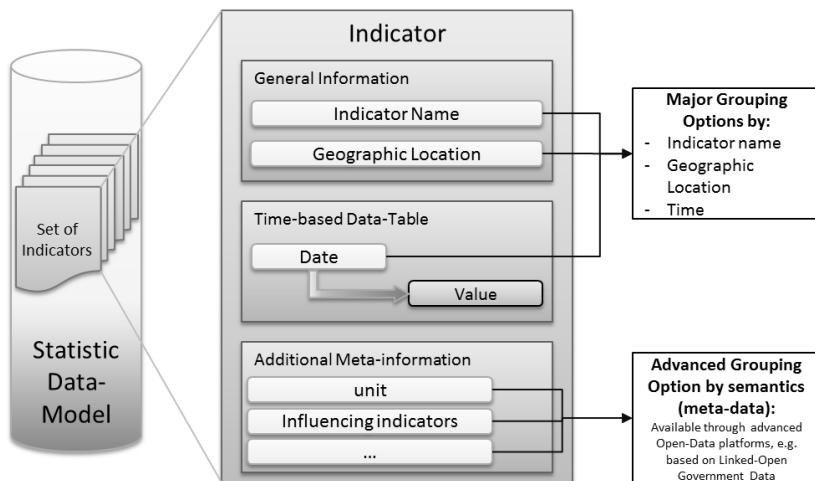


Fig. 2. The structure of the statistic data model is shown, including the main properties of an indicator which is primarily used for the grouping process. An advanced grouping can also be done through additional meta-information

3.2 Extension and Reduction of Data-Complexity for the Statistical Visualizations

A general challenge in coupling various visualizations is the differently supported dimension of the underlying data. A multi-dimensional visualization can show all available dimensions, but simple visualizations, like pie charts, support only two dimensions. As a consequence, the application on a single data base or data model then proves to be difficult in general.

In the previous section we introduced a novel design of a technical data model for statistical data. Based on this concept we show how the high-dimensional data, that

³ More information about the SDMX format on: <http://sdmx.org> (accessed on 29/04/2013).

are available in the presented data model, can be abstracted so that also a low-dimensional visualization can show the same data. This is achieved through a simple limitation of the dimensionality. In this section we focus on the major grouping options.

In the statistical data model we mentioned the indicator's name, the geographic location, and the time data table as major grouping options. We can now adjust the dimensionality through enabling, disabling, or selection/filtering of entries based on the data model. For multi-dimensional data all available indicators with their geographic location and time data table can be shown. In contrast, two-dimensional visualizations reduce the data on just one concrete indicator (e.g. GDP) and for a specific date (e.g. 2012). The final visualization then shows the GDP from 2012 for all available locations. As a consequence, any developed visualization can reduce the complexity in dependence of the supported dimensions. By default, the visualization reduces the complexity in its own way, but it is also possible that the user defines what information needs to be included. This means that the visualization generally can be configured during the interaction process. This also includes changes on the presentation, so that e.g. a line chart can be changed into an area or plot chart. It is also possible that visualizations can be decoupled to allow for a comprehensive view on the data.

Based on this simple data model, different kinds of statistical visualizations can be used on the same data model and therefore on the same data source.

3.3 Creations of Visualization Cockpits

To allow for an effective analysis, the personal orchestration of different visualizations is a beneficial approach. We entitle this visualization orchestration ability "cockpit", because it supports to control the view on the data. In general, the terms "cockpit" and "dashboard" [15, 16] are synonymously used, but we prefer the term cockpit which focuses more on an active use in a very complex environment. For this purpose, the cockpit provides a higher degree of interaction and opportunity to orchestrate a personalized cockpit so that a given task can be solved more efficient.

As a general design we consider a list of data sources, i.e. EuroStat and some local municipality data bases in the top. The user can switch between these data sources. Next to the data base options, the user can also search for one or more indicators that should be visualized. On the right side, a set of visualizations (simple chart visualization, as well as complex visualizations like Parallel Coordinates) is displayed. In the center, the user can drag and drop visualizations to analyze the data. Any visualization can be configured in detail, e.g. decoupling the visualization for a comparative view. In order to focus on the most relevant visualization, the user can also resize the visualizations as needed.

4 Implementation

In a first prototype of a statistics cockpit we implemented some basic statistical visualization as examples, e.g. line chart, pie chart, bar chart, and a data table. On the top, the user can search an indicator. On the right side the user can drag and drop the preferred statistical visualization on the cockpit.

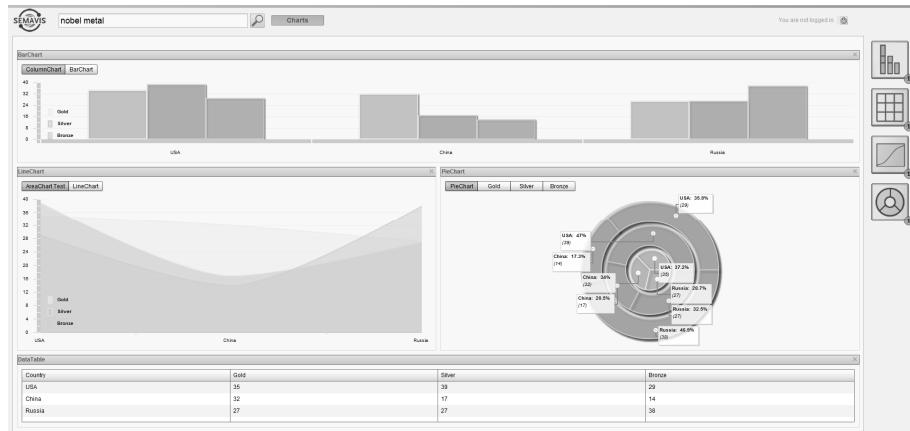


Fig. 3. The picture shows the implemented visual statistics cockpit of noble resources data. The cockpit can be individually composed and resized.

The prototype is currently not yet fully implemented, but it can already be shown that it provides the ability to analyze the data in a sufficient way. Especially in the information foraging phase [17] where analysts aims to identify problems and solutions, a multi-angle view on the data supports the work.

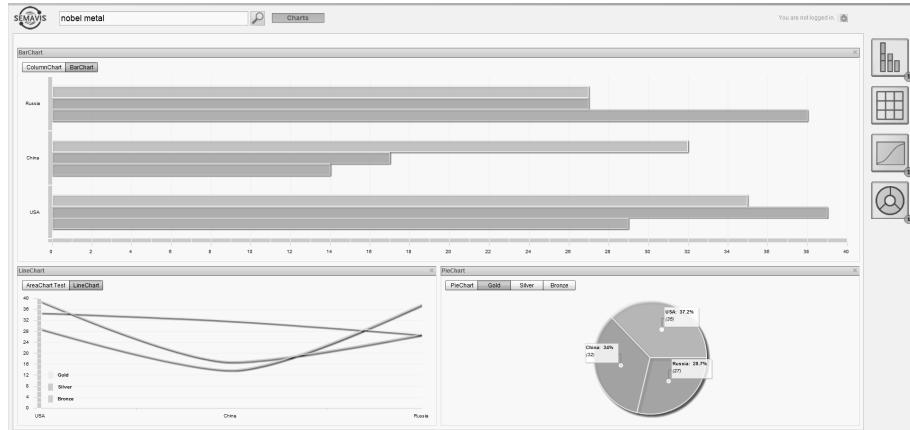


Fig. 4. The picture shows another cockpit view of noble resources data. As can be seen here, the cockpit can be customized and resized easily to the user's preferences and requirements.

We propose to include this statistical analysis in process-driven environment, to support the user to solve his task more efficient [20, 21]. This enables the user to get visualization or other kinds of visual tools in dependence on the current task in the process.

5 Discussion

With this prototype we have built the (conceptual) foundation for an interactive statistics cockpit. Through the use of established and modern analytical visualizations, the user is able to analyze the data. An important question is in how far interactivity can be supported in an elementary way. The goal for explorative analysis depends on the interactivity and the user experience. It is currently difficult to achieve an explorative character in the analysis, because the users have a clear goal they want to achieve. An approach for explorative analysis needs increasing interactivity of a system and more space for just “playing” around with the data, but currently no established strategies are available.

Another open question is the coupling of SDMX meta-information from different data sources. The SDMX meta-information are similar to LOD data source (especially *DBpedia*⁴), but also for LOD no full working and optimal mapping approach does exist. A contribution of such a coupling of SDMX meta-information and other LOD information (e.g. dbpedia) is the ability to show additional explanations to users on demand. So, the user gets, among other, further information about the used unit or how the data items are imposed. So the coupling can act as a possibility to enrich the information about the available statistical data.

6 Conclusion

This paper introduces a concept to provide interactivity in statistical visualizations, which are orchestrated in a statistics cockpit. For this purpose, a new technical statistical data model was introduced that allows for extending or reducing the dimension of the data for different existing statistics visualizations. This approach allows for using the same data on different visualizations types and provides an improved linking ability. The linking is beneficial to support the provision of statistics cockpits and to give a well-organized overview on the data. Such orchestrated cockpits showing the same data in different forms create a multi-angle view on the data.

This overview on the data is useful in the process of policy modeling. Both analyst and expert have to deal with a large number of data in the process of information gathering in order to identify problems and to find possible solutions. The ability to interact with the data allows for an improved solution finding. Additionally, the orchestration of various visualizations also allows for an improved insight in the data. Thus, a problem can be better analyzed and thus better understood as it would be possible in a single visualization.

In future work we aim at bringing the SDMX meta-information with LOD together. The primary goal is to increase the interactivity component. We plan to include better explanation of the statistical data e.g. about the usage of some indicators. For this, some premature works exist, e.g. the linking approach described in [14], which can be enhanced especially in the visual integration.

⁴ DBpedia is a knowledge data-base in Linked-Open Data (LOD) structure, and it is available under: <http://dbpedia.org> (accessed on 29/04/2013)

Acknowledgements. This work has been carried in the FUPOL project, partially funded by the European Commission under the grant agreement no. 287119 of the 7th Framework Programme. This work is part of the SemaVis visualization framework, developed by the Fraunhofer IGD (<http://www.semavis.com>). SemaVis provides a comprehensive and modular approach for visualizing heterogeneous data for various users.

References

1. Hao, M., Dayal, U., Keim, D., Schreck, T.: A Visual Analysis of Multi-Attribute Data Using Pixel Matrix Displays. In: Proc. VDA 2007 (2007)
2. Yuan, X., Guo, P., Xiao, H., Zhou, H., Qu, H.: Scattering Points in Parallel Coordinates. IEEE Transactions on Visualization and Computer Graphics 15(6), 1001–1008 (2009)
3. Heinrich, J., Weiskopf, D.: Continuous Parallel Coordinates. IEEE Transactions on Visualization and Computer Graphics 15(6), 1531–1538 (2009)
4. Ward, M.O., Grinstein, G., Keim, D.: Interactive Data Visualizations: Foundations, Techniques, and Applications. Taylor & Francis Ltd. (2010)
5. Viau, C., McGuffin, M.J., Chiricota, Y., Jurisica, I.: The FlowVizMenu and Parallel Scatterplot Matrix: Hybrid Multidimensional Visualizations for Network Exploration. IEEE Transactions on Visualization and Computer Graphics 16(6), 1100–1108 (2010)
6. Sharko, J., Grinstein, G., Marx, K.A.: Vectorized Radviz and Its Application to Multiple Cluster Datasets. IEEE Transactions on Visualization and Computer Graphics 14(6), 1077–1427 (2008)
7. Draper, G., Livnat, Y., Riesenfeld, R.F.: A Survey of Radial Methods for Information Visualization. IEEE Transactions on Visualization and Computer Graphics 15(5), 759–776 (2009)
8. Wilkinson, L., Friendly, M.: The History of the Cluster Heat Map. The American Statistician 63(2), 179–184 (2009)
9. Rao, R., Card, S.: The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Celebrating Interdependence, CHI 1994, pp. 318–322 (1994)
10. Keim, D.A., Kriegel, H.-P., Ankerst, M.: Recursive pattern: a technique for visualizing very large amounts of data. In: Proceedings of the 6th Conference on Visualization 1995, pp. 279–286 (1995)
11. Havre, S., Hetzler, E., Whitney, P., Nowell, L.: ThemeRiver: visualizing thematic changes in large document collections. IEEE Transactions on Visualization and Computer Graphics 8(1), 9–20 (2002)
12. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings IEEE Symposium on Visual Languages, pp. 336–343 (1996)
13. Jern, M.: Collaborative web-enabled geoanalytics applied to OECD regional data. In: Luo, Y. (ed.) CDVE 2009. LNCS, vol. 5738, pp. 32–43. Springer, Heidelberg (2009)
14. Cyganiak, R., Field, S., Gregory, A., Halb, W., Tennison, J.: Semantic Statistics: Bringing Together SDMX and SCVO. In: Proceedings of LDOW, Raleigh, North Carolina, USA (2010)
15. Few, S.: Information Dashboard Design. Overview article about Dashboards (2012), [http://blogs.ischool.berkeley.edu/i247s12/files/2012/01/](http://blogs.ischool.berkeley.edu/i247s12/files/2012/01/Dashboard-Design-Overview-Presentation.pdf) Dashboard-Design-Overview-Presentation.pdf

16. Duval, E.: Attention please! learning analytics for visualization and recommendation. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, LAK 2011, pp. 9–17. ACM, New York (2011)
17. Kohlhammer, J., Nazemi, K., Ruppert, T., Burkhardt, D.: Toward Visualization in Policy Modeling. *IEEE Computer Graphics and Applications* 32(5), 84–89 (2012)
18. Burkhardt, D., Nazemi, K., Sonntagbauer, P., Sonntagbauer, S., Kohlhammer, J.: Interactive Visualizations in the Process of Policy Modeling. In: Proceedings of IFIP eGov 2013. GI-LNI (2013)
19. Jern, M.: (OECD), What does OECD eXplorer enable you to do? An introduction to its main features. In: Handbook of the OECD eXplorer (2009), <http://www.oecd.org/gov/43142629.pdf>
20. Burkhardt, D., Ruppert, T., Nazemi, K.: Towards process-oriented Information Visualization for supporting users. In: Proceedings of 15th International Conference on Interactive Collaborative Learning, ICL 2012, pp. 1–8 (2012)
21. Burkhardt, D., Nazemi, K.: Dynamic process support based on users' behavior. In: Proceedings of 15th International Conference on Interactive Collaborative Learning, ICL 2012, pp. 1–6 (2012)

Feature Weight Optimization and Pruning in Historical Text Recognition

Fredrik Wahlberg and Anders Brun

Centre for Image Analysis
Uppsala University
Sweden

Abstract. In handwritten text recognition, “sliding window” feature extraction represent the visual information contained in written text as feature vector sequences. In this paper, we explore the parameter space of feature weights in search for optimal weights and feature selection using the coordinate descent method. We report a gain of about 5% AUC performance. We use a public dataset for evaluation and also discuss the effects and limitations of “word pruning,” a technique in word spotting that is commonly used to boost performance and save computational time.

1 Introduction

In off-line recognition of a historical text, the starting point is are images of some manuscript or other material that can host written characters. With historical documents, problems like degraded ink, rough handling over generations or geometrically distorted parchment due to moisture needs to be taken into account. The problem of performing a full recognition (i.e. computerized transcription) is so far unsolved. To still be able to search and index the treasures in our libraries today, searching for user selected templates, called word-spotting, has been shown to be useful.

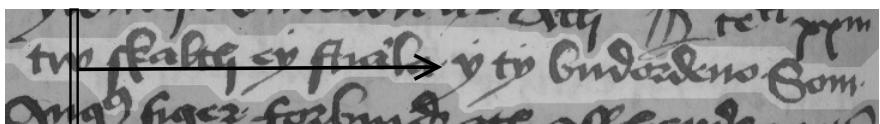


Fig. 1. An automatically segmented line of handwriting in old Swedish from the 16:th century. A sliding window is passed over the text line and feature vectors are extracted along its path.

Written text is linear i.e. each letter should be read before the next (with a few exceptions ranging two or three letters). By treating all text as if it were on the same line, the problem of parsing visual information can be reduced into a sequence matching problem. Extracting feature vectors along a centre path

of each text line is called “sliding window” feature extraction, an illustration is shown in figure 1. In this paper, we will examine the importance of each single feature, potential redundancy and benefits from weighing features differently. We also present a method for finding weight combinations.

In contrast to earlier papers on the same theme [1–3], we have not used pruning (i.e. heuristic exclusion rules based on simple geometric features) to exclude potential word matches. We argue that pruning gives rise to an unnecessary limit on the recognition rate. Different recognition techniques are more or less robust to noise. We have chosen a training free feature matching approach to not “obscure” the performance on the raw data with effects of robustness techniques.

1.1 Previous Work

“Sliding window” feature extraction, pioneered by [4] and commonly used in text recognition[5, 6], catches some information relevant to text recognition, at each pixel column. The quantified information is then concatenated into a feature vector. In [7], the most commonly used “sliding window” feature vector was proposed. The authors also proposed a preprocessing scheme for the word image before feature extraction to increase robustness.

In [1], the concept of dynamic time warping (DTW) for word matching was developed, with an application focus. The goal was to be able to create an ordered list of matches between template word images and some collection of word images. DTW finds the optimal flexible best match of two “sliding window” feature vector sequences, returning a dissimilarity cost.

In [3], several computationally fast pruning rules were introduced, later developed further by [8] and used by [2, 1]. By using these rules, a large portion of potential matches can be removed before executing DTW. However, false negatives are introduced, which we argue creates an unnecessary limitation on the word matching.

In [2], some commonly used features, including most from [7] were evaluated. Average precision scores for some features were given and discussed. Collections of features were only analyzed in passing. We have extended this analysis to collections of features together with using optimization to find improved weighting of specific features.

2 Method

Each element in the feature vector, given by the “sliding window” feature extraction scheme, is created from several local features. To examine the effect of weighting the influence of each feature, we have built an evaluation environment reproducing what is described in [1, 2]. Their implementation consisted of a two part word matching pipeline, pruning and DTW, described below. We have used a standard evaluation set and focused on the word spotting problem after text lines and words have been segmented (part of a page can be seen in figure 2). Below, we describe the effects of pruning and methods for finding feature weights in this context.

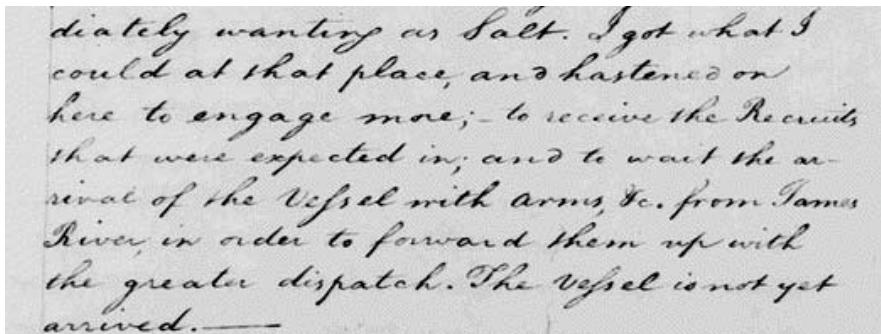


Fig. 2. A part of a page from the Washington letters

2.1 Sliding Window Feature Extraction

After segmentation and normalization (as in [7]) of the text lines, relevant information at each pixel column was reduced to a vector in a feature space. Each element in a feature vector captured some, to handwriting, important characteristic of the text line around the associated column. Several commonly used features exist in the literature (e.g. [7, 2, 4, 5]), the ones we have used in our experiments are described below.

- 1. Projection profile.** The vertical projection of the pixel column i.e. the sum of foreground pixels in one column.
- 2. Upper contour.** The position of the foreground pixel with the highest position i.e. lowest y coordinate value.
- 3. Lower contour.** The position of the foreground pixel with the lowest position i.e. highest y coordinate value.
- 4. Upper projection.** The projection profile above the upper baseline. The upper baseline is a vertical line through the image that is placed directly above the upper part of most lower case letters and crosses through ascenders. This roughly gives a projection profile of only the ascenders and upper case letters.
- 5. Lower projection.** The projection profile below the lower baseline. The lower baseline is a vertical line through the image that is placed directly below the lower part of most letters and crosses through descenders. This roughly gives a projection profile of only the descenders.
- 6. Centre of Weight.** The mean of all y coordinate values of the foreground pixels in a column.
- 7. Foreground/background transitions.** The number of transitions between the foreground and the background while following the pixels in a column from lowest to highest y coordinate value.
- 8. Second moment.** The variance of all y coordinate values of the foreground pixels in a column.

- 9. Gradient of the upper contour.** First derivative of the upper contour, this feature took neighbouring pixel columns into account to form a more reliable gradient.
- 10. Gradient of the lower contour.** First derivative of the lower contour, this feature took neighbouring pixel columns into account to form a more reliable gradient.
- 11. Foreground fraction.** The fraction of pixels between the upper and lower contours belonging to the foreground.

Some of the features listed above were considered by [7] to need preprocessing to increase robustness (e.g. in the upper contour, the ascender of a letter like “k” was spread out or seen as a short spike depending on slant). A “normalization” was introduced to the word images, normalizing letter height and removing slant and skew. In the evaluation database, described below, the normalization step has already been performed. All features were normalized to the interval [0, 1].

2.2 Word-Spotting

By using “sliding window” feature extraction, word-spotting can be treated as a sequence matching problem. Word images are compared by treating them as sequences of feature vectors. Using DTW for sequence comparison, gives a dissimilarity measure between two word images. This concept originates from speech recognition, hence the wording of “time warping.” Three words from our evaluation set, described below, can be seen to the left in figure 3.

An optimal warping for matching sequences of feature vectors can be found by using dynamic programming. The feature vector sequence from the template word, sequence A , is compared to the sequence of some other word, sequence B . A cost matrix W is generated where each feature vector of A is compared to every feature vector of B , as in equation 1, where $d(\cdot)$ is some distance function (square Euclidean distance in our case).

$$W(i, j) = \sum_{k=1}^N d(A_i, B_j)^2 \quad (1)$$

The minimal cost path from the upper left corner to the lower right corner of the matrix represent an optimal warping, with respect to the distance function. The allowed steps through this matrix are down (skip one feature vector of A), right (skip one feature vector of B) or down-right (match). By using the Sakoe-Chiba band constraint [9], matching performance can be increased and calculation time lowered. By only allowing a diagonal path through the weight matrix for warping, pathological warpings (where a very small portion of one word is matched to large parts of another) are avoided. In [1], the allowed warping (i.e. the number of elements around the diagonal) was set to 15. An example, with an allowed warping of ± 15 and with the lowest cost path marked, is shown to the right in figure 3.

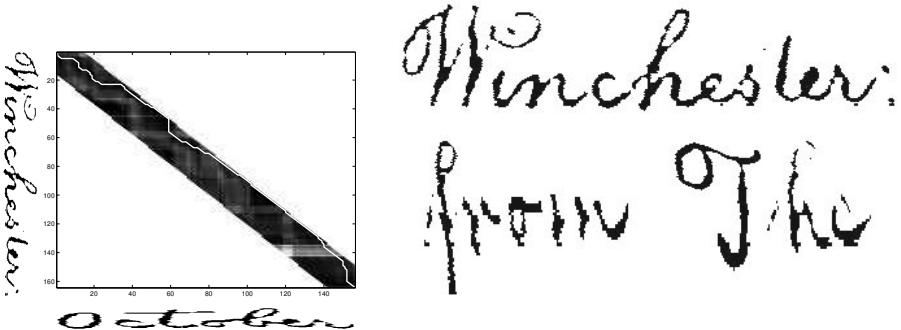


Fig. 3. Right: Illustration of the Sakoe-Chiba constraint on the weight matrix in dynamic time warping. Only the diagonal part of the weight matrix is considered valid for warping (diagonal ± 15 elements). The lowest cost path (white line) through the diagonal band represent the warping. Left: Three segmented, binarized and normalized words from the Washington database, used in our evaluation. Note the unnatural slant on the “f” in “from.” This is because letters are normalized according to the dominant slant.

Pruning. In the word spotting pipeline, many potential matches can be removed by pruning. We have examined the rules used in [1, 2], recommended by [8]. The recommended pruning rules were limits on the area and aspect ratio of the word bounding boxes. One such rule is shown in equation 2, the parameter β was set by experimentation. Also, a rule only allowing matches between words with equal number of descenders was recommended. The number of descenders was defined as the number of connected components below the lower baseline.

$$\frac{1}{\beta} \leq \frac{\text{template}_{bbxArea}}{\text{image}_{bbxArea}} \leq \beta \quad (2)$$

The matching pipeline of [2] was divided into two steps, pruning and DTW. Using the rules described above reduced the need for matching by DTW and thus the computational cost. We have tried to set the pruning parameters to mimic earlier results, where approximately 85% of the word pairs (potential matches) were ruled out. Using such aggressive pruning identify many true negatives but also misidentifies a significant number of false negatives.

In figure 4, the parts of the Receiver Operating Characteristics (ROC) plot (explained in 3.1) affected by the pruning are shown. We also illustrate in the figure the little room left for DTW to improve on the overall result by using random word dissimilarity scores, instead of DTW. Pruning has been shown to be successfully applied to specific word-spotting scenarios but gives little room for feature based matching. It sets an upper limit on the true positive ratio, Area Under the Curve (AUC) (explained in section 3.1) is constrained to the interval [0.69, 0.82]. Hence, we will not use it in our evaluation.

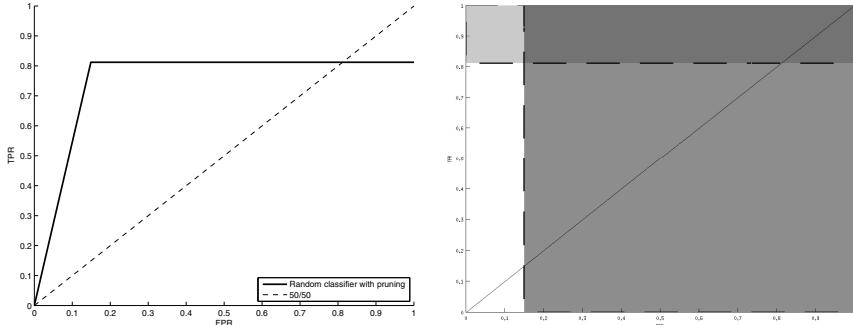


Fig. 4. Left: ROC curve from using pruning and random dissimilarity scores instead of DTW. Right: A ROC plot showing the part affected by DTW, after pruning, in white. The horizontal dashed line is the upper limit of the ROC curve because of false negatives for the pruning. The vertical dashed line show the discovered true negatives from the pruning and before DTW. Gray areas are added for illustration.

2.3 Feature Weight Parameter Space

The 11 features, described above, catches some relevant characteristics of the text (i.e. they contain information relevant to the word matching). At some points many features are strongly correlated, e.g. the upper projection, upper contour and projection at an ascender. If each feature were weighted equally, some word characteristics (e.g. number of ascenders) might dominate the matching. To avoid this in training free matching, manual tuning is required.

Leave One Out. If some features are redundant, for the current evaluation data, removing it should not give a lower evaluation result than using the full set. By running the word-spotting evaluation one time for each single feature removed, an estimation of which features are the least important can be made. Repeating the process for the remaining features can remove one more etc. This method is not as accurate as an extensive search since not all correlation and noise effects of different feature combinations are taken into account.

Weight Optimization. The problem of finding the optimal weights, given the data set, can be treated as an optimization problem. The function to minimize is $1 - \text{AUC}$, for an 11D vector with weights used in a evaluation. Finding a better weight vector pushes the ROC curve upwards and to the left, which corresponds to earlier true positives.

The discrete nature of evaluating matches between a limited number of words give rise to a piecewise flat parameter space. Commonly used gradient based optimization methods, like gradient descent, then fail to find a direction to move in. Experiments using the coordinate descent algorithm were successful [10]. When running one iteration, the objective function is minimized one coordinate at a time. The performance increases monotonically, but the algorithm cannot

guarantee to find a global optimum. In fact, this kind of block relaxation technique may not even find a local minimum.

In our implementation, each direction was searched using the relatively large step length of 0.1 in a space where each weight can have the interval [0, 1]. In cases where the performance did not change with the new weight, we chose the lowest possible weight to create bias towards removing features. We used a training set of 20% (1000 words) of the full database, separate from our evaluation set. The algorithm has empirically been shown to improve the results, even though convergence could not be guaranteed.

3 Experiments

Potential feature redundancy must be seen in context of the remaining features. Removing, or weighting down, one feature does not necessarily mean that it is not a good feature, only that it does not contribute in some collection of features. Also, some features are probably very valuable when separating certain letters but not as useful in most cases. In our evaluation, we could only analyse how important a feature is when comparing whole words.

3.1 Evaluation Database

George Washington Papers at the Library of Congress is a collection of letters written by George Washington before, during and after his presidency. The original images (of which a part can be seen in figure 2) are kept by The United States library of congress and from those a public database [11] has been created. All of the material used is from letter book 1, series 2, pages 270-279 & 300-309, written between Aug. 11, 1754 and Dec. 25, 1755¹. The database contains 5000 segmented, normalized and labeled words. Three examples of these words are shown in figure 3.

Performance Measure. Figure 5 and 4 show a Receiver Operating Characteristic (ROC) plot. On the y -axis is the rate of true positive matches (TPR) i.e. the number of correctly identified matches at a specific threshold. On the x -axis, the corresponding rate of false positives (FPR) are given. Along the diagonal, a 50/50 line is drawn to show a comparison to a random classification. The performance measure used in table 1 is the Area Under the Curve (AUC)[12], giving a single number for performance comparison.

3.2 Evaluation of Single Features and Collections

In table 1 the results from some evaluations are shown. First, each feature was evaluated separately showing that using only the projection feature was enough to get a significant improvement. Note that this only gives insight into how useful a feature is overall. Second, one feature was taken out and the others weighted uniformly. Third, the lower gradient feature (the least important one from the previous step) was taken out together with each one of the remaining features.

¹ <http://memory.loc.gov/ammem/gwhtml/gwseries2.html>

Table 1. Results from performance evaluation of both every single feature and some collections. Performance metric is Area Under the Curve. Bold text indicates that the results was better than using all feature types with uniform weights, giving an AUC of 0.8418.

Feature	Single	Leave one out	Leave two out
Projection	0.877	0.839	0.858
Upper contour	0.853	0.840	0.859
Lower contour	0.788	0.842	0.861
Upper projection	0.871	0.840	0.856
Lower projection	0.648	0.836	0.856
Centre of weight	0.784	0.842	0.861
BW transitions	0.802	0.843	0.863
Second moment	0.792	0.843	0.862
Upper gradient	0.794	0.841	0.867
Lower gradient	0.796	0.861	-
Foreground fraction	0.827	0.826	0.843

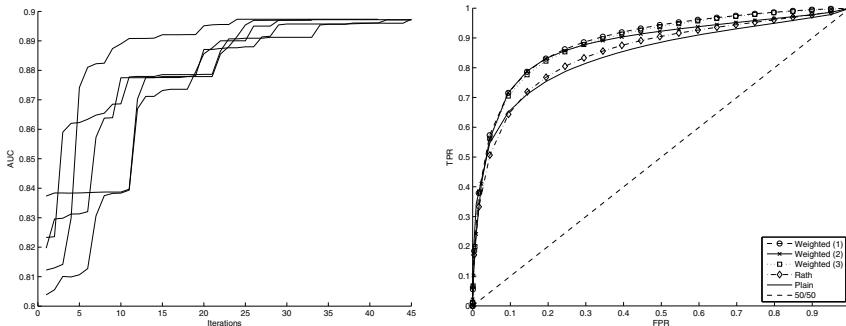


Fig. 5. Left: Curves of AUC over several iterations using random starting vectors. Right: Some results from using weights chosen by the optimizing algorithm. A plain evaluation using uniform weights and the feature subset proposed by [1] are shown as a comparison.

3.3 Parameter Search

In figure 5, the improvement from the weight optimization are shown. In our experiments, a lot of improvement of the AUC score could be achieved by a small number of iterations. We used a subset of the words (20%) for training and a disjunct set for evaluation.

When trying 15 random initial vectors and one with uniform weights (all elements set to 1), the algorithm converged to approximately the same result independent of starting point. This would indicate that the weight parameter space does not contain that many local minima. The AUC score was also improved (> 0.05). We have done preliminary evaluations of the weight vectors on the public data set Saint Gall with encouraging results. This supports the generalizability of the weight vectors trained on the Washington letters.

Table 2. Examples of vectors in the weight parameter space generated by the optimization algorithm

	11 features (ordered as in section 2.1)											AUC
Weights	0.6	0	0	1	1	0.4	0	0	0.2	0	0.3	0.899
	0.9	0	0	0.9	1	0	0	0	0.2	0	0.3	0.900
	0.8	0	0.4	0.9	1	0	0	0	0.2	0	0.3	0.899
Uniform	1	1	1	1	1	1	1	1	1	1	1	0.842
Weights from [1]	1	1	1	0	0	0	1	0	0	0	0	0.851

Some features were in most cases set to zero or a low number. The features most often removed were the contours, centre of weight, second moment, foreground/background transitions and the gradient of the lower contour. This would indicate that these are redundant, assuming our evaluation database. Some examples of vectors in the weight parameter space generated by the optimization algorithm are shown in table 2.

4 Conclusions

In context of our evaluation strategy, several features seem to be less important or even redundant. One of the most important features seem to be the projection profile, supported by it's high AUC score in table 1. A projection profile “spikes” at ascenders and descenders. Matching using this feature only would give existence and position of ascenders/descenders a high impact. The optimization consistently removed (or significantly lowered) the weight on the contours, centre of weight, second moment, foreground/background transitions and the gradient of the lower contour. Even though the optimizer can not guarantee a global (or local) minimum, the approach was successful in terms of increasing the performance.

We show that the limits set on the DTW by the pruning makes even a random classification of the remaining word pairs perform adequately. We conclude that in previous studies of word spotting, the accuracy was to a large extent limited by the initial pruning step. The following DTW can not bring back false negatives and that limits the AUC to the interval [0.69, 0.82]. Compare this to using uniform weights with and AUC of 0.84, using only projection at an AUC of > 0.87 or the features in [1] without pruning where the AUC was 0.85. Using our optimization strategy, the evaluation results was pushed to > 0.89.

References

1. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II-521–II-527 (2003)
2. Rath, T.M., Manmatha, R.: Features for Word Spotting in Historical Manuscripts. In: International Conference on Document Analysis and Recognition, pp. 218–222 (2003)

3. Manmatha, R., Croft, W.B.: Word Spotting: Indexing Handwritten Archives (1997)
4. Schwartz, R., LaPre, C., Makhoul, J., Raphael, C., Zhao, Y.: Language-independent ocr using a continuous speech recognition system. In: Proceedings of the 13th International Conference on Pattern Recognition, vol. 3, pp. 99–103 (1996)
5. Wienecke, M., Fink, G., Sagerer, G.: Toward automatic video-based whiteboard reading. International Journal of Document Analysis and Recognition (IJDAR) 7, 188–200 (2005)
6. Pltz, T., Fink, G.: Markov models for offline handwriting recognition: a survey. International Journal on Document Analysis and Recognition (IJDAR) 12, 269–298 (2009)
7. Marti, U.V., Bunke, H.: Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. International Journal of Pattern Recognition and Artificial Intelligence 15, 65–90 (2001)
8. Kane, S., Lehman, A., Partridge, E.: Indexing george washingtons handwritten manuscripts. Center for Intelligent Information Retrieval. Computer Science Department, University of Massachusetts, Amherst, MA 1003 (2001)
9. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 26, 43–49 (1978)
10. Bertsekas, D.P., Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific (1999)
11. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character hmms. Pattern Recogn. Lett. 33, 934–942 (2012)
12. Bradley, A.P.: The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159 (1997)

A Constraint Inductive Learning- Spectral Clustering Methodology for Personalized 3D Navigation

Nikolaos Doulamis¹, Christos Yiakoumetsis²,
George Miaoulis², and Eftychios Protopapadakis³

¹ National Technical University of Athens
ndoulam@cs.ntua.gr

² Technological Education Institute of Athens
{chgiak,gmiaoul}@teiath.gr

³ Technical University of Crete
eft.protopapadakis@gmail.com

Abstract. The recent advances in ICT boost research towards the generation of personalized Geographical Information Systems (p-GIS). It is clear that selection of a route based only on geometrical criteria, i.e., the route of the shortest distance or the minimum travel time, very rarely coincides with a “satisfactory itinerary” that respects users’ preferences, that is their desires to navigate through buildings or places of his/her own particular interest. Additionally, 3D navigation gains more popularity compared with 2D approaches especially in virtual tourist and cultural heritage applications. In a p-GIS, user’s preferences can be set manually or automatically. In an automatic architecture, user preferences are expressed as a set of weights that regulate the degree of importance on the route selection process and on line learning strategies are exploited to adjust the weights. In this paper, the on-line learning strategy exploits information fed back to the system about the relevance of user’s preferences judgments given in a form of pair-wise comparisons. Then, we use a constraint fusion methodology for the dynamic modeling of user’s preference in a 3D navigation system. The method exploits an active inductive learning approach that is combined with an adaptive spectral clustering scheme in order to avoid smoothing during the weight adjustment process.

1 Introduction

A Personalised Route Guidance System is a geographically-enriched decision support system that incorporates methods for automatic or semi-automatic, in the sense of minimum users’ interaction, route selection by exploiting user, geographical and 3D object metadata [1],[2]. It is clear that selection of a route based only on geometrical criteria, i.e., the route of the shortest distance or the minimum travel time, very rarely coincides with a “satisfactory itinerary” that respects users’ preferences, that is their desires to navigate through buildings or places of his/her own particular interest.

However, creation of an architecture for personalised route planning requires, on the one hand, (i) advanced knowledge methodologies able to facilitate efficient description of the geographical information,(i.e., *the feature extraction process*) and,

on the other, (ii) dynamic *on-line learning strategies* that relate the extracted metadata with the user's preferences dynamics. However, it is clear that extraction of representational features is a challenging and application-dependent process [3]. This is mainly due to the so-called "*semantic gap*"; humans perceive content with high level concepts that cannot be modelled with the extracted low level features [4].

Another difficulty towards a truly personalised route planning arises from the *human's subjectivity* [5]. Different persons or even the same under different circumstances perceive the same visual content quite differently leading to several issues in user profiling [6]. This means that there is no a unique mapping scheme that relates the extracted low-level metadata (features) with the high-level concepts [7], [4]. The current, personalised route planning architectures use a set of explicit metadata (usually textual) to describe user's preferences [9]. Such systems are called "adaptable personalized route guidance systems". On the contrary, the systems that use an automatic procedure in the user's preferences estimation are called "adaptive personalized route guidance systems" [10]. In both cases, weights are used to regulate the degree of importance of each feature element on route selection [2]. In the manual case, the weight values are explicitly provided by the user, while in the automatic case, the weights are provided by the system based on an on-line learning algorithm. However, setting manually the weights is a very demanding process, since there is no a quantitative association between user's preferences and feature metadata. Our research, is focused on automatically setting the weights.

1.1 Previous Work

Usually, the current personalized route guidance systems exploit an explicit methodology for defining the weight values [2], [9], which are, then involved in multi-criteria optimization strategies for best route selection. To address the inconsistency of describing the rich geographical media content with low level descriptors, the work of [9] introduces an ontological framework. In this way, [9] provides a better description of the semantics of geo-referenced and contextual information, such as tourist attraction, road safety issues (telephone, medical assistance, etc) and road facilities (gas station, services, terminal). However, this work focuses only on the ontological representation and fails to provide on-line learning strategies, which are necessary for a truly adaptive architecture. Another approach incorporates symbolic information with visual components, extracted through the application of image processing tools to provide a cognitive spatial path planning [11]. However, again, this work is focused on a better organization of the metadata (textual or visual) that describes the geographical content, instead of modelling user's preferences. A different approach is presented in [2] where multi-criteria strategies are presented for best route selection.. However, again the focus is on the optimization for the best route instead of personalization. In the same context (e.g., route path creation) there are works that use agents for route planning [12]. However, still the personalization aspects are limited.

1.2 Contribution

In this paper, we propose an adaptive personalized route guidance system for cultural heritage purposes. The architecture exploits a constrained fusion methodology that

combines active inductive learning with spectral clustering algorithm. In particular, our method, in contrast to the current approaches, estimates 3D paths which are more suitable for cultural heritage purposes. Additionally, our research automatically estimates the weight values which are associated with metadata elements used to describe the rich geographic media content. Automatic estimation of the weights values is performed through on-line learning strategies based on a pair-wise comparison methodology; the metadata weights are adjusted by information fed back to the system about the relevance of user's preferences judgments given in a form of pair-wise comparisons.

Using a small set of pair-wise compared elements, we construct a graph that dedicates the degree that one object is preferred against another. In this way, our research exploits an on-line inductive learning algorithm that based on constructed preferred judgments graph calculates an overall ordering of the depicted to the user objects according to his/her information needs. The overall object ordering is used to initially estimate the weights of metadata elements provided an initial estimation of user's preferences. Then, an adaptive spectral clustering algorithm is implemented to partition the ranked objects into a set of disjoint classes. We select spectral clustering as it provides good classification capabilities for non-Gaussian, complex distributions. The outcome of the spectral clustering is constrained with the outcome of inductive learning to formulate a non-linear best route selection scheme being optimized on the use of a genetic algorithm. In contrast with previous approaches like [13], the proposed constrained methodology increases the capabilities of modeling user's preferences. This is mainly due to the fact that few extreme preferred objects are classified into different classes and therefore their influence on the route selection is not averaging by the remaining many preferred objects.

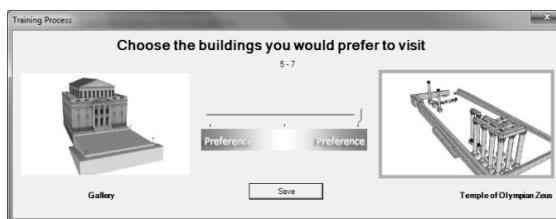


Fig. 1. The *pair wise comparison process*: The National Gallery of Athens is presented against the Temple of Olympian Zeus and the user selects the Olympian Zeus Temple

2 Modeling User Preferences through Pair-Wise Comparisons

The first step of the proposed architecture is to extract features, that is, metadata, that describe the attributes of the cultural objects. In this paper, we use as features the ones presented in the [13]. For clarity of presentation, we briefly describe, in the following section, the features used. Following the feature extraction process, the second step of the proposed architecture is to automatically estimate the weight values that regulate the degree of importance of the feature elements involved in the route selection process. In this paper, the automatic estimation of the weight values is performed through a pair-wise comparisons scheme. In particular, initially the system

depicts to the user two randomly selected cultural heritage objects and forces the user to select the most preferred one between the two. Fig. 1 indicates an example of our pair-wise process adopted in this paper to trigger the on-line learning algorithm used for the automatic estimation of the weight values that model user's preferences.

2.1 Cultural Heritage Attributes

Quantitative and qualitative metadata are used to describe the attributes of a cultural heritage object [13]. Qualitative metadata includes the type of building use, the building style, the construction material, the building location, attractive content and the type of city block. Each of the aforementioned qualitative metadata get predefined values. For example, construction material can be wood, marble, clay etc. On the other hand quantitative metadata includes the building cost of visit, interactivity, popularity, age and geometry. In the following, we denote as $\mathbf{m}_{o_i} = [m_{o_i}^1 \ m_{o_i}^2 \ \dots]^T$ the metadata vector of an object o_i , where $m_{o_i}^i$ is the i -th attribute of the object o_i .

3 Active Inductive-Based Learning Process

Let us also denote as $g(\cdot)$ a user preference function defined as

$$g(u,v) : S \times S \rightarrow [0 \ 1]. \quad (1)$$

where $u, v \in S$ are two selected objects (see the previous Section). Function $g(u,v)$ defines the order between the object pair (u,v) with respect to user's preferences. Variable $g(u,v) \rightarrow 1$ means a strong recommendation of being u more preferred. The opposite is happened in case of $g(u,v) \rightarrow 0$, whereas a value around $\frac{1}{2}$ indicates an abstention of making any recommendation.

Using the pair-wise recommendations, we form a graph $G = \{V, E\}$ where variable V indicates the vertices while variable E the edges of the graph. It is clear that the vertices of the graph coincide with the number of objects depicted to the user for the pair-wise comparison, denoted as N . Instead, the graph edges coincide with values of function $g(u,v)$ of Eq. (1). An example of the constructed graph is depicted in Fig. 2.

Using this graph, we are able to perform an overall ordering of all objects depicted to the user according to his/her preferences [14]. As mentioned in [14], to derive the overall order is a NP-hard problem. To overcome this problem, the authors of [14] present an approximate algorithm to compute the overall ordering of the depicted CH objects to the users. Let us denote as $\beta(u)$ a value that indicates the difference between the outgoing and ingoing edges. It is clear that large value for the outgoing metric means that object u is highly preferred, while the opposite is held for the ingoing metric. Variable $\beta(u)$ is estimated through the following equation

$$\beta(u) = \sum_v g(u,v) - \sum_v g(v,u). \quad (2)$$

Using variable $\beta(u)$, we can define a greedy algorithm to approximate the overall ordering of the CH objects depicted to the user. In particular, the greedy algorithm proceeds as follows: Initially, we select as the most suitable object, the one that maximizes the value of $\beta(u)$ and therefore, the respective node is removed from the graph. In the next step, the values of the remaining objects (vertices) of the graph are potentially updated and the node is the next maximum value of $\beta(u)$ is selected as the most appropriate. This process is iteratively repeated until no nodes are included in the graph. The aforementioned recursive approach is presented in Fig. 2. .

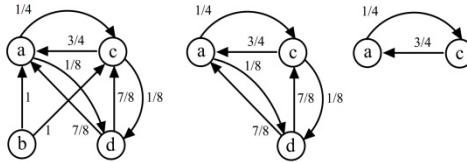


Fig. 2. The greedy algorithm used to estimate the overall ordering. In the leftmost graph object b will assign the maxima value and will be deleted, resulting to the middle graph. Then, node d will assign the maxima value of $\beta(u)$ and thus it will be deleted, leading to the rightmost graph. Finally, node c will be ranked ahead a. Therefore, we have $b > d > c > a$ [14].

The main advantage of the proposed algorithm is that it can provide an estimate of the overall ordering of the objects of the graph, by exploiting only pair-wise comparisons. The main limitations is that the inductive learning algorithms a) fail to direct estimate the weight values associated of the metadata elements of the objects to model users' preferences and b) therefore is not able to generalize the results to objects that are not depicted for judgment by the user. To address the aforementioned limitation, we combine in the following a clustering algorithm with the active inductive learning in order to categorize the ordered objects into clusters of similarity.

4 Spectral Clustering

Spectral representation in classification presents many advantages compared to the traditional center-based approaches (see [15]), since it has the advantage of performing well with non-Gaussian and complex clusters as well as being easily implementable. Using the graph representation presented in Section 3, we are able to apply the spectral clustering algorithm with the purpose of detecting the common distributions among the selected preferred objects. In contrast with the aforementioned described active inductive learning algorithms, spectral clustering partitions the depicted objects to the user into groups of common properties. In the continuous case the algorithm optimize the following equation

$$(\mathbf{D} - \mathbf{E}) \cdot \mathbf{x} = \lambda \cdot \mathbf{D} \cdot \mathbf{x} . \quad (3)$$

where \mathbf{E} denotes the matrix of the edges of the graph (see for example the graph of Fig.2 as created by the feedback of the user in the form of preference judgments

between pairs of selected 3D objects), while \mathbf{D} is a diagonal matrix $\mathbf{D} = \text{diag}(\cdots d_i \cdots)$, whose elements d_i , $i=1,2,\dots$ are the cumulative degrees of $d_i = \sum_j e_{ij}$.

Rounding the aforementioned solution into the district space of the K available classes, we result the final clusters of the spectral algorithm. However, rounding is an NP-hard process [16]. A simple rounding process is to set the maximum value of each row of the eigenvector matrix, derived as a solution of (3), equal to 1, while the remaining values equal to zero.

5 Best Route Selection

Using the active inductive learning algorithm, we have the order of the objects selected by the user. Let us rank objects u in descending order according to their values $\beta(u)$. Let us denote as $u_{i_1}, u_{i_2}, u_{i_3}, \dots$ the first, second, third, ... ranked object of S respectively. It is held that $\beta(u_{i_1}) \geq \beta(u_{i_2}) \geq \beta(u_{i_3}) \geq \dots$. Then, we can select $M < N$ (recall that N refers to the total number of objects depicted to the user) objects according to the energy of the M first ranked objects, defined by the values of $\beta(u_{i,j})$.

On the other hand, using the spectral clustering algorithm, we are able to estimate a set of K classes that contain objects of similar properties.

The first step of the fusing algorithm is to estimate the weights of the metadata that regulate the degree of importance of the features-elements to the selection score.

$$S(u) = \sum w_u^j f_u^j, \quad (4)$$

where f_u^j denotes the j -th feature element for object u , while w_u^j the respective weight. The weights w_u^j are estimated by the first M ranked objects as provided by the inductive learning algorithm. In particular, the weight factors w_u^j are estimated as the inverse ratio of the standard deviation of the respective feature element over all the M selected objects.

$$w^j = \frac{1}{\text{std}(m_{u_{i_1}}^j, m_{u_{i_2}}^j, \dots, m_{u_{i_M}}^j) + \delta}, \quad (5)$$

where the $\text{std}(\cdot)$ denotes the standard deviation operator, while the $m_{u_{i_m}}^j, m=1,2,\dots,M$

the j -th element of metadata vector $\mathbf{m}_{u_{i_m}}$.

Equation (5) means that feature elements that share similar values among all the M objects lead to large weight values (small standard deviation) since in this case the respective feature element seems to be consistent with respect to user's preferences. The opposite happens in the non consistent case. Using Eq. (4), the best personalized route is selected as the one that optimizes the following equation

$$C = \gamma_1 \cdot S(\vartheta(r,1)) + \gamma_2 \cdot S(\vartheta(r,2)) + \dots , \quad (6)$$

where $\vartheta(r,m)$ is an operator that returns the m-th object of route r and γ_i weight factors. In [13], a genetic algorithm is used to solve the aforementioned optimization problem. It should be mentioned that, in contrast to this work, in [13] additional terms have been included to estimate the best route, such as affective properties. Other alternatives exist in the literature (see [2]) a quantifier-guided ordered weighted averaging (OWA) aggregation operator is adopted for selecting the best route.

5.1 Inductive Learning Constrained by Spectral Clustering

The aforementioned procedure for estimating the best route does not take into account the results obtained from the spectral clustering algorithm. In other words, the values of the weight metadata are estimated by the order of the selected objects as provided only by the inductive learning. The main limitation of this approach which is also adopted in [13] is that a cultural object that significantly differs from other similar objects but it may be salient for the user, it cannot affect the weights in such a degree to play an important role in route selection. This is mainly due to the fact that Eq. (5) is in fact an averaging operator that smoothes extreme user's preferences.

To compensate this drawback, we fuse in this paper, the results obtained by spectral clustering on the solutions provided by the inductive learning. In particular, we classify the M ranked objects according to the available clusters provided by the spectral method. Therefore, we can define a probability distribution of user's preferences with regards to the K available classes.

$$p(i) = \frac{n_i}{M}, i = 1, \dots, K, \text{ with } \sum_{i=1}^K n_i = M \quad (7)$$

Using eq.(7) we can define the probability distribution of a route r . In particular, let us denote as r a route that consists of M objects $r = \{u_1, u_2, u_3, \dots, u_M\}$. Then, we can define the probability f the route r to belong to one of the K available classes provided by the spectral clustering algorithm,

$$p_r(j) = \frac{\sum_{i=1}^M \text{Index}(u_i, j)}{M} \quad (8)$$

where operator $\text{Index}(u_i, j) = 1$ if the object u_i belongs to the j -th cluster out of K available and $\text{Index}(u_i, j) = 0$ otherwise. Having estimated the probability of a route r to belong to one of the K available clusters, we can estimate the probability distribution as the concatenation of the probabilities $p_r(j)$ over all clusters $j=1,2,\dots,K$.

$$\text{Pdf}(r) = \{P_r(1), P_r(2), \dots, P_r(K)\} \quad (9)$$

Using the previous equation, we can modify the route selection process as follows;

$$C = \begin{cases} \gamma \cdot S(\vartheta(r,1)) + \gamma_2 \cdot S(\vartheta(r,2)) + \dots & \text{if } \text{Pdf}(r) \approx \text{Pdf}^{(\text{user})}(r) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

In eq. (9), $Pdf(r)$ is the probability density of the route r as defined in Eq. (8), while $Pdf^{(user)}(r)$ denotes the targeted probability density of the route r provided by the user's judgments.

6 Simulations

In this paper the personalized route planning platform have been developed using the Google Earth Building maker tools, which they have been integrated into the collaborative web GIS platform developed in our research laboratory. Each building was reconstructed as a 3D object model and each face of the 3D model was described using the metadata of Section 2. Furthermore, we assume that all the 3D models have five faces only (the face that points to the ground is not evaluated).

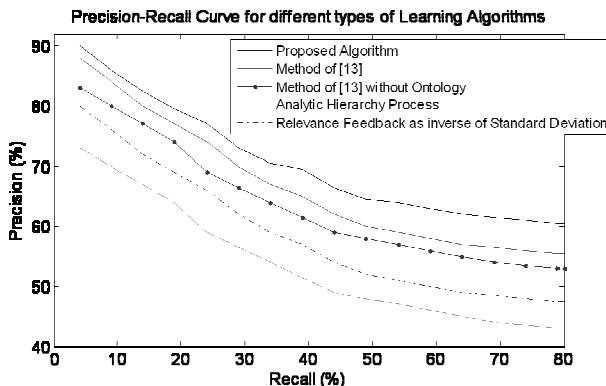


Fig. 3. Comparisons of the proposed personalized route selection algorithm with other approaches

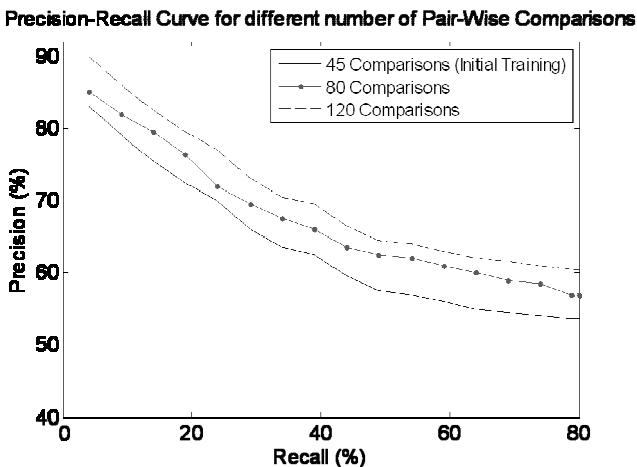


Fig. 4. The effect of the number of objects depicted to the user N on the system performance

In the following, we present experimental results that indicate performance of the proposed fusion algorithm that constraints inductive learning with spectral clustering.

Figs. 3 depicts the precision recall curve of the proposed algorithm compared with other approaches used for route selection, while Fig. 4 presents the precision-recall curve with respect to the number of objects depicted to the user, that is the number of pair-wise comparisons.

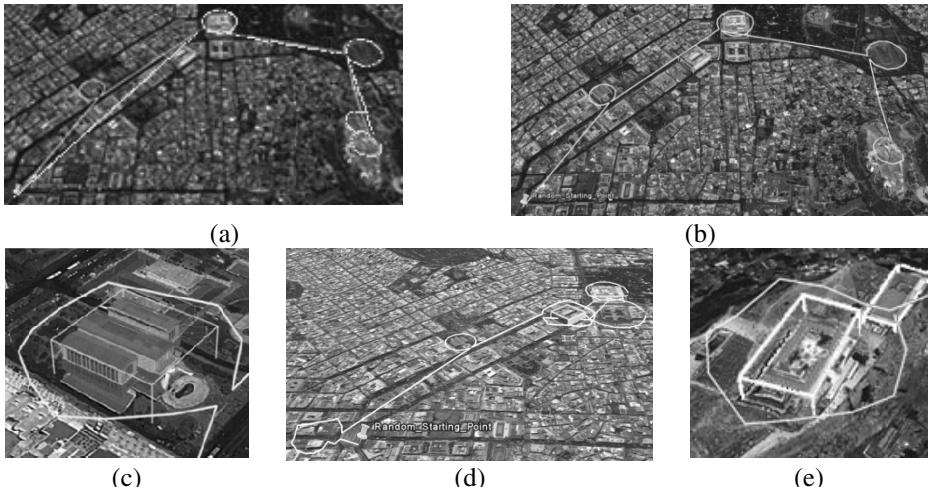


Fig. 5. (a) Generated path with the proposed algorithm compared with the method of [13], (b) Generated path using method of [13], (c,e) Zoom in two preferred 3D objects of the itinerary retrieved by the proposed genetic algorithm approach, (d) Generated path using Dijkstra shortest path algorithm.

Fig. 5 presents a subjective (qualitative) evaluation of the proposed genetic optimization strategy. Simulation results have been performed, according to the scenario that the user prefers archaeological places and buildings of neoclassical architectural style in his/her 3D navigation. In this figure we compare the results with the method of [13] and with the results of the shortest path using the Dijkstra's algorithm. In particular, the shortest path algorithm yields the shortest navigation route that connects SP and FP points. On the contrary, the proposed algorithm selects one building from the cluster with the green buildings (neo-classical buildings) and three from the cluster with the red ones (archeological sightseeing), as the user prefers more to visit archeological sightseeing than visiting neo-classical buildings. The path is presented with purple line on Fig. 5(a). On the other hand the according to approach of [13] includes in the itinerary objects that are most preferred by the user without any classification {yellow path Fig. 5(a,b)}. Fig. 5(c,e) zooms in two of the preferred objects of the itinerary extracted by the viewing algorithm.

As is observed, in the first case, the Temple of Olympian Zeus was selected as most appropriate as user prefer to see archeological sites more than visiting neoclassical buildings. The first building belongs to the cluster of archeological buildings which id more preferred by the user.

Acknowledgment . This paper is supported by the project E-Park, "Exploitation of new Technological Trends for payment and handling public parking" approved under the Interreg III Programme, Greek-Cypriot cooperation and funded from European Union and Greek National funds.

References

1. Chalvatzis, C., Virvou, M.: Fuzzy logic decisions and web services for a personalized geographical information system. In: Tsihrintzis, G.A., Virvou, M., Howlett, R.J., Jain, L.C. (eds.) New Directions in Intelligent Interactive Multimedia. SCI, vol. 142, pp. 439–450. Springer, Heidelberg (2008)
2. Nadi, S., Delavar, M.R.: Multi-criteria, personalized route planning using quantifier-guided ordered weighted averaging operators. Int. J. of Applied Earth Observation and Geoinformation, 322–3358 (2011)
3. Shapiro, L.G., Stockman, G.C.: Computer Vision. Prentice Hall (2011)
4. Doulamis, N., Doulamis, A., Varvarigou, T.: Adaptive Algorithms for Interactive Multimedia. IEEE Multimedia Magazine 10(4), 38–47 (2003)
5. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: A power tool for interactive content-based image retrieval. IEEE Trans. on CSVT 8(5), 644–655 (2008)
6. Bardis, G., Miaoulis, G., Plemenos, D.: User Profiling from Imbalanced Data in a Declarative Scene Modeling Environment. In: Plemenos, D., Miaoulis, G. (eds.) Artificial Intelligence Techniques for Computer Graphics. SCI, vol. 159, pp. 123–140. Springer, Heidelberg (2008)
7. Doulamis, A., Doulamis, N.: Generalized Non-Linear Relevance Feedback for Interactive Content-Based and Organization. IEEE Trans. on Circuits and Systems for Video Technology 14, 656–671 (2004)
8. Doulamis, N., Doulamis, A., Varvarigou, T.: Adaptive Algorithms for Interactive Multimedia. IEEE Multimedia Magazine 10, 38–47 (2003)
9. Niaraki, S.A., Kim, K.: Ontology based personalized route planning system using a multi-criteria decision making approach. Expert Systems with Applications, Science Direct, Experts systems with Applications 36, 2250–2259 (2009)
10. Zipf, A., Jost, M.: Implementing adaptive mobile GI services based on ontologies: examples from pedestrian navigation support. Comput. Environ. Urban Syst. 30, 784–798 (2006)
11. Reitter, D., Lebiere, C.: A cognitive model of spatial path-planning. Computational & Mathematical Organization Theory 16, 220–245 (2010)
12. Mekni, M., Moulin, B.: Hierarchical Path Planning for Multi-agent Systems Situated in Informed Virtual Geographic Environments. In: Second International Conference on Information, Process, and Knowledge Management, Saint Maarten, pp. 48–55 (2010) ISBN 978-1-4244-5688-8
13. Yiakoumettis, C., Doulamis, N., Miaoulis, G., Ghazanfarpour, D.: Active Learning of User's Preferences Estimation Towards a Personalized 3D Navigation of Geo-referenced Scenes. Springer (to appear)
14. Cohen, W., Schapire, R., Singer, Y.: Learning to Order Things. Journal of Artificial Intelligence Research 10, 243–270 (1999)
15. Luxburg, U.: A tutorial on Spectral Clustering. Journal Statistics and Computing 17, 395–416 (2007)
16. Doulamis, N., Kokkinos, P., Varvarigos, E.: Resource Selection for Tasks with Time Requirements using Spectral Clustering. IEEE Transactions on Computers 10.1109/TC.2012.222 (to be appeared)

Beat Synchronous Dance Animation Based on Visual Analysis of Human Motion and Audio Analysis of Music Tempo

Costas Panagiotakis¹, Andre Holzapfel²,
Damien Michel³, and Antonis A. Argyros^{4,3}

¹ Dept. of Commerce and Marketing, TEI of Crete, Greece
cpanag@csd.uoc.gr

² Universitat Pompeu Fabra, Barcelona, Spain
andre.holzapfel@upf.edu

³ Institute of Computer Science, FORTH, Crete, Greece
michel@ics.forth.gr

⁴ Computer Science Department, University of Crete, Greece
argyros@ics.forth.gr

Abstract. We present a framework that generates beat synchronous dance animation based on the analysis of both visual and audio data. First, the articulated motion of a dancer is captured based on markerless visual observations obtained by a multicamera system. We propose and employ a new method for the temporal segmentation of such motion data into the periods of dance. Next, we use a beat tracking algorithm to estimate the pulse related to the tempo of a piece of music. Given an input music that is of the same genre as the one corresponding to the visually observed dance, we automatically produce a beat synchronous dance animation of a virtual character. The proposed approach has been validated with extensive experiments performed on a data set containing a variety on traditional Greek/Cretan dances and the corresponding music.

1 Introduction

Synthesizing realistic human or animal motion is a very important research topic in computer animation with a high number of applications like virtual reality, computer games, movies and entertainment systems [1–3]. Motion synthesis algorithms usually should take into account several constraints in order to create realistic animations that are related with the virtual environment of animation. For example, to achieve realistic dance animation synthesis, the motion of the virtual character should synchronize with music. The rhythm of dance music can be considered to be based on several related periodicities. The periodicity which is the most convenient to cause a human to move his body is referred to as the beat. The problem of detecting the beat has recently attracted considerable research interest [1, 3, 4].

Usually, 3D motion dance data are available e.g. by capturing devices or by motion synthesis algorithms. So, a problem of a great interest is to synchronize them to a given dance music. Given proper and automatic synchronization, the

realism of the resulting audiovisual experience is increased. Such a capability is also expected to contribute to the development of important applications regarding the demonstration, study, teaching, spread and preservation of music and dances.

This paper addresses directly the aforementioned problem when the dance is periodic, meaning that it consists of repetitive motion patterns. The target case study considers traditional Cretan dances. Nevertheless, the adopted approach is applicable to a much broader class of dances. In order to solve the problem, we employ signal processing techniques for combined music and motion analysis, that is a vibrant and rapidly evolving field of research [5]. A growing trend in music and motion analysis is to tackle the problems globally and to exploit, whenever possible, the multimodal or multi-faceted aspects of music and motion.

A lot of research has been already devoted to the motion analysis of dance videos in order to estimate the rhythm of motion. In [6], a method of rhythmic information extraction from dance videos and music has been proposed. The rhythm of motion is estimated by the analysis of motion trajectories of points that are detected using an adaptation of the Shi-Tomasi (ST) corner detector [7]. Since the 2D visual information is not always sufficient to solve the problem of motion rhythm estimation with high accuracy, other methods [4] have been applied to 3D motion capture data. The method in [4] first detects rapid directional change of joints, estimating candidate beats and then transform this information to continuous motion signals using sequential cosine functions. Finally, the power spectrum density of signals is analyzed to estimate the dominant period.

In [8], two methods have been presented for segmenting periodic human motion capture data for mobile gait analysis. The first method is a model-based algorithm which operates directly on the joint angles detecting local minima and maxima. By considering pairs of successive minima and maxima, it is possible to identify distinct intervals in a periodic motion. However, this method suffers from some limitations, as it might still be possible to get conflicting segmentation sets between different joints. In such cases, deciding which joint produces more reliable results is a difficult task. The second method is a model-free, Latent Space algorithm, a dimensionality reduction method which first aggregates all the sensor data and transforms them into an 1D signal. Finally, the segmentation is given by the detection of local minima and maxima in the resulting 1D signal.

Beat detection in music aims at automatic estimation of the time instances where a human listener would tap his foot to the music. There have been several methods presented over the last decades. The approach of Klapuri [9] is widely considered as a state-of-the-art approach. While in the years after the publication of [9] quite innovative approaches were presented (*e.g.* [10]), no large improvement in general accuracy has been observed. As we aim at synchronising dance movement with audio signals in the context of traditional dances, we will apply the modification of the Klapuri method as proposed in [11]. This modification uses a signal representation derived from phase characteristics as input to a beat tracker similar to [9], which was shown to improve the alignment of the beat sequence to the audio signal in the case of traditional dances [11].

The beat detection in music has been used by several methods that aim to create new unseen dance animations that are synchronized with a given music [1, 3]. In [1], a fast, greedy algorithm analyzes a library of stock motions and generates new sequences of movements that were not described in the library. The greedy algorithm with backtracking tries to find the best matching frame among the closest dance moves, take it as a greedy choice and repeats the same process. A second, genetic algorithm tries to optimize the dance sequence by taking a number of valid random dance figures as a population and applies the genetic operators of crossover and mutation to create new generations. In [3], the generation of dance performances is based on a given musical piece by matching the progressions of musical and motion patterns and by correlating musical and motion features. The proposed method uses similarity matrices for musical and motion sequences and matched the progressions of musical and motion contents by minimizing the difference between the two similarity matrices.

Most of the existing approaches that try to provide a temporal segmentation of human motion are heuristic and use simplifications or signal approximations without any global optimality criterion. Many approaches based on visual information use 2D tracking data and suffer from visual limitations like occlusions and noise. In addition, many approaches can be only applied to simple human motions (e.g. walking), where the period can be defined by the local minima and maxima of the signal. Moreover, certain methods synthesize new unseen animations that are synchronized with a given music.

On the contrary, in this paper, instead of creating new unseen animations (that usually requires a high number of 3D motion datasets), we solve the problem of synchronizing the 3D motion of a given dance with a given music. We have proposed an optimization approach that computes the optimal solution for the problem of temporal segmentation of human motion using 3D dance motion data. An advantage of the proposed method is that it can be applied to complex multidimensional signals such as those representing dance movements. We also apply an autocorrelation based criterion in order to segment the music signal into periods. This information is then used to produce beat synchronous dance animations. The experimental results show that the proposed method achieves very promising results. It should be also noted that the input to the algorithm is not marker-based motion capture data but rather data produced by a home-build markerless human articulation tracking data. In that sense, the proposed approach is also capable of tolerating noise in the representation of human motion.

The rest of the paper is organized as follows. Section 2 gives a brief overview of the proposed approach. Sections 3, 4 and 5 present the details of the three main building blocks of the proposed method, that is, temporal segmentation of periodic human motion, music beat detection algorithm and beat synchronous dance animation creation, respectively. The experimental results are given in Section 6. Finally, a summary of this work and the main directions of future work are provided in Section 7.

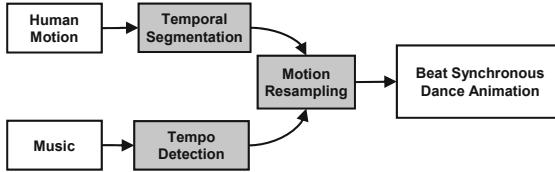


Fig. 1. Scheme of the proposed system architecture

2 Overview of the Proposed Approach

An overview of the proposed approach is illustrated in Fig. 1. The input to our method consists of (a) motion capture data, that is, the 3D position, orientation and articulation of the body parts of a human dancer while dancing a particular periodic dance and (b) the acoustic signal of a music that is compatible to that dance genre. The goal is to animate automatically a virtual/synthetic character who dances according to (a) but is also synchronized to the rhythm of (b). To achieve this result, the proposed approach employs three building blocks. The first one consists of an efficient algorithm for the temporal segmentation of the complex human motion capture data into the periods of dance. The second building block segments the acoustic signal into parts of duration equal to the music tempo period. Finally, a motion resampling algorithm is responsible for remapping each period of the motion capture data according to the estimated music tempo and for producing the beat synchronous dance animation. The following three sections present these building blocks in more detail.

3 Temporal Segmentation of Periodic Human Motion

The input to the temporal segmentation of human motion is the time series of the joint angles of an articulated human model. These joint angles are estimated by a recently proposed method [12] that relies on markerless, multicamera observations of a moving person. The employed method estimates accurately the parameters of an articulated human body model that has 11 joints and a total of 29 degrees of freedom.

Let $S \in \mathbb{R}^{m,n}$ be the given multidimensional signal of captured human motion that contains the time series of the $m = 29$ degrees of freedom (i.e., joint angle) of the human motion. Let also n be the number of temporal samples of each of these series. $S_i(j)$ denotes the j -sample of i -angle time series, $i \in \{1, \dots, m\}$, $j \in \{1, \dots, n\}$. Assuming that the human motion (dance in our case) is periodic, the goal of temporal segmentation is to segment S into its periods. Let $T_p = \{t_0, t_1, t_2, \dots, t_p\}$, $1 = t_0 < t_1 < t_2 < \dots < t_p \leq n$ be a temporal segmentation of S into p segments. For each such segmentation, we define the following energy function

$$E(T_p) = \sum_{i=1}^m \sum_{k=1}^{p-1} d(S_i(t_{k-1} : t_k - 1), S_i(t_k : t_{k+1} - 1)). \quad (1)$$

In Eq.(1), $d(.,.)$ denotes a function that computes the distance between the signal segments $S_i(t_{k-1} : t_k - 1)$ and $S_i(t_k : t_{k+1} - 1)$. In this work, the Pearson's distance [13] is used to implement $d(.,.)$. The Pearson's distance is defined based on Pearson's linear correlation coefficient $C(x, y)$ between the signals x, y , i.e.,

$$d(x, y) = 1 - C(x, y), \quad (2)$$

which is minimized when the signals' autocorrelation is maximized. Since the number of samples of each segment are not necessary equal, in order to estimate the autocorrelation between $S_i(t_{k-1} : t_k - 1)$ and $S_i(t_k : t_{k+1} - 1)$, $d(x, y)$ is estimated through a uniform resampling of the signal $S_i(t_k : t_{k+1} - 1)$, so that the resulting signal consists of $t_k - t_{k-1}$ samples.

According to our problem definition, it holds that the optimal temporal segmentation should minimize the energy $E(T_p)$ of Eq.(1). Thus, the temporal segmentation of the human motion amounts to estimating the segmentation T_p with this property.

An extra complication arises by the fact that in the application domain we consider, the duration of each period slightly changes over the time. This means that for all $i \in \{1, \dots, p\}$ there exists a small positive value α (e.g. $\alpha < 0.1$) such that

$$\frac{(i - \alpha) \cdot n}{p} \leq t_i \leq \frac{(i + \alpha) \cdot n}{p}. \quad (3)$$

The quantity $\frac{n}{p}$ corresponds to an upper bound estimation of the mean period of the motion signal. Then, the proposed method for temporal segmentation consists of two steps:

- Estimation of the number of periods p .
- Estimation of the optimal T_p under the assumption that the signal consists of p periods.

The number of periods p can be automatically computed by getting the global maximum of the amplitude signal of Fast Fourier Transform of a given motion signal. p can be also estimated by the music beat detection (see Section 4) under the assumption that the source music corresponding to the given dance motion signal is available. Alternatively, p can be given by the minimization of mean of $E(\hat{T}_s)$ over the periods, where \hat{T}_s denotes the uniform time segmentation into s periods. This means that the duration of each period is equal to $\frac{n}{s}$. We restrict the search space to periods between $0.25s$ and $1.25s$, which relates to the range of possible tempo in music. In notation,

$$p = \operatorname{argmin}_{s \geq 2} \frac{E(\hat{T}_s)}{s}. \quad (4)$$

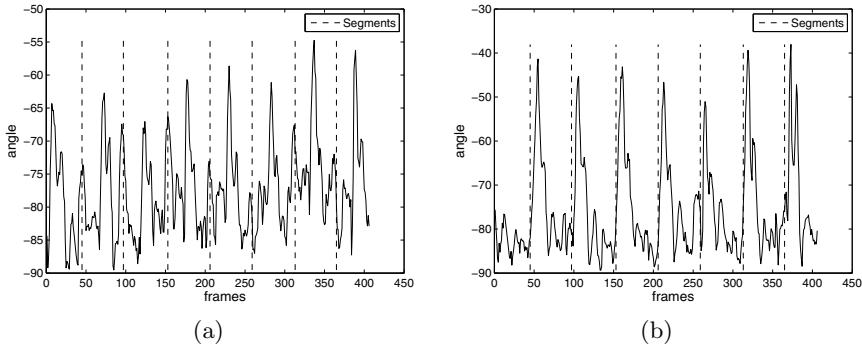


Fig. 2. The proposed temporal segmentation of the left and right knee angles

Having estimated the number of periods p (see Eq. (4)), the goal of the proposed method is to find the segmentation T_p that minimizes $E(T_p)$ under the constraint of Eq.(3). Let $D(u, v)$, $v > u$ be a metric that measures how periodic is the signal segment that corresponds to the time interval $[u, v]$. $D(u, v)$ takes its minimum value of zero if the segment from u to v is periodic. According to the definition of $E(T_p)$ (see Eq.(1)), $D(u, v)$ is given by

$$D(u, v) = \min_{-\alpha \leq \psi \leq \alpha} \sum_{i=1}^m d(S_i(u : v - 1), S_i(v : w)), \quad (5)$$

where $w = v - 1 + \lceil (1 + \psi) \cdot (v - u) \rceil$. In order to reduce the computation cost (see also the constraint of Eq. (3)) $D(u, v)$ is computed according to Eq. 5 only when $\frac{(1-2\alpha)\cdot n}{p} \leq v - u \leq \frac{(1+2\alpha)\cdot n}{p}$. Otherwise, $D(u, v)$ is set to ∞ .

Then, we construct a graph G as follows. A node $u \in \{1, \dots, n\}$ of G corresponds to the time instance u . The weight of edge $u \sim v$ is given by $D(u, v)$. Finally, we add the virtual node $n + 1$ that is connected with the last time instances $\{\lfloor n - \alpha \cdot \frac{n}{p} \rfloor, \dots, n\}$ with an almost zero weight, since we don't know the end of the signal periods. Then, the global minimum of $E(T_p)$ under the constraint of Eq.(3) is given by the sum of weights of the shortest path between the nodes 1 and $n + 1$. The nodes (time instances) of the shortest path correspond to the optimal solution for the problem of signal segmentation. Using the Dijkstra's algorithm [14], the time complexity for the shortest path computation is $O(\log(N) \cdot E)$, where N and E are the number of nodes and edges of G , respectively. This cost can be reduced to $O(N \log(N))$, since $E = O(N)$ due to the constraint of Eq.(3).

Figure 2 illustrates the proposed temporal segmentation using as input the angles of the left and right knees of a dancer dancing the traditional Cretan dance “Siganos” (see Section 5). As it can be observed, although the signals are quite complex and exhibit some small differences in period synchronization, the proposed method successfully segments both of them.

4 Music Beat Detection

As our main focus lies upon the synchronization of movement data to a music signal of a traditional dance, we apply a beat tracking algorithm that was tailored towards specific properties of the music signals at hand [15]. The central aspect of the method presented in [11] is the usage of group delay defined as the derivative of the phase spectrum over frequency. The method takes advantage of the fact that the average group delay, also referred to as phase slope function, can provide insight about the position of impulses that are caused by note onsets. The phase slope function is used to obtain a signal representation, which emphasizes time instances of instrument note onsets and can be used to track the beat in a music signal. The parameters of the phase slope computation are the same as those presented in [11]. Onset candidates are determined separately in four frequency bands, which results in four band-wise onset signals $\mathbf{y}_c[n], c = 1 \dots 4$.

For the estimation of beat times from the band-wise onset signals, an algorithm based on the method proposed by Klapuri et al. [9] has been used. The algorithm first determines a tempo trajectory for a piece of music, and then aligns a sequence of impulses having a period related to that tempo to the music signal. The tempo trajectory is obtained by computing a weighted sum of $\mathbf{y}_c[n]$

$$\mathbf{y}[n] = \sum_{c=1}^4 (6 - c) \mathbf{y}_c[n] \quad (6)$$

and then, by weighting $\mathbf{y}[n]$ with the spectral flux at each sample n :

$$\mathbf{y}_{flux}[n] = \mathbf{y}[n] \sum_{\omega} HWR(|X(\omega, n)| - |X(\omega, (n-1))|). \quad (7)$$

In Eq.(7), HWR denotes a half wave rectification and $X(\omega, n)$ denotes the (short time) Fourier transform of the signal as used in the group delay computation. In order to obtain a set of tempo periods, the sample autocorrelation of $\mathbf{y}_{flux}[n]$ is computed in rectangular windows of $t_{win} = 8s$ length with a hop size of 1s. The obtained sequence of autocorrelation vectors describes the development of rhythmic content over the duration of a piece. In the following, the tempo periods β have been estimated using a *Hidden Markov Model*(HMM) as described in [9]. This results in a sequence of beat period estimations $\beta[k]$, with $k = 1 \dots N$, with N being the number of autocorrelation vectors for a piece.

In order to align a beat pulse with the signal, we compute the likelihood of an alignment phase $\Phi[k]$ in analysis frame k

$$P(\hat{\mathbf{r}}_{\mathbf{y}\mathbf{y}} | \Phi[k] = l) = \sum_{c=1}^4 (6 - c) \sum_{n=0}^{8f_o} \tilde{\mathbf{y}}_k[n + l] \mathbf{y}_c[(k - 4)f_o + n] \quad (8)$$

where $\tilde{\mathbf{y}}_k$ is a reference pulse train of $t_{win}f_o + 1$ samples length, having an impulse at the middle position and a period equal to $\beta[k]$. Thus, just like in the estimation of the beat period, an eight second length window has been used. The weighted sum of the band wise correlations as computed in (8) is then used in an HMM framework as suggested in [9].

Table 1. The number of frames for the beat synchronous dance animations

$MSig_1$	$MSig_2$	$MMal_1$	$MSyr_1$	$MSyr_2$
794 - 827	682 - 710	570 - 571	388 - 450	388 - 400

5 Beat Synchronous Dance Animation

Having segmented the given human motion and music into periods, the next task is to create a beat synchronous dance animation. To achieve this, we resample the motion signal so that it becomes equal to the target music tempo. More specifically, let $T_p = \{t_0, t_1, t_2, \dots, t_p\}$ and $T'_p = \{t'_0, t'_1, t'_2, \dots, t'_p\}$ be the temporal segmentations of the human motion and target music signals, respectively. In order to get beat synchronous dance animation, the i -segment $[t_i, t_{i+1}]$ of the motion signal should be resampled with $r_i = \frac{t'_{i+1} - t'_i}{t_{i+1} - t_i}$ oversampling rate. If $r_i = 1$, then there are no changes in the resulting motion signal. In order to avoid rapid changes on sampling rate on the borders of segments, a continuous oversampling rate $r(t)$ at time $t \in [t_i, t_{i+1}]$ of the motion signal can be used, which is defined as

$$r(t) = \frac{\sum_{k=-1}^1 w(t, i+k) \cdot r_{i+k}}{\sum_{k=-1}^1 w(t, i+k)}, \quad (9)$$

where $\delta = \frac{t_p - t_0}{p}$ is equal with the mean period of motion signal. In Eq.(9), $w(t, k) = \exp\left(\frac{-2 \cdot (t - (t_{i+k} + t_{i+k+1})/2)^2}{\delta^2}\right)$. In addition, the application of $r(t)$ keeps the animation beat synchronous.

6 Experimental Results

The proposed method has been evaluated on a data set consisting of several traditional Greek/Cretan dances. A professional dancer has performed a variety of such dances. His performance was recorded by a fully calibrated and synchronized multicamera system operating at 25Hz. Three types of dances were investigated, namely “Siganos”, “Maleviziots” and “Syrto” which are danced in 6, 8 and 12 steps, respectively. Like most of traditional Cretan dances, a theme can be repeated with a practically infinite number of variations. The articulated motion of the dancer was tracked with a recently proposed method [12], which estimates the 3D position, orientation and full articulation of the human body based on the markerless observations provided by the camera system.

We employed motion recordings of Siganos ($MSig_1, MSig_2$), Maleviziots ($MMal_1$) and Syrtos ($MSyr_1, MSyr_2$). The number of temporal samples of $MSig_1, MSig_2, MMal_1, MSyr_1, MSyr_2$ is 750, 638, 658, 445 and 417, respectively.

Each of these recordings has been synchronized using two audio recordings of different tempos. We have used two audio recordings for each dance type: Siganos ($ASig_1, ASig_2$), Syrtos ($ASyr_1, ASyr_2$) and Maleviziots ($AMal_1, AMal_2$), respectively. Table 1 presents the number of frames of the beat synchronous dance

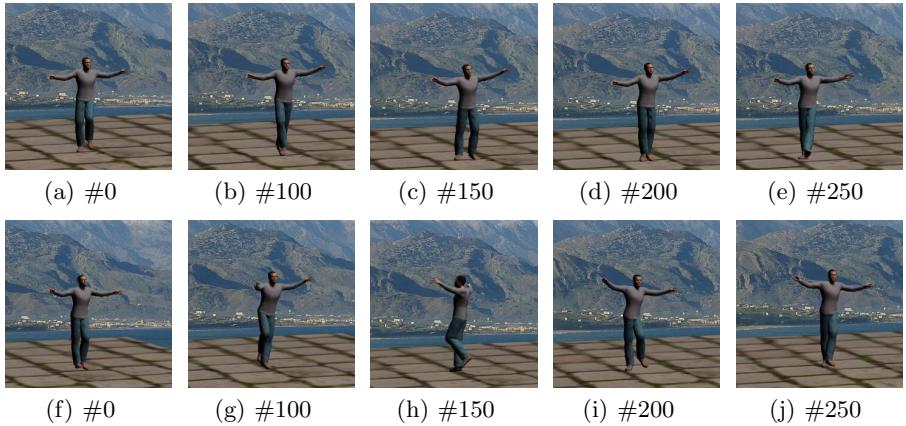


Fig. 3. ((a) - (e)) Frames of synchronized dance animation for $MSig_2$ under $ASyr_1$. ((f) - (j)) Frames of synchronized dance animation for $MSyr_2$ under $ASyr_2$.

animations using the two corresponding audio recordings. So, the number of frames of $MSig_1$ with $ASig_1$ and $ASig_2$ is 794 and 827, respectively. Figures 3(a)-3(e) and 3(f)-3(j) show sample frames where the motion of $MSig_2$ was aligned to the music of $ASig_1$ and the motion of $MSyr_2$ was aligned to the music of $ASyr_2$, respectively. A more complete set of video results containing ten beat synchronous dance animation videos can be downloaded at <http://alturl.com/64ihx>. As it can be verified, the proposed framework provides beat synchronous dance animations of good quality, in all tested cases.

7 Conclusions

In this work, we proposed a framework that generates beat synchronous dance animation combining complex human motion capture data with an audio signal of a target music. The proposed approach has been successfully tested on variety of dances containing cyclic activities such as traditional Greek/Cretan dances. The proposed method yields the optimal solution for the problem of temporal segmentation into periods of multidimensional signal applied on complex human motion data such as dance movements. Regarding future work, we plan to apply the temporal signal segmentation to other types of periodic signals (e.g. ECG or geophysical signals) in order to segment them into periods. Moreover, the proposed method can be used as a last part of a motion synthesis scheme. Thus, by providing audio input only, the envisioned system will be able to provide new unseen motions of beat synchronous realistic animations of a virtual dancer.

Acknowledgments. This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS-UOA-ERASITECHNIS.

References

1. Alankus, G., Bayazit, A.A., Bayazit, O.B.: Automated motion synthesis for dancing characters. *Computer Animation and Virtual Worlds* 16, 259–271 (2005)
2. Panagiotakis, C., Tziritas, G.: Snake terrestrial locomotion synthesis in 3d virtual environments. *The Visual Computer* 22, 562–576 (2006)
3. Kim, J.W., Fouad, H., Sibert, J.L., Hahn, J.K.: Perceptually motivated automatic dance motion generation for music. *Computer Animation and Virtual Worlds* 20, 375–384 (2009)
4. Kim, T., Park, S., Shin, S.: Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (TOG)* 22, 392–401 (2003)
5. Essid, S., Richard, G.: Fusion of multimodal information in music content analysis. *Multimodal Music Processing* (2012)
6. Chu, W., Tsai, S.: Rhythm of motion extraction and rhythm-based cross-media alignment for dance videos. *IEEE Transactions on Multimedia*, 1 (2012)
7. Shi, J., Tomasi, C.: Good features to track. In: *CVPR*, pp. 593–600 (1994)
8. Valtazanos, A., Arvind, D., Ramamoorthy, S.: Comparative study of segmentation of periodic motion data for mobile gait analysis. In: *Wireless Health 2010*, pp. 145–154. ACM (2010)
9. Klapuri, A.P., Eronen, A.J., Astola, J.T.: Analysis of the meter of acoustic musical signals. *IEEE Trans. on Audio, Speech, and Language Processing* 14, 342–355 (2006)
10. Böck, S., Schedl, M.: Enhanced Beat Tracking with Context-Aware Neural Networks. In: *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx 2011)*, Paris, France (2011)
11. Holzapfel, A., Stylianou, Y.: Beat tracking using group delay based onset detection. In: *Proc. of ISMIR - International Conference on Music Information Retrieval*, pp. 653–658 (2008)
12. Michel, D., Panagiotakis, C., Argyros, A.A.: Tracking human body articulations with multiple rgbd sensors. Technical report, FORTH-ICS (2013)
13. Fulekar, M.: Bioinformatics: applications in life and environmental sciences. Springer (2009)
14. Dijkstra, E.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959)
15. Holzapfel, A.: Similarity methods for computational ethnomusicology. PhD thesis, University of Crete, Greece (2010)

Combining Unsupervised Clustering with a Non-linear Deformation Model for Efficient Petroglyph Recognition

Vincenzo Deufemia and Luca Paolino

DISTRA, Università di Salerno, Via Giovanni Paolo II, 84084 Fisciano(SA), Italy
`{deufemia, lpaolino}@unisa.it`

Abstract. Petroglyphs are prehistoric engravings in stone unrevealing stories of ancient life and describing a conception of the world transmitted till today. In the current paper we consider the problem of developing tools that automate their recognition. This is a challenging problem mainly due to the high level of distortion and variability of petroglyph reliefs. To address these issues, we propose a two-stage approach that combines unsupervised clustering, for quickly obtaining a raw classification of the query image, and a non-linear deformation model, for accurately evaluating the shape similarity between the query and the images of the more appropriate classes.

1 Introduction

Petroglyphs are descriptions of the ancient real life depicted by means of basic tools onto rock surfaces by prehistoric people. Most of them are engravings representing (or better we believe that represent) some kind of symbolic or ritual language and describe a simple vision of the world to transmit the future generation. Unfortunately, the nature of these representations do not allow to preserve them in optimal and controlled environments such as museum and petroglyphs are endlessly exposed to weathering as well as vandalism of careless or malicious visitors. For this reason, it is necessary to find some alternative methods that permit to protect or in some way do not lose such demonstrations of the past real life and pass on them towards the future.

The objective of the IndianaMAS project is to develop a system able to “digitally protect” and conserve the rock art natural and cultural heritage sites, by storing, organizing, and presenting petroglyphs in a systematic way [7, 11, 12]. At the basis of this project, an important role is played by the algorithm for the classification of the petroglyphs on the basis of their shapes aiming to identify similar petroglyphs stored in different archives [6]. Unfortunately, classifying petroglyphs is a challenging task due to the high level of variability in the drawing that was carried out by cast percussion [17, 21].

In past researches, in order to overcome this problem and provide an adaptable means to perform good matches between images with high variability some models have been developed [8]. The Image Deformation Model (IDM) presented in [9] was successfully applied to handwritten character recognition and shown good retrieval performance in the medical automatic annotation task [5]. IDM computes the

distances by each pixel of an image and the corresponding pixels of the second images and every pixels of the second image located within a certain warp range by taking into account the surrounding pixels (local context). According to our opinion, this method could be successfully adapted to our problem since it is less sensitive to local changes that often occur in the presence of symbol variability. Unfortunately, the time complexity of this approach makes this choice not particularly suitable for interactive retrieval tasks [19].

In this paper we propose a two-step process for improving the IDM performance by means of SOM clustering [10]. Basically, in the first step the SOM works as a discriminative classifier that analyzes the symbols, classifies them by exploiting the Shape Context [1], and produces a first classification. In the second step, the set of images associated to the more relevant classes previously obtained are compared one-to-one to the image query by an IDM. The idea is to use a raw classifier for obtaining a list of possible solution classes quickly, then to apply a more precise but slower algorithm, as IDM, for refining the results. We demonstrate, on an image database of 1530 petroglyph symbols from Mount Bego rock art site belonging to 17 classes that the time performance of the recognition system can be considerably increased without significantly affect the accuracy results.

The outline of the paper is organized as follows. Some background information and a discussion of related work are presented in Section 2. The approach we propose is then described in Section 3, while Section 4 presents the outcome of the experimental evaluation. Finally, Section 5 introduces the conclusion of this work.

2 Background and Related Work

2.1 Rock Art

With the term of Rock art, archeologists indicate any human-made markings made on natural stone. Basically, with this terms they want to specify the, in most part, the petroglyphs, this is to say symbols created on top of stones by removing part of a the surface by incising, picking, carving, or abrading. They are very popular in many parts of the world but especially in Africa, Italy, Scandinavia, Siberia, southwestern North America and Australia [15].

In particular, the petroglyph site of Mont Bego located between Italy and France is very important for the community because the richness of the place in both qualitative and quantitative. For this reason, we based the following experiment on the symbols collected and catalogued from it. Archaeologists consider this place as an incredibly valuable source of knowledge, due to the up to 40,000 figurative petroglyphs and 60,000 non-figurative petroglyphs scattered over a large area at an altitude of 2,000 to 2,700 meters. Between 1898 and 1910 Clarence Bicknell realized up to 13,000 drawings and reliefs, part of which were then published in [2]. From the Bicknell's classification process, it is possible to identify seven types of figures: horned figures (mainly oxen), ploughs, weapons and tools, men, huts and properties, skins and geometrical forms [3]. As an example, Figure 1(a) shows a typical petroglyph representing a horn form, while on Figure 1(b) the same example digitized by the de Lumley's team.

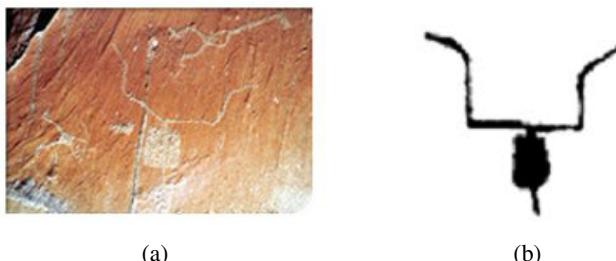


Fig. 1. The picture of an engraving representing a horn form (a), and the digitalized relief made by de Lumley's team (b)

2.2 Image Processing of Petroglyphs

In the past years several algorithms, techniques, applications, and formulae have been proposed for the recognition of symbols. Differently, petroglyph recognition was only minimally touched by this human force, likely because some unique properties, which render most of them unsuitable for computational tasks [21].

A first work in the field of petroglyph recognition was proposed in [18] with the aim of cataloguing the petroglyphs based on the parts of animal bodies, and relations among them. In [20] Takaki *et al.* presented a new method to characterize shapes of the petroglyphs and the properties of the group they belong to. On the basis of these new approach the skeleton of the petroglyph is firstly extracted by applying several image processing algorithms. Successively, the shape is specified in terms of elementary symbols even for allowing quantitative comparison.

Zhu *et al.* presented a tool based on the crowdsourcing, which allows human volunteers to “help” computer algorithms to segment and annotate petroglyphs [14, 21]. They also introduced a distance measure and algorithms based on the Generalized Hough Transform that allow data mining of large collections of rock art images [22]. Lastly, in [17] Seidl and Breiteneder presented the outcome of their experiment about their segmentation algorithm of rock art images.

3 The Two-Stage Classification Process

The process proposed for the classification of petroglyph symbols is composed of two stages: coarse-grained classification through unsupervised clustering and fine-grained image matching with a nonlinear deformation model.

3.1 SOM Clustering

The initial step of the classification process is the generation of the Shape Context descriptors [1] from the binary images of the symbols. However, to recognize a petroglyph symbol regardless of its size and position, the input image is normalized to a standard size by translating its center of mass to the origin. Then, to increase tolerance

to local shifts and distortions we smooth and downsample the feature images. In particular, to ensure that small spatial variations in the symbol correspond to gradual changes in the feature values, we apply a rotationally symmetric Gaussian lowpass filter. We then downsample the images by performing symbol removing and resizing (see Figure 2).

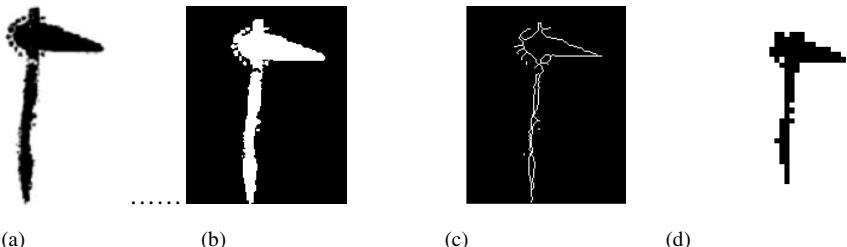


Fig. 2. An example of normalization of a petroglyph representing an halberd. (a) the original image, (b) the smoothed image, (c) the point removed image, and (d) the resulting resized image.

The proposed clustering approach uses the Shape Context descriptors to describe the shape of petroglyph symbols [1]. The shape context belongs to the category of methods which measure the shape similarity on the basis of the point correspondances. The method identifies n points on the shape contour and, then, connects each point p_i to all the other points through $n-1$ vectors. The resulting vector set is not yet a good descriptor but may be used for successive analysis. The key idea is that the distribution over relative positions is a robust, compact, and highly discriminative descriptor. So, considering each point p_i , it is possible to create an histogram by aggregating the relative distances of the remaining $n - 1$ points using the following formula,

$$h_i(k) = \text{count}(q \neq p_i : (q - p_i) \in \text{bin}(k))$$

The relative distances are not aggregated in a normal space but in a uniform log-polar space. Figure 3 shows an example of descriptors obtained by applying the shape context method on a halberd petroglyph symbol.

Although the pairwise comparison of the descriptors may be very successful, its computational cost may result excessive when the dataset makes bigger. Thus, the descriptor vectors obtained by the Shape Context are clustered to reduce the number of different descriptors to compare. Symbols are then indexed by computing the occurrences of the descriptors in each cluster.

The clustering is computed by means of Self-Organizing Maps that organize the clusters in two-dimensional topologically ordered feature maps [10]. In particular, SOM is a type of artificial neural network that performs clustering by means of unsupervised competitive learning. SOM reduces the size of the symbol set into few patterns called codebook vectors. The SOM attempts to order, while simultaneously generating, the codebook vectors from the training inputs. The aim of the ordering process is to arrange the codebook vectors such that those with common features are positioned geometrically close to one another. At the conclusion of the ordering

process codebook vectors, which are most similar, should be adjacent to one another and those least similar at maximal distance, while operating within the constraints of the geometry of the map, often a 2D grid.

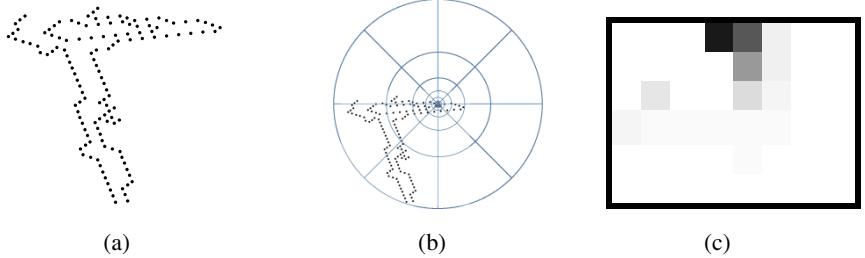


Fig. 3. (a) represents the set of points extracted on the edges of a halberd petroglyph. (b) shows the log polar diagram obtained by overlapping the halberd shown in (a). (c) shows the shape context for a point in the halberd

In the proposed system, at the end of the training process, the symbol S_i is represented as a vector V_i of size C (the number of SOM units) whose elements contain the occurrences of the Shape Context descriptors in each cluster. According to the vector space model, the indexed dataset is normalized with the *tf-idf* weighting schema obtaining for S_i the vector $V_i = (v_{1,i}, v_{2,i}, \dots, v_{c,i})$ with

$$v_{i,j} = \frac{freq_{i,j}}{\max_k(freq_{k,j})} \log(N/n_i)$$

where $freq_{i,j}$ is the number of occurrences of the descriptors belonging to cluster c_i in the symbol S_j , N is the total number of petroglyph symbols, and n_i is the number of petroglyph symbols containing Shape Context vectors into c_i .

Once the SOM map is trained, the similarity measure between a query and a representative symbol is performed by means of the cosine similarity model. Basically, the cosine similarity is defined by the following formula:

$$sim(V_q, V_i) = \frac{\sum_{j=1}^C v_{j,q} v_{j,i}}{|V_q| |V_i|}$$

where $V_q = \{v_{1,q}, v_{2,q}, \dots, v_{c,q}\}$ is the image symbol representation of the query obtained counting the number of Shape Context descriptors fallen into each cluster and, $V_i = \{v_{1,i}, v_{2,i}, \dots, v_{c,i}\}$ is the symbol class representation.

3.2 Image Deformation Model

The successive step of the approach consists of the one-to-one comparison with the images belonging to the first classes obtained by the SOM classification. To this aim, we use a deformation model that is robust to distortions and local shifts. In particular, the image deformation model (IDM) performs a pixel-by-pixel value comparison of the query and reference images determining, for each pixel in the query image,

the best matching pixel within a region around the corresponding position in the reference image.

The IDM has two parameters: warp range w and context window size c . The algorithm requires each pixel in the test image to be mapped to a pixel within the reference image not more than w pixels from the place it would take in a linear matching. Over all these possible mappings, the best matching pixel is determined using the $c \times c$ local gradient context window by minimizing the difference with the test image pixel.

In particular, the IDM distance D between two symbols S_1 (the query input) and S_2 (the template) is defined as:

$$D^2 = \sum_{x,y} \min_{d_x d_y} \|S_1(x + d_x, y + d_y) - S_2(x, y)\|^2$$

where d_x and d_y represent pixel shifts and $S_i(x, y)$ represents the feature values in S_i from the patch centered at x, y .

Figure 4 illustrates how the IDM works and the contribution of both parameters, where the warp range w constrains the set of possible mappings and the $c \times c$ context window computes the difference between the horizontal and vertical gradient for each mapping.

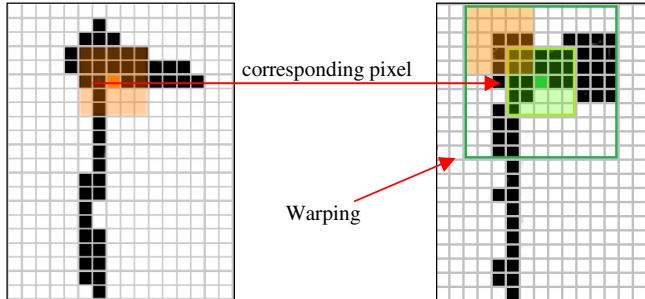


Fig. 4. Example of areas affected by the comparison of pixels with IDM, where $w=3$ and $c=2$. The query pixel context (indicated by the orange area in the query image) is compared with each equal-sized rectangle within the warping area (dark-green rectangle of the reference image). The warping area is calculated by building a $m \times m$, with $m=(w+c)^2+1$, square around the corresponding reference pixel (dark-green pixel)

The algorithm considered in this paper includes the early termination strategy proposed in [18]. This optimization does not degrade the result quality and has been demonstrated to speedup of a factor of 4-5.

4 The Experiment

In this section, we provide a description of the experiments we performed to evaluate the improvement, in terms of accuracy and time performances, achieved by applying SOM clustering to prune not relevant classes.

In the first part of this section we provide a description of the dataset of petroglyphs we used for the experiments and how we divided them into folds. In the second part, we describe the configuration parameters of the algorithms composing the proposed approach. Finally, we present and discuss the results achieved in the experiments

4.1 Dataset

In order to perform our experiments we created a dataset based on the petroglyphs collected by de Lumley's team on Mount Bego and published in [4]. The basic dataset consists of 51 images from 17 classes (3 images for each class). To obtain a larger dataset, every image has been computed by a deformation algorithm, which rotates, translates, and skews the petroglyph images. Thus, for each original image we generated 30 deformed images obtaining 1530 images in total.

The dataset has been processed during the experiments by using a modified version of the k -fold approach [16], with $k = 3$. In particular, for 3 times a set of $m=17$ images - one for each class - have been chosen from the 51 original images, and the corpus of $N = 1530$ images has been split into two subsets: a subset of $[N - (m * 30)] = 1020$ images was used for training, while m extracted images were left out for testing.

4.2 Settings

The system parameters have been set to maximize the recognition likelihood. In particular, to find the right learning parameters for the recognizer we used a cross-validation setup where parameters have been set in the following way:

- *Image resizing*: 32 pixels. It represents the size of the normalized images classified by the SOM;
- *SOM units*: 200. It indicates the number of clusters of the SOM structure;
- *Warp range*. We evaluated the algorithm using the value of 3 pixels for the first IDM configuration setting and 4 px for the second one;
- *Local context*. This value was set to 2 pixels for the first setting and 3 for the second.

4.3 Results and Discussion

Table 1 shows the results of the conducted experiments. The table is organized showing in the first row the different approach settings: *No SOM* in case results are obtained without a preliminary SOM clustering, *Best 3*, *Best 5*, and *Best 7*, in case the IDM computes results on the basis of the best 3, 5, and 7 classes obtained by SOM, respectively. The second and the third rows shows the performance of the algorithm expressed in terms of efficacy and efficiency, namely the rate of correct classes on the amount and the minutes to compute results, for the configuration setting $w=3$ and $c=2$. The fourth and the fifth rows show the results for the setting $w=4$ and $c=3$.

Table 1. The efficacy and efficiency results achieved by applying SOM and IDM algorithm with different configuration settings

	No SOM	Best 7	Best 5	Best 3
Efficacy 3x2	69.3%	65.3%	64.6%	58.5%
Efficiency 3x2	150	86	71	56
Efficacy 4x3	74.6%	68.6%	68%	63%
Efficiency 4x3	266	141	115	84

Results prove the retrieval quality of the proposed approach. The best performances, as expected, are obtained by applying IDM to all images of the dataset. However, it is worth to note that good precision values are already obtained by considering the first 5 classes provided by SOM.

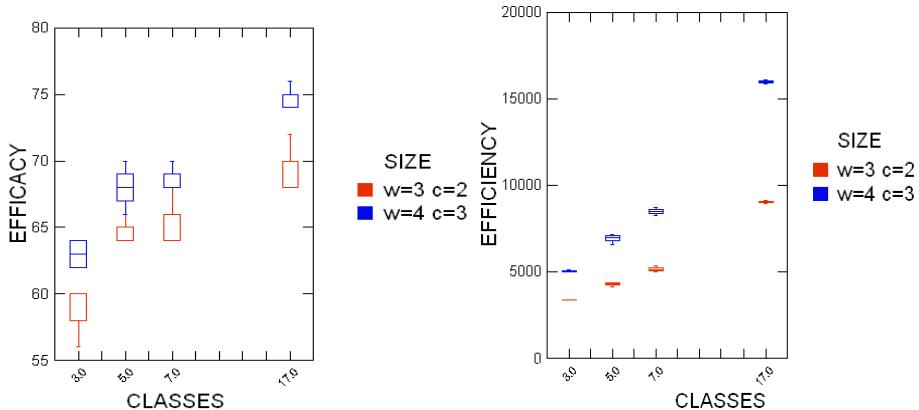


Fig. 5. (a) The trend of the accuracy performances. The y-axis indicates the precision rate. (b) The trend of the time performances. The y-axis indicates the time (in sec.) for executing the algorithm

As a matter of fact, as for 3x2 settings, the increment between 5 classes and 17 classes is only 4.7%, while for 4x3 is 6.6%. This claim may be inferred also by considering their slopes in the box plots shown in Fig. 5(a). As for 3x2, the angular coefficient is 0.4 in the range [5,17], and 3.05 in the range [3,5]. This is to say, initially the precision has a fast improvement till 5 classes, then the improvement get slower till reaching the maximum number of classes. The same consideration holds for 4x3, in this case, the angular coefficient is 0.55 in the range [5,17], and 2.5 in [3,5].

With respect to the simple classification using the SOM clustering, this is a significant improvement. As a matter of fact, the precision performance we obtained by means of the approach is much higher of the simple 33% obtained using the Shape Context descriptors within SOM.

By considering time performances, the box plot in Fig. 5(b) shows that both 3x2 and 4x3 configuration settings the slopes are similar in the ranges [3,5] and [5,17]. Considering both time and precision, it is possible to deduce that good performances

in terms of precision are obtained also considering the first 5 classes produced by the SOM clustering. It implies that, in case of 4x3, with respect to the 266 minutes required to perform all the IDM comparisons we can obtain, more or less, the same results only in 115 minutes. A similar consideration may be done for 3x2, in fact in this case, we can obtain good performance in 56 minutes instead of the 150 minutes required to compare all the images.

5 Conclusion

The image deformation model represents an effective measure to compute similarity between distorted images. In the literature several optimizations have been introduced in order to reduce the retrieval time without degrading the result quality [19]. However in order to efficiently analyze dataset of thousands of images further optimization strategies are needed. In this paper we presented a method for improving the classification of petroglyph images with IDM based on SOM clustering. We tested the approach on a dataset of petroglyphs derived from [4] and compared the achieved results with those obtained using IDM only. Results have shown that by applying the IDM algorithm only to the images belonging to the first five classes obtained by the SOM classification, we got great improvements in terms of speed while preserving a satisfying precision in terms of accuracy.

A significant improvement is also obtained with respect to the simple SOM classification based on Shape Context descriptors. Indeed, precision performance passes from 33% to 64,6% if five classes are considered.

References

1. Belongie, S., Malik, J., Puzicha, J.: Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 509–521 (2002)
2. Bicknell, C.M.: A Guide to the Prehistoric Rock Engravings in the Italian Maritime Alps. Tip. G. Bessone (1913)
3. Chippindale, C., Bicknell, C.: Archaeology and Science in the 19th Century. *Antiquity* 58, 185–193 (1984)
4. de Lumley, H., Echassoux, A.: The Rock Carvings of the Chalcolithic and Ancient Bronze Age from the Mont Bego Area. The Cosmogonic Myths of the Early Metallurgic Settlers in the Southern Alps. *L'Anthropologie* 113, 969–1004 (2009)
5. Deselaers, T., Weyand, T., Keysers, D., Macherey, W., Ney, H.: Fire in ImageCLEF 2005: Combining content-based image retrieval with textual information retrieval. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) *CLEF 2005. LNCS*, vol. 4022, pp. 652–661. Springer, Heidelberg (2006)
6. Deufemia, V., Paolino, L., de Lumley, H.: Petroglyph Recognition using Self-Organizing Maps and Fuzzy Visual Language Parsing. In: *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2012*, pp. 852–859. IEEE (2012)

7. Deufemia, V., Paolino, L., Tortora, G., Traverso, A., Mascardi, V., Ancona, M., Martelli, M., Bianchi, N., de Lumley, H.: Investigative Analysis Across Documents and Drawings: Visual Analytics for Archaeologists. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI 2012, pp. 539–546. ACM (2012)
8. Keysers, D., Deselaers, T., Gollan, C., Ney, H.: Deformation Models for Image Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 1422–1435 (2007)
9. Keysers, D., Gollan, C., Ney, H.: Local Context in Non-linear Deformation Models for Handwritten Character Recognition. In: Proceedings of International Conference on Pattern Recognition, ICPR 2004, pp. 511–514. IEEE (2004)
10. Kohonen, T., Schroeder, M.R., Huang, T.S. (eds.): *Self-Organizing Maps*, 3rd edn. Springer-Verlag New York, Inc., USA
11. Mascardi, V., Deufemia, V., Malafronte, D., Ricciarelli, A., Bianchi, N., de Lumley, H.: Rock art interpretation within Indiana MAS. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) *KES-AMSTA 2012. LNCS*, vol. 7327, pp. 271–281. Springer, Heidelberg (2012)
12. Mascardi, V., Briola, D., Locoro, A., Grignani, D., Deufemia, V., Paolino, L., Bianchi, N., de Lumley, H., Malafronte, D., Ricciarelli, A.: A Holonic Multi-Agent System for Sketch, Image and Text Interpretation in the Rock Art Domain. *International Journal of Innovative Computing Information and Control* 1 (2014)
13. Paolino, L., Sebillio, M., Tortora, G., Vitiello, G., Laurini, R.: Phenomena - A Visual Environment for Querying Heterogeneous Spatial Data. *Journal of Visual Languages and Computing* 20, 420–436 (2009)
14. Petroannotator,
<http://www.cs.ucr.edu/~qzhu/papers/CAPTCHA/PetroAnnotator/>
15. Petroglyphs, <http://en.wikipedia.org/wiki/Petroglyph> (last accessed June 28, 2013)
16. Salzberg, S.L.: On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. *Data Min. Knowl. Discov.* 1, 317–328 (1997)
17. Seidl, M., Breiteneder, C.: Detection and Classification of Petroglyphs in Gigapixel Images - Preliminary Results. In: Proceedings of the 12th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, VAST 2011, pp. 45–48. Eurographics Association (2011)
18. Sher, Y.A.: *Petroglyphs in Central Asia*. Nauka (1980) (in Russian)
19. Springmann, M., Dander, A., Schuldt, H.: Improving Efficiency and Effectiveness of the Image Distortion Model. *Pattern Recognit Letters* 29, 2018–2024 (2008)
20. Takaki, R., Toriwaki, J., Mizuno, S., Izuhara, R., Khudjanazarov, M., Reutova, M.: Shape analysis of petroglyphs in central Asia. *Forma* 21, 243–258 (2006)
21. Zhu, Q., Wang, X., Keogh, E., Lee, S.-H.: Augmenting the Generalized Hough Transform to Enable the Mining of Petroglyphs. In: Proceedings of the KDD 2009, pp. 1057–1066 (2009)
22. Zhu, Q., Wang, X., Keogh, E., Lee, S.-H.: An Efficient and Effective Similarity Measure to Enable Data Mining of Petroglyphs. *Data Mining and Knowledge Discovery* 23, 91–127 (2011)

Analysing User Needs for a Unified 3D Metadata Recording and Exploitation of Cultural Heritage Monuments System

E. Maravelakis¹, A. Konstantaras¹, A. Kritsotaki²,
D. Angelakis², and M. Xinogalos³

¹ Technological Educational Institute of Crete

² Foundation for Research & Technology – Hellas

³ Astrolabe Engineering

Abstract. This research paper aims to address the problem of lack of a unified system for 3D documentation, promotion and exploitation of cultural heritage monuments via complete 3D data acquisition, 3D modeling and metadata recording using terrestrial laser scanners. Terrestrial laser scanning is a new fast developing technology that allows for the mapping and exact replication of the entire 3D shape of physical objects through the extraction of a very large number of points in space (point cloud) in short time periods, with great density and precision, and with no actual physical contact with the object of interest. The problem lies on the various types of hardware equipment and software systems used in the whole workflow of the 3D scanning process, including for the extraction of point clouds and the building process of the computerized 3D model development and the final products presentation. These often results in a large volume of interim and final products with little if no standardization, multiple different metadata, various user-dependent annotation requirements and vague documentation which often casts repeating a certain process impossible. This paper presents a user requirement analysis for a complete metadata recording during the whole lifecycle of a 3D product, aiming at supporting workflow history and provenance of 3D products of cultural heritage monuments.

Keywords: 3D modeling, terrestrial laser scanning, 3D model annotation, ancient and cultural heritage digital documentation, metadata recording.

1 Introduction

The creation of 3D models for ancient monuments is a not an easy task, due to the difficulty of creating a dimensionally accurate 3D model and a 3D virtual representation of the monument. Common practice for surveying monuments include non-automated procedures, using conventional measurement methods such as measuring tapes or - at best - total station equipment. The result in this case is not a complete 3D model of the monument but a synthesis of measurements from target points focused on 2D drawings and orthophotos of the monument.

The rapid growth of light detection and ranging (LIDAR) technology enabled the near instant gathering of a vast volume of uniformly distributed 3D points' measurements recording information such as orientation in space, distance from the source to name but a few. This capability was embraced by terrestrial laser scanners which are capable of providing highly accurate 3D images enabling designers to experience and work directly with real-world conditions by viewing and manipulating rich point-clouds in CAD software, benefiting further by the many possible outputs available from point-clouds and basic measurements to ortho-images, derived 2D/3D drawings, meshing/surfacing and solid modeling.

Because of these advantages terrestrial laser scanners have been introduced in order to enhance the development of 3D models of monuments and the promotion of cultural heritage [1-5]. Every monument can be replicated with the outmost quality and accuracy of detail, with no limitations in terms of position, orientation, shape, size or accessibility as long as they are directly visible and within scanning range of the terrestrial laser scanner. This ensures the retrieval, from the entire surface of the object of interest, of the necessary primary information for its detailed and precise geometric documentation and modeling.

The process for scanning, data processing and documentation of objects of cultural heritage undergoes several stages before producing the final three dimensional model of the initially scanned object. During these stages all the involved parties (field engineers, scanned-data processing users, archeologists, civil and architectural engineers, etc.) follow different steps and directions for the transition from one interim stage to another. Each step obeys its own workflow, which might encompass various software and hardware applications of multiple parameters that change dramatically depending on the object, the requested type of representation of the final 3D model and its final extracts [6-8]. It is often the case at the end of a stage of processing data to have to return to initial data, change some of their parameters and repeat the process in order to improve the final product. Also it is not uncommon during interim processing stages to realize that additional data must be obtain in order to produce a better final product.

This lack of metadata recording during 3D scanning ,3D modeling of interim and final product and incomplete or absent documentation have stalled the infiltration of such a powerful technology in the field of recording, digital preservation and promotion of cultural heritage. This paper investigates this problem and analyses the different user requirements for a modern, user-friendly and flexible system for documenting and exploiting the whole process of 3D scanning.

2 Procedure for the Recording and Documentation of 3D Monuments

A solution to the above problems should ensure the recording of metadata during the process of obtaining data, eg. during terrestrial laser scanning, both regarding the scanned data themselves as well as the overall process executed to obtain them. Equally as important is the recording of metadata during the various processes of scanned data processing, both for the produced results as well as the various

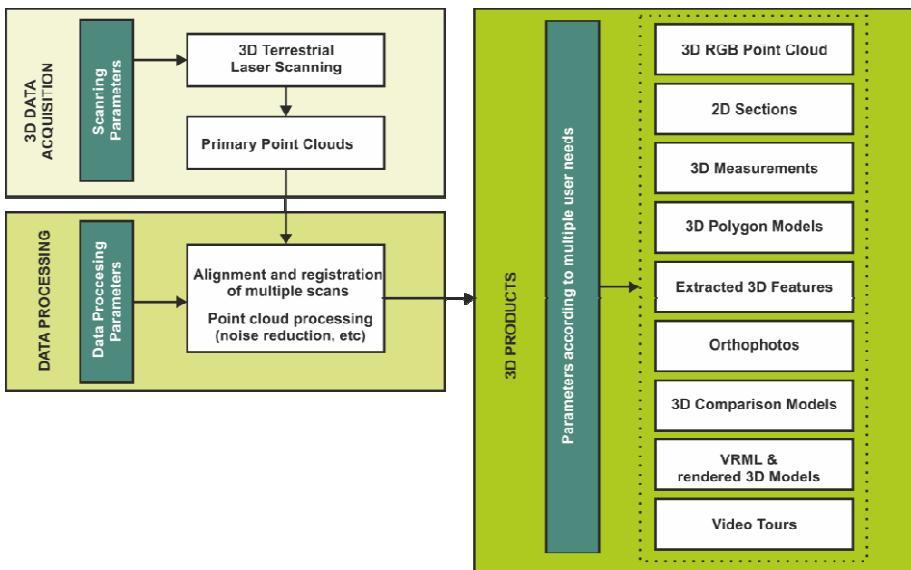


Fig. 1. A typical workflow of a 3D scanning process

parameters/variables used to produce them. This is important in order to fully characterize a process, return to initial parameters if necessary, and repeat a process under the very same conditions or with selective amendments.

Also during the entire both processes that of obtaining data through terrestrial laser scanning and that of 3D-data processing it is an important necessity the ability to import comments (annotation) [9], both regarding the process as well as the produced extracts (e.g. 3D models) or selected parts of theirs, such as a pillar in a temple or the gown of a philosopher or a knight in an ancient statue, etc.

A typical workflow of a 3D scanning process includes 3 phases: 3D data acquisition, 3D data processing presented and the exploitation & promotion of the derived 3D products (figure 1). 3D laser scanning is a method for fast collection of topical metrical data from a distance (non destructive methodology) allowing the mapping and recording the overall three-dimensional geometrical shape of all in range objects as well as chromatic and photographic information. The product of the scanning process is a metrical point cloud with additional chromatic information per point (x,y,z,i) or (x,y,z,r,g,b) for each various scanning position on any pre-specified coordinate system depending on the application. Multiple scans from various locations are most often necessary to obtain a representative point cloud of the region of interest. Typical objects of interest might be objects of cultural, archeological, architectural interest, objects of geological geotechnical interest, large scale manufacturing, mining areas and heavy industrial mechanical objects, forensic objects, etc.

To produce a 3D model from laser scanning several processes must be followed outlined next (figure 2).



Fig. 2. Production of a 3D model from Terrestrial Laser Scanning

- Evaluation of the object of interest, what is its nature, where is it and what are its dimensions.
- The purpose of the scan, is every little detail necessary or just a skeleton design is needed.
- Type of products.
- Meta-data required (especially in cartographic and topographic applications these are a necessity under the INSPIRE directive [10]).
- Annotation requirements.
- Scaling and precision for the design and 3D printing of the required objects.
- Completeness – density analysis to ensure satisfactory resolution.
- Chromatic requirements to ensure satisfactory RGB representation of the object of interest when necessary.
- Selection of suitable co-ordinate system of reference.

Then it is important to specify exactly how to execute the scanning process, having taken under serious consideration the following outlined issues:

- Selection of equipment including a suitable type of laser scanner as discussed earlier.
- Scanning positions and spherical cover of the object of interest.
- Resolution, quality and time of scanning exposure per scanning position.
- Way of scanning, step'n'stare or profile scan, per scanning position.
- Alignment methodology, technical markers, physical markers, ICP algorithms, etc.
- Georeference techniques, if necessary, encompassing GPS or geodetic stations.
- Techniques for color yielding, such as gray-scaled intensity, RGB from internal or external camera, etc.

After all these considerations are taken into account, the scanning process occurs under the following general steps, which may vary slightly depending on the each individual laser scanner:

- Selection and stationary of the scanning positions.
- Placement of technical markers upon and/or amongst the object of interest

- Scanning initiation.
- Assembling of the laser scanner, controls and cameras if required and placement on the scanning locations' stationeries.
- Selection of mode and scanning parameters (resolution, quality, field of view, etc.)
- Repeat of scanning at every scanning position.
- Gathering of topographic measurements for georeference (if necessary)

This is followed by an initial processing of the collected raw data. The initial scan detects "unwanted" information that is other objects beside the object of interest, which are being removed manually. The resulting point cloud is then filtered to reduce artificial noise caused by the surface of the object (rough or smooth) which may stray the incoming laser beams. These filters take into account parameters such as maximum distance, average distance and mean square deviation of the beam travel to reduce artificial noise. In the case of multiple scans the resulting point clouds are being brought to the same scale and orientation so as they all much the same coordinate system. This process commonly known as cloud alignment or registration can be done with either using common to all scans artificial markers or through georeference of known topographically recorded physical objects. Further 3D data processing conclude to a 3D model that can be imported to several computer aided design software CAD tools that can handle 2D and 3D representations allowing for the extraction of greater features, such as front views, top views, sections, as well as 3D viewing allowing material texture representation.

The scope of this paper is to investigate, analyze and define the processes and the metadata necessary for the production of 3D models of objects of archeological interest and cultural heritage and establish a framework upon which a software solution could rely upon.

To establish such a framework the following actions were taken: Thorough description of the terrestrial laser scanning process in a typical process of 3D documentation. Study of 3D modeling procedures using the scanned-data. Recording of the requirements for the overall or partial annotation of 3D models for the as complete as possible digital documentation of ancient monuments or objects of cultural heritage. Recording of the existing standardization for three dimensional objects' modeling in combination with a study of their use and scope of existence. Study of the existing standardization of metadata related to the work-flow leading to the production of three dimensional models. Study of the users' needs regarding the proposed system with respect to operational specifications and user-system interaction.

3 Metadata Recording and Annotation Requirements during the Whole 3D Product Lifecycle

3D data acquisition and 3D data processing are the two main events in the lifecycle of a 3D cultural heritage monument product. General description metatada should include the 3D project title, the name of company/institution which is involved and the starting and ending dates of the project.

3.1 Metadata Requirements Analysis 3D Data Acquisition Event

During the 3D data acquisition event the following metadata should be recorded:

a) Data for the acquisition event including: The type of the 3D data acquisition process (contact – non contact scanning, laser – phase scanning, stereophotography, dome based etc). The person in charge for the 3D scanning including his team and his role (Field engineer, 3D modeling & processing expert, Researcher, Archeologist, Art historian – museum expert, Monument Conservator, trainee or student). Starting and Ending dates. Monument location. Type of monument including name, code number, size, texture and georeferencing requirements.

b) Data for the Setup & Devices including: Name of device. Type of device (Terrestrial laser scanner, phase scanner, Structured-light 3D scanner). Other device data (Serial number, producer, firmware etc). Name, version and type of software used. Configuration & calibration files). Ability to add more than one devices

c) Output Digital Objects produced from the 3D data acquisition event

3.2 Metadata Requirements Analysis 3D Data Process Event

During the 3D data process event the following metadata should be recorded:

a) Input digital object used (coming from the data acquisition event)

b) Data for the 3d data process event including: The type of process (Reverse Engineering, 3D modeling, 3D content creation, 3D point cloud processing, coding, Image processing, video processing, Processing with Geographic Information System). The name, version and the type of software used. Processing parameters. The person involved and its role (field engineer, 3D modeling & processing expert, Researcher, Archeologist, Art historian – museum expert, Monument Conservator, trainee or Student)

c) The Output Digital Objects produced from the 3D data process event and their type (3D point cloud with/without geo-referencing, 3D model with/without texture, 3D sections, 3D measurements, Vector drawings, 3D comparison – Quality, orthofotos, VRML &3D rendered models, Video)

3.3 3D Model Whole and Partial Annotation Requirements Analysis

Throughout the duration of the 3D scanning process and the processing of the obtained data towards the formation of a 3D model it is important to be able to add comments regarding both the process as well data, metadata, interim and final products, either for the overall object or for partial features of the latter (figure 3).

A considerable part of metadata information has to do with the description of the object of interest and it is often the case that this data are readily available and stored in the database of a related institute, eg. a museum or an archeological agency, etc. Typical information describing objects for 3D scanning may include basic information used to identify it, such as a title, an identifier , the object dimensions, internal and external measurements, a classification based on its shape and geometry,

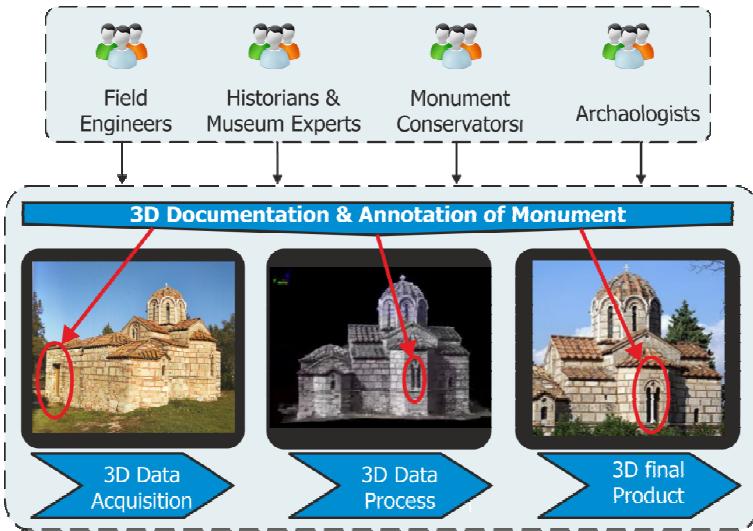


Fig. 3. Annotation requirements during the whole 3D scanning process

the materials used for its construction, information that could lead to the approximation of its date of origin, basic colour, colour variations due to degradation, type of use, current condition, intact or partial remains, additions or subtractions of parts over the eons, current institute, collection or place of display, inscriptions and their translations, etc.

Further annotation is necessary to meet the needs of art historians and museum experts depending on the target of their research. The use of an annotation system should be simple and flexible for all scientists with little if any experience or specialized knowledge of the discussed software. Typical annotation requirements from this target group may include measurement dimensions and comparisons with other similar objects, construction materials with reference to eras of use, colours and their deviation at various eras as well as potential symbolisms of theirs, interpretations of possible usage types, current or historical, resulting from literature references or other sources, current condition and suggestions on how to preserve, location of display and exhibition conditions, other related objects, etc.

Another group of people that should make good use of an annotation tool could be monument conservators whose comments arise for every monument by treating it as a distinct object. In that respect, annotation categories for that purpose should ensure the ability to re-evaluate the monuments status, the degradation of the condition in the years to come and relate it to its current or past condition. This is important as certain enhancements could improve the hosting environment and also help conservators specify the most suitable methods of preservation. Annotation capabilities for this target group of scientists should cover: the extent of damages and their depths with abilities of precise point to point measurements, colour peeling, cracked paint, over-colouring and colour thickness, conservation work employed on specific parts of the

object, materials used for conservation (if any), dating of repairs, signatures, scripts and inscriptions, marks, vandalisms, objects' comparison, filing of all preservation work to date, etc.

Last but not least we are discussing perhaps the most important target group of scientists that could make extensive use of an annotation facility, the archeologists. The scientific background of an archeologist demands precision of representation and access to specialized information. Furthermore, scientific annotation must often be related to sources, literature, references, experts' opinions, photographs, videos and all sorts of existing documentation materials related to a monument. Annotation demands for that group of scientists focus upon: the structural description of an object fully or partially and the relation of various parts, chromatic and schematic analysis and inner relations, conclusions on dating and the provenance of the monument (e.g. identification of a local workshop that produced it), typology, , technotropy-style comparison to other monuments, aesthetics analysis that distinguishes a prototype from a replica, conclusions on the potential use of an object emerging from its shape, size, material and construction technique, which may yield a pointer on the era of use, identification and analysis of decoration patterns that can be used to the identification, classification and dating of the object, dimensions measurements and comparison with analogues, information regarding the surrounding space now and then if the object was moved from the place of its discovery, descriptions of the current condition of an object with respect to its initial state, realizations regarding previous interventions on the object and their effect on its initial status, identification of various construction phases of a monument dated to various periods of time and characterized by different technotropy-styles, etc.

4 Discussion, Conclusions and Future Work

Nowadays, a completed infrastructure that allows the systematic creation of 3D models by managing related metadata of different sources and setups that maintains them for reuse has not been yet presented to the best of our knowledge.

For this reason we believe there is a need for a system that will integrate the 3D documentation, promotion and exploitation of cultural heritage monuments via complete 3D data acquisition, 3D modeling and metadata recording.

For the metadata we propose to use (and extend) CIDOC-CRM (ISO/FDIS 21127).: an extendable core ontology [11-12] that allows the integration of more complete and specialized models and system terms beyond the core model. The integration of information could be achieved by using a mechanism of annotation and referencing that allows the linking of 3D models and its metadata with all available sources and knowledge bases [13]. Thus, we achieve the easy access, (re)use and maintenance of 3D models. The system will be based on the following components:

The **Integrated Repository**: capable of storing, managing complex objects that were produced in any of the phases described above (acquisition, production etc) and metadata concerning their provenance, and workflow history. The repository will be based on two separate components the Object-repository (OR) and the Metadata-repository

(MR). The OR contains all the content files (different classes of binary datasets): 3D models, 2D-Images and any kind of digital document (text, multimedia, etc). All types of factual semantic data, i.e., metadata on provenance, object description etc., annotations and co-reference information, are stored in the MR.

The **Query Manager (QM)**: a component responsible to redirect the specific parts of a query to the respective components of the integrated repository: OR and MR, by splitting it into the corresponding parts. The tool will enable intuitive querying, based on relationships and classes that can be combined/simplified into: Fundamental Categories (Actors, Things, Events, Place, Time, Type) are connected by using interpretations of “Fundamental Relationships” (refers to, is from, has part, is similar, has met, at), but also on keyword search.

A set of tools for storing, retrieving and annotating objects:

Deposit-Tool: The tool will guide the user to manually store sufficient and consistent metadata, based on event-centric [14] descriptions (3D data acquisition, 3D data processing events).

Browser-Tool: The tool will be devoted to searching, querying, browsing, and viewing multimedia objects stored in the repository, based on the Fundamental Categories in relation to the Fundamental Relationships, but also on keyword search.

Annotation-Tool: The tool will help the user to segment and annotate objects stored in the repository.

Acknowledgements. The authors wish to thank the General Secretariat for Research and Technology of Ministry of Education and Religious Affairs, Culture and Sports in Greece for their financial support (via program Cooperation: Partnership of Production and Research Institutions in Small and Medium Scale Projects, Project Title: “3D-SYSTEK - Development of a novel system for 3D Documentation, Promotion and Exploitation of Cultural Heritage Monuments via 3D data acquisition, 3D modeling and metadata recording”).

References

1. Maravelakis, E., Bilalis, N., Mantzorou, I., Konstantaras, A., Antoniadis, A.: 3D modelling of the oldest olive tree of the world. *IJCER* 2(2), 340–347 (2012)
2. Kersten, T., Lindstaedt, M.: Virtual Architectural 3D Model of the Imperial Cathedral (Kaiserdom) of Königslutter, Germany through Terrestrial Laser Scanning. In: Ioannides, M., Fritsch, D., Leissner, J., Davies, R., Remondino, F., Caffo, R. (eds.) *EuroMed 2012*. LNCS, vol. 7616, pp. 201–210. Springer, Heidelberg (2012)
3. Manferdini, A.M., Remondino, F.: Reality-Based 3D Modeling, Segmentation and Web-Based Visualization. In: Ioannides, M., Fellner, D., Georgopoulos, A., Hadjimitsis, D.G. (eds.) *EuroMed 2010*. LNCS, vol. 6436, pp. 110–124. Springer, Heidelberg (2010)
4. Tapete, D., Casagli, N., Luzi, G., Fanti, R., Gigli, G., Leva, D.: Integrating radar and laser-based remote sensing techniques for monitoring structural deformation of archaeological monuments. *Journal of Archaeological Science* 40, 176–189 (2012)

5. Maravelakis, E., Andrianakis, M., Psarakis, K., Bolanakis, N., Tzatzanis, G., Bilalis, N., Antoniadis, A.: Lessons Learned from Cultural Heritage Digitisation Projects in Crete. In: Proceedings of the 14th International Conference on Virtual Systems and Multimedia, pp. 152–156 (2008)
6. Workflow Management Coalition Workflow Standard: Workflow Process Definition Interface – XML Process Definition Language. Technical report, Workflow Management Coalition, Lighthouse Point, Florida, USA (2002),
<http://www.wfmc.org/standards/docs.htm>
7. Zhiming, Z., et al.: Scientific workflow management: between generality and applicability. In: Proc. the 5th International Conference on Quality Software, Melbourne, Australia, pp. 19–20 (2005)
8. Workflow management coalition (2005), <http://www.wfmc.org>
9. Sinclair, P., Addis, M., Choi, F., Doerr, M., Lewis, P., Martinez, K.: The Use of CRM Core in Multimedia Annotation. In: First International Workshop on Semantic Web Annotations for Multimedia (2006)
10. Infrastructure for Spatial Information in Europe. INSPIRE Architecture and Standards Position Paper (2002)
11. Doerr, M.: The CIDOC CRM – An Ontological Approach to Semantic Interoperability of Metadata. AI Magazine 24 (2003)
12. Doerr, M., Theodoridou, M.: CRMdig: A generic digital provenance model for scientific observation. In: 3rd USENIX Workshop on the Theory and Practice of Provenance, Heraklion, Crete, Greece, pp. 20–21 (2011)
13. Theodoridou, M., Tzitzikas, Y., Doerr, M., Marketakis, Y., Melessanakis, V.: Modeling and Querying Provenance by Extending CIDOC CRM. Distributed and Parallel Databases (2010)
14. Doerr, M., Kritsotaki, A.: Documenting events in metadata. In: The e-volution of Information Communication Technology in Cultural Heritage, pp. 56–61 (2006)

Precise 3D Reconstruction of Cultural Objects Using Combined Multi-component Image Matching and Active Contours Segmentation

Christos Stentoumis¹, Georgios Livanos², Anastasios Doulamis²,
Eftychios Protopapadakis², Lazaros Grammatikopoulos³, and Michael Zervakis²

¹ National Technical University of Athens, Photogrammetric Lab, Zografou, Athens

² Technical University of Crete, Image processing and Computer Vision Lab, Chania

³ Technological Education Institute of Athens, Depart. of Topography, Aegaleo, Athens

Abstract. Cultural and creative industries constitute a large range of economic activities. Towards this expansion we need to state the inclusion of ICT technologies, as such of 3D reconstruction methods. However, precise 3D reconstruction under a computationally affordable manner is a research challenge. One way to precisely reconstruct a cultural object is through the use of photogrammetry with the main goal of finding the correspondences between two or more images to reconstruct 3D surfaces. A cultural object is often surrounded by visual background data that should be excluded to improve 3D reconstruction accuracy. Background conditions dynamically change, especially if the object is captured under outdoor conditions, where many occlusions occur and the shadows effects are not negligible. In this paper, we propose a combine image segmentation and matching method to yield an affordable 3D reconstruction of cultural objects. Image segmentation is performed on the use of active contours while image matching through novel multi-cost criteria optimization functions. Experimental results on real-life ancient column capitals indicate the efficiency of the proposed scheme both in terms of performance efficiency and cost.

1 Introduction

Surveys indicate that cultural and creative industry represents 4.5% of total European GDP and for 3.8% of the workforce [1]; so, this sector of economy can constitute one of the main European engines for growth and job creation. A crucial driving force for the development and economic growth of the creative industries is ICT technologies, which can open new frontiers in growing areas of the creative sector. 3D reconstruction and modeling of tangible cultural objects constitutes a significant task in digitalization area. Our world is a 3D world, and we perceive most of the events occurring in this world by depth information. One way to precisely reconstruct a cultural object is through the use of photogrammetry with the main goal of finding the correspondences between two or more images to reconstruct 3D surfaces. Image matching remains an active research field since finding a unique match or no match at all (occlusions, light

variations, geometry distortion) is in fact an ill-posed problem, which can be solved only if suitable constraints are set.

However, a cultural object is often surrounded by visual background data that should be excluded to improve 3D reconstruction accuracy. Background conditions dynamically change, especially if the object is captured under outdoor conditions, many occlusions occur and the shadows effects are not negligible. Thus, new sustainable and innovative computer vision tools should be investigated for automating the capturing technology, able to maximize performance, computational complexity while maintaining the precision in 3D reconstruction.

1.1 Previous Works

The collection of data for 3D modeling can be done by using *passive* methods, e.g. *shape from X* (X: stereo, shade, focus etc.), and /or *active* methods as time-of-flight (ToF), phase shift technology, or structured light scanners. Although, 3D point clouds can be successfully created by using range cameras, the approaches based on this technology are not able to capture the texture of the scene [2]. In [3] a system based on KinectTM is proposed to bridge this gap.

On the other hand, matching techniques have been proposed in the literature for describing the dissimilarity between potentially corresponding pixels. [4] evaluates the cost function itself under different optimization schemes in a thorough survey of stereo-methods. Non-parametric image transformations, such as *rank* and *census* [5], produce robust results based on relationships of pixels with their neighborhood. In [6] a dissimilarity measure was proposed to cope with differences in image sampling. Recently, the *mutual information* approach has been proposed for effectively handling radiometric differences [7]. Here, a combination of methods is proposed; the benefits of this approach are thoroughly discussed in [8]. Local approaches of stereo-matching are based on the definition of pixel *neighborhood*. [9] discusses the *support region* formation, where the costs are aggregated to enhance the cost of \mathbf{p} . A variety of methods exist to enhance this *cost aggregation* step; in [10] weights are attributed in a constant-sized neighborhood around each pixel in accordance with color similarity and geometric proximity; shiftable windows change the position of the central pixel of a support region [11]; and shape-adaptive windows can be based on separate circular sectors across multiple directions around a pixel [12].

To improve 3D reconstruction efficiency image *segmentation* techniques are exploited. Image segmentation extracts attributes of interest considering common properties, such as discontinuities and similarities, within different object classes. Several approaches have been introduced in literature; *point and line detection* techniques where the detected edges are linked in order to accurately represent the shape of each object, *thresholding methods* (histogram, adaptive, multi-level) that divide the image into segments according to distinct bands of pixel intensities and *region growing/splitting methodologies* which iteratively classify neighboring pixels of "seed-points" into a region through appropriately selected similarity criteria [13].

1.2 Our Contribution

In this paper, we propose an innovative 3D reconstruction methodology for cultural heritage objects, by combining state of the art image matching algorithms with novel image segmentation techniques. Initially the image segmentation algorithm is applied on the 2D data to precisely localize object boundaries in color domain. This way, we remove noise in the 3D reconstruction space since the matching algorithm focuses only on the cultural object of interest instead of exploiting the entire imagery plain. Simultaneously, computational cost is significantly reduced since the disparity space is limited on the foreground object. For image segmentation, we use active contours that evolve curves on color domain to obtain the segmentation. Selection of this segmentation technique among many others existing in the literature is due to its efficiency as regards precise localization of the object contours and its robustness in illumination variations since our approach faces outdoor image content. As far as 3D reconstruction is concerned, in this paper, we adopt a methodology that combines state of art techniques and novel considerations regarding a multi-component cost function, an adaptive support region and geometrically constrained 3D smoothing over the cost volume. Experimental results on real-life cultural heritage objects like ancient column capitals of Acropolis reveal the effectiveness of the proposed combined methodology in automatically and simultaneously precisely reconstructed tangible 3D cultural objects from high resolution images.

2 Active Contour Based Curve Evolution

In our case, the input images constitute of a column capital bedded on a platform placed on an outdoor environment. Such images usually contain textures, noise, shading, abrupt variations of lighting, background of numerous and overlapping items, thus introducing additional limitations to segmentation techniques and preventing them from converging to accurate object boundaries [14]. As a first step, a transformation to a different color space is required. The RGB model appears adequate for digital representation but unproved for color segmentation. HSV (Hue-Saturation-Value) and Lab (Luminance - a-b (color-opponent dimensions)) models decorrelate the pixel intensity from the pure color components, facilitating the detection of specific color bands, while in the RGB space color information originates from the combination of all the channels (Red, Green, Brown). In addition, the application of adaptive thresholding to the S-V and a-b channels is capable of isolating the foreground object from the background scene. Another limitation arisen through the segmentation procedure is the shading of the objects of interest. To handle this, histogram equalization of the pure color bands is used.

Once the image defects have been subdued through the preprocessing procedure, an appropriate segmentation algorithm must be selected. Contour-based techniques are well established in international bibliography, providing accurate and robust results even in noisy environment, having the drawback of suffering from initialization, local minima and stopping criteria problems. For this reason, we select these approaches in this paper. The principle of these techniques lies on the linking of edge

points extracted via an edge detection scheme, attempting to exploit curvilinear continuity to iteratively approximate the borders starting from a closed curve [15].

Then, active contours object detection is applied, by combining curve evolution techniques, level sets and the Mumford-Shah functional, accomplishing to detect corners and any topological change as in [16]. The model begins with a contour in the image plane defining an initial segmentation and then this contour gradually evolves according to a level set method until it meets the boundaries of the foreground region. According to the model, a curve S is represented via a function φ (the level-set function) as $S=\{(x, y)|\varphi(x, y)=0\}$, where (x, y) are coordinates in the image plane while the evolution of the curve is given by the zero level curve at time t of function $\varphi(x, y, t)$. Negative values denote points outside the curve while positive values originate from points belonging to the internal area of the curve, as depicted in Fig.1.

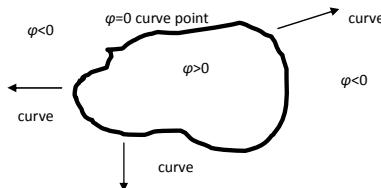


Fig. 1. An example of the proposed method used for curve evolution

At any given time, the level set function simultaneously defines an edge contour and a segment being evolved according to the partial differential Eq. (1), iteratively converging to a meaningful segmentation of the image.

$$\frac{\partial \phi}{\partial t} = |\nabla \phi| F, \phi(x, y, 0) = \phi_o(x, y) \quad (1)$$

where F denotes the speed of the curve evolution.

3 Multi-component Cost Function Image Matching

The matching process discussed in this paper is wrapped in a hierarchical scheme. Processing of high resolution images is necessarily based on scaled representations of the stereo-pair. The aim is to limit the disparity search space to a computationally feasible range, and also guide matched disparities in a coarse-to-fine context through scale-space. This approach also reveals structures in different layers of image pyramids, which lead from a rough, yet close to reality, 3D surface to finer detail as one proceeds through the image pyramid.

The three matching costs that form the complete matching function C are: the census transformation on image gradients, C_c (expressed through the Hamming distance), the absolute difference in colour values, C_{ADc} , and the absolute difference on principal image gradients, C_{ADg} . A robust exponential function [17] which resembles a Laplacian kernel [see Eq. (2)], has been preferred to model these costs

$$C(x, y, d) = 1 - \exp\left(-\frac{C_c}{\lambda_c}\right) + 1 - \exp\left(-\frac{C_{ADc}}{\lambda_{ADc}}\right) + 1 - \exp\left(-\frac{C_{ADg}}{\lambda_{ADg}}\right) \quad (2)$$

The terms of C have the quality of truncating costs that are too large, thus preventing outliers from pervading the matching cost. The values of C_c , C_{ADc} and C_{ADg} are also scaled in the same value field, assuming suitable selection of respective regularization factors λ , since each term of C takes values in the field [0, 1], and thus C is always positive. The impact of each dissimilarity measure on the overall cost can be tuned by adjusting the values of each λ .

Census on Intensity Principal Derivatives: To implement census cost C_c we first evaluate the census transformation T_C being a non-parametric image transformation [5]. For a support neighborhood $N_{m \times n}$ of a pixel \mathbf{p} , a binary vector forms a map of neighbouring pixels with intensities $I(\mathbf{p})$ less than that of \mathbf{p} . Unlike usual approaches, in the present implementation the transformation is performed not on grey-scale image intensity function I , but on its principal derivatives $\partial I / \partial x$, $\partial I / \partial y$ [8] providing an extended binary vector

$$T_c(p) = \bigotimes_{p \in \left\{ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right\}} \bigotimes_{q \in N_p} c(p, q) \quad (3)$$

where \otimes denotes the act of concatenation, following the original definition of T_C . In Eq. (3), $c(p, q)$ yields zeros values if $I(\mathbf{p}) \leq I(\mathbf{q})$, otherwise is one.

Census transformation T_C depends on how a pixel relates to its surroundings within the image patch, thus T_C is robust against individual outliers around discontinuities and noisy pixels. The direct introduction of the gradients in two image directions into the binary vector doubles the size of the produced vector T_c , thus exploiting the representational potential of image gradients. Finally, the matching cost C_c between a pixel \mathbf{p} of the reference image and its corresponding pixel \mathbf{p}' in the matching image is calculated as the Hamming distance, which represents the number of unequal elements in the two binary vectors:

$$C_{census} = \sum_{x=1}^n \left(T_c^{ref}(p) \oplus T_c^{mat}(p') \right) \quad (4)$$

Absolute Difference on Image Color: The absolute difference on color channels (ADc), or on intensity, is a simple and easily implementable measure, widely used in matching in the sense of L_1 norm. Though sensitive to radiometric differences, it has been proven as an effective measure when combined with flexible aggregation areas and referring to combination of all color layers. The cost term C_{ADc} is defined as the average absolute value over all three color channels. This turns out to improve results compared to matching on separate channels or grey-scale.

Absolute Difference on Image Principal Gradients: Here, the derivatives of image intensity in the two principal directions are extracted, and the sum of absolute differences of each derivative value in the x and y directions is used as a cost measure.

The use of directional derivatives separately, i.e., before summing them up to a single measure ADg [see Eq. (5)], introduces into the cost measure the directional information for each derivative:

$$C_{ADg}(\mathbf{p}, d) = \sum_{x,y} (\nabla I^{ref}(\mathbf{p}(x,y)) - \nabla I^{mat}(\mathbf{p}(x,y), d)) \quad (5)$$

A Gaussian filter (size 3x3, $\sigma = 0.5$) is applied on the grey-scale images before calculating partial derivatives for reducing noise and smoothing around image edges.

3.1 Support Region for Cost Function

In our approach a modification of the cross-based support region approach introduced by [18] is used. *Adaptive* approaches are based on the fact that pixels of a support region ought to have similar colors and are expected to decrease in coherence with their distance from the reference pixel in image space. The construction of the aforementioned cross-based support regions is achieved by expanding around each pixel \mathbf{p} a cross-shaped skeleton; the support region of \mathbf{p} is defined by the combination of cross skeletons belonging to pixels in the neighborhood. In [19] a linear threshold is imposed on skeleton expansion based on color similarity $\tau(\cdot)$ of neighboring pixels:

$$\tau(l_q) = -\frac{\tau_{max}}{L_{max}} \times l_q + \tau_{max} \quad (6)$$

Variables in Eq. (6) express: a) the maximum semi-dimension L_{max} of the window size, b) the maximum color dissimilarity τ_{max} between pixels \mathbf{p} and \mathbf{q} and c) l_q is the spatial distance between pixel \mathbf{q} and \mathbf{p} . The accepted difference τ between successive pixels is also restrained after [20]. Support regions generated according to the above considerations are presented in Fig. 2 for the data-set tested here.

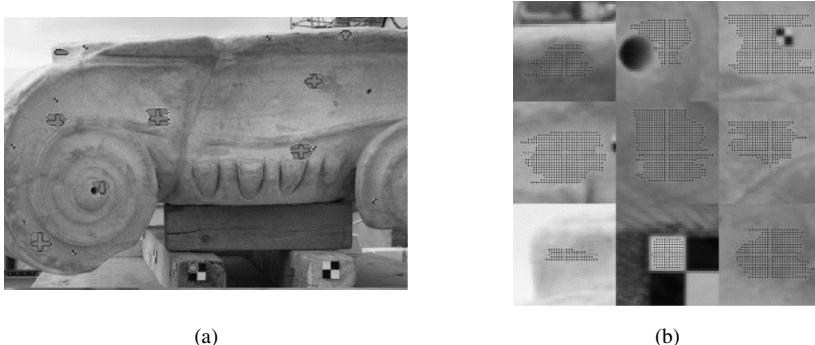


Fig. 2. Examples of the adaptive support regions formed with the linearly expanded cross-skeletons. (a) The overview of the processed image. (b) Image patches of the adaptation these windows have to image texture.

Aggregation is applied on cost using the *combined* support region W , which are the intersection of support regions of the reference pixel and its corresponding pixel on

the matching image. As a result the support region is variable according to each possible disparity value. Aggregated pixel costs C_{aggr} are normalized by the number of pixels in the support region to ensure that costs per pixel have the same scale:

$$C(\mathbf{p}, d) = \frac{C_{\text{aggr}}}{\|W(\mathbf{p}, d)\|} \quad (7)$$

Cost aggregation is implemented through *integral* images [21], in order to achieve feasible computational load and real-time performance (for low resolution images).

3.2 Geometrically Constrained Smoothing of Cost Volume

The cost values of each pixel per each potential disparity value are stored in *Disparity Space Image* (DSI) representation, thus a *cost volume* is formed. This cost volume is smoothed through a 3D filter based on Gaussian distribution and geometric constraints regarding matching. Cost filtering satisfies the need for 3D support of the aggregation step, since usual cost aggregation is defined on 2D and has the inherent limitation of assuming that all pixels in a neighborhood share the same depth (fronto-parallel assumption). Here, we propose the weighted aggregation of 2D aggregated costs $C_0(x, y, d)$ belonging to geometrically possible disparities around a pixel through the convolution of cost volume with a 3D Gaussian filter:

$$C(x, y, d) = k * C_0(x, y, d) \quad (8)$$

The Gaussian kernel k is adapted in order to serve the *ordering* and *uniqueness constraints* [22]. This kernel has the properties of attributing weights to neighboring costs inversely proportional to their spatial distance in the DSI. The advantage of this approach for 3D local support is that it avoids the need for explicit identification of slanted surfaces in 3D world space.

The estimation of disparity is carried out in the ‘winner-takes-all’ mode, as in most local and semi-global approaches, i.e., the disparity label with the lowest cost is selected. The estimated disparity map is refined through a robust post-processing procedure which includes left-right consistency check, outlier median smoothing via cross-based regions, occlusion/ mismatch labeling, sub-pixel estimation and edge-preserving smoothing on the disparity map [8, 19].

4 Experimental Results

4.1 Segmentation Results

In this section, we present segmentation results of the proposed active contour methodology. Fig. 3 presents the segmentation results. In this figure, we initially depict the original images along with all the segmentation steps as being described in Section 2.

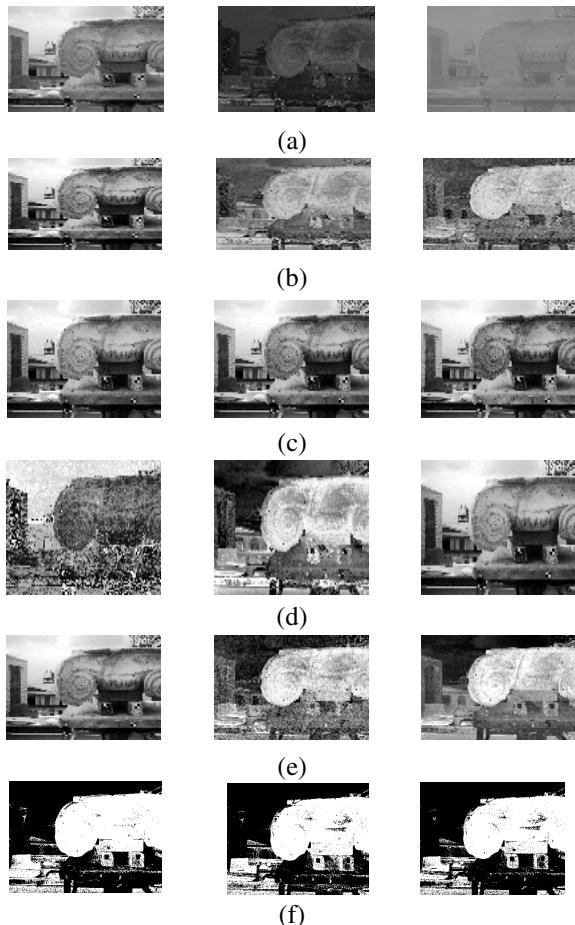


Fig. 3. (a) Original images in the RGB, HSV and LAB color space. (b) Equalized images in each color space. (c) Equalized color channels of the RGB space. (d) Equalized color channels of the HSV space. (e) Equalized color channels of the Lab space. (f) mask containing the white objects (HSV space, Lab space and fusion of them).

4.2 Stereo Matching Results

The presented algorithm has been evaluated on an object of high archaeological interest; the column capital in Fig. 4 belongs to the temple of Athena Nike on the Acropolis of Athens (<http://ysma.gr/en/athena-nike>) and it is a typical part of ancient monuments with complex architectures. Column capitals are structural parts of ancient Greek temples. Thus, as the whole temple was decomposed during restoration, the visual and geometrical documentation should be thorough to ensure that the structural and the aesthetic restoration would be complete. The images seen in the top row of Fig. 4 (camera: 12 Mp Canon EOS5; pixel size: 8.24 μm) had originally been taken for creating an orthomosaic of the capital with conventional photogrammetric techniques and are a part of the Acropolis Restoration Service photographic archive. The intrinsic and extrinsic

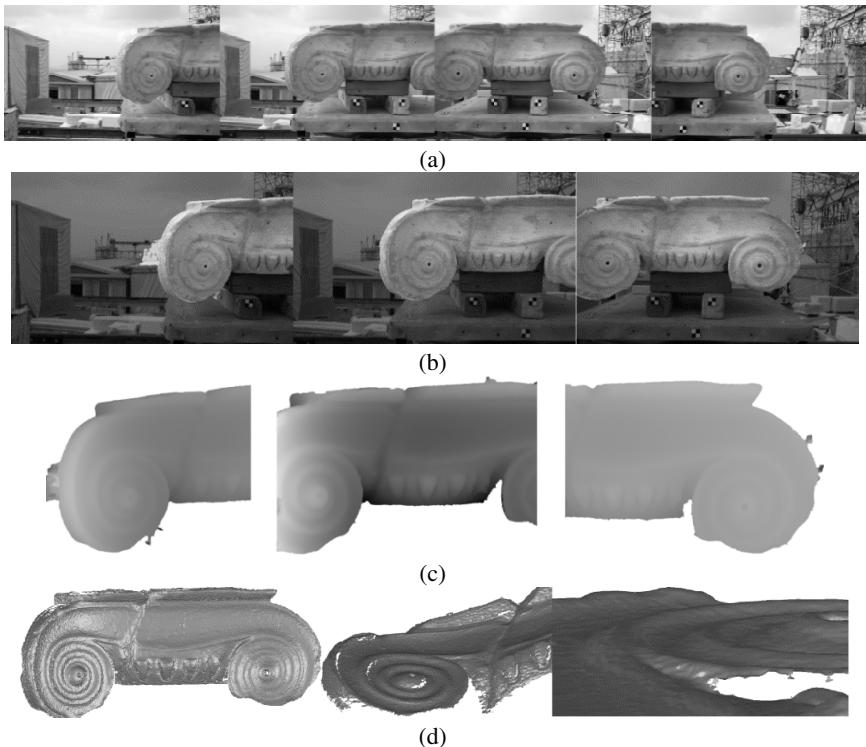


Fig. 4. A highly detailed 3D reconstruction of an important cultural heritage object has been created from multiple-base stereo-matching. (a) Original high resolution images, (b) the recognized foreground in three of the images, that present the reference for the combined stereo-pairs, (c) disparity maps corresponding to each stereo-pair used for 3D reconstruction, (d) illustration the complete face of the capital and some surface details.

orientation parameters were determined with our automatic bundle adjustment software. The control points seen in the images (Fig. 4, 1st row) have simply served scaling purposes. Three pairs were used for complete reconstruction, but matching was based on stereo. The main object of interest was detected during the foreground step and it was described through a binary “mask”. These masks define the boundaries of the object and are transformed through the epipolar geometry of each stereo-pair.

Acknowledgment. This paper is supported by the project E-Park, "Exploitation of new Technological Trends for payment and handling public parking" approved under the Interreg III Programme, Greek-Cypriot cooperation and funded from European Union and Greek National funds.

References

1. Hesmondhalgh, D.: The Cultural Industries. Sage (2002)
2. Yan Cui, S., Schuon, D., Chan, S., Thrun, T.C.: 3D shape scanning with a time-of-flight camera. In: Computer Vision and Pattern Recognition (CVPR), pp. 1173–1180 (2010)

3. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P.T., Shotton, J., Hedges, S., Freeman, D., Davison, A., Fitzgibbon, A.: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In: UIST, pp. 559–568 (2011)
4. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans. on PAMI 31(9), 1582–1599 (2009)
5. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994)
6. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(4), 401–406 (1998)
7. Hirschmüller, H.: Stereo processing by semi-global matching and mutual information. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(2), 328–341 (2008)
8. Stentoumis, C., Grammatikopoulos, L., Kalisperakis, I., Petsa, E., Karras, G.: A local adaptive approach for dense stereo matching in architectural scene reconstruction. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XL-5/W1, pp. 219–226 (2013)
9. Tomboli, F., Mattoccia, S., Di Stefano, L., Addimanda, E.: Classification and evaluation of cost aggregation methods for stereo correspondence. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
10. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(4), 650–656 (2006)
11. Bobick, A.F., Intille, S.S.: Large occlusion stereo. IJCV 33(3), 181–200 (1999)
12. Foi, A., Katkovnik, V., Egiazarian, K.: Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. IEEE Transactions on Image Processing 16(5), 1395–1411 (2007)
13. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, New Jersey (2002), Florida, R.: The Rise of the Creative Class and How It's Transforming Work, Leisure and Everyday Life. Basic Books (2002)
14. Markovic, D., Gelautz, M.: Experimental Combination of Intensity and Stereo Edges for Improved Snake Segmentation. Pattern Recogn. and Image Analysis 17(1), 131–135 (2007)
15. Malik, J., Belongie, S., Leung, T., Shi, J.: Contour and texture analysis for image segmentation. Journal on Computer Vision 43, 7–27 (2001)
16. Chan, T.F., Vese, L.A.: Active Contours Without Edges. IEEE Transactions on Image Processing 10(2) (2001)
17. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Trans. on Pattern Analysis and Machine Intelligence 28(4), 650–656 (2006)
18. Zhang, K., Lu, J., Lafruit, G.: Cross-based local stereo matching using orthogonal integral images. IEEE Trans. on CSVT 19(7), 1073–1079 (2009)
19. Stentoumis, C., Grammatikopoulos, L., Kalisperakis, I., Karras, G.: Implementing an adaptive approach for dense stereo-matching. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII(5), 309–314 (2012)
20. Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H., Zhang, X.: On building an accurate stereo matching system on graphics hardware. In: Proc. ICCV Workshop on GPU in Computer Vision Applications, pp. 467–474 (2011)
21. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on CVPR, pp. I-511–I-518 (2001)
22. Yuille, A.L., Poggio, T.: A generalized ordering constraint for stereo correspondence, MIT, AI Lab., Memo 777 (1984)

People Tracking Based on Predictions and Graph-Cuts Segmentation

Amira Soudani and Ezzeddine Zagrouba

Lab. RIADI, Équipe Systèmes intelligents en Imagerie et Vision Artificielle (SIIVA)
Institut Supérieur d'Informatique, Université de Tunis El Manar
2 Rue Abou Rayhane Bayrouni, 2080 Ariana, Tunisia
amira.soudani@gmail.com, ezzeddine.zagrouba@fsm.rnu.tn

Abstract. This paper presents a new approach to segment and track multiple persons in a video sequence via graph-cuts optimization technique. In fact, first, we extract the initial silhouettes that will be modeled by ellipses. Then, a prediction step based on optical flow vectors allows us to detect if an occlusion will handle in the next frame. Hence, we identify the occluding persons by the use of the chi-squared similarity metric based on the intensity histogram and we update the objects models of the interacting persons. Finally, a segmentation based on graph-cuts optimization is performed based on the predicted models. The experimental results show the efficiency of our algorithm to track multiple persons even under occlusion.

Keywords: Tracking, Segmentation, Occlusion, Prediction, Graph-cuts.

1 Introduction

The main goal of person tracking is to extract and follow persons in successive frames. In this aim, many challenges are encountered such as the presence of noise, the illumination variation, the scale change and occlusion. Several works are proposed to resolve the aforementioned problems. The existing tracking algorithms can be classified into three categories depending on the used features. The first category concerns the point tracking approaches in which the target object is represented by an object model that can be detected in every frame independently. Detected objects are represented by points and their association is based on the their position and motion. They include deterministic methods which associate silhouettes to tracked objects. This is done by minimizing a distance based on characteristics of the object such as proximity and appearance [1], [2]. Then, there are the probabilistic methods that deal with variations (noise, movement, appearance) [3], [4], [5]. Other methods are based on the minimization of energy functions which take into account the topological changes [6], [7]. In [3], the authors proposed a detection-based approach to multi-object tracking. They used a CRF model while considering tracking as a labeling process. In [4], a multiple objects tracking algorithm based on the flow linear programming is proposed. The authors formalize the motion of targets as flows along the

edges of a graph of spatio-temporal locations and they resolve the association problem with linear programming optimization. Anton et al.[5] proposed an approach that addresses data association and trajectory by minimizing a consistent discrete-continuous energy.

The second category is the kernel tracking methods which are based on the tracking of a predefined shape associated to the tracked object. They are based on the conservation of the color appearance [8], [9] or luminance of the object for at least two consecutive frames [10], [11].

The third category is the silhouette tracking methods which apply dynamic segmentation without prior knowledge about the shape of the objects. They are based on successive segmentations. Those approaches are based on the state models [12], [13], [14].

In this paper, we address the problem of the tracking of multiple persons with occlusion handling. The motivation of our work is to track people by the segmentation of the successive frames based on the prediction of their location with occlusion handling. First, we extract initial targets that will be modeled by ellipses. Then, we process iteratively the successive frames. In fact, we predict the new locations of the silhouettes and we detect if an occlusion will occur in order to update those associated to the interacted persons. A segmentation based graph-cuts is finally performed providing us the silhouettes of the tracked persons.

The reminder of the paper is organized as follows. Section 2 outlines the proposed approach for the tracking process. In section 3, we present the experimental results followed by the conclusion and the future research directions in section 4.

2 Proposed Tracking Process

The fig.1 presents a block diagram of the proposed approach that will be detailed in the following paragraphs.

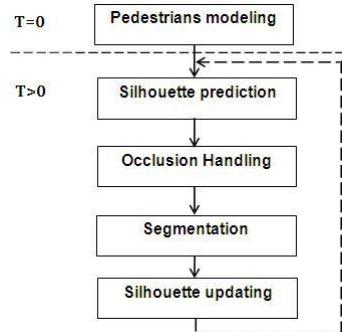


Fig. 1. Block diagram

2.1 Appearance Model

The extracted silhouette at each frame is represented by a model. In this work, we choose ellipses to model the extracted silhouettes. This choice is due to the nature of tracked targets which are persons. As shown on fig.2(a), each silhouette $Silh_i^t$ is modeled by an ellipse denoted $\xi_i^t(o, a, b, \theta)$ with center o , main axis of half-length a and b , and orientation θ . We assume that when a person move, the topological variations of its shape occur especially on the border. Thereby, we extract two ellipses, the first encompass $Silh_i^t$ by applying dilation on the initial ellipse, while the second is circumscribed inside it by applying an erosion. This process provides us respectively $\xi_{i,d}^t(o, a_d, b_d, \theta)$ and $\xi_{i,e}^t(o, a_e, b_e, \theta)$.

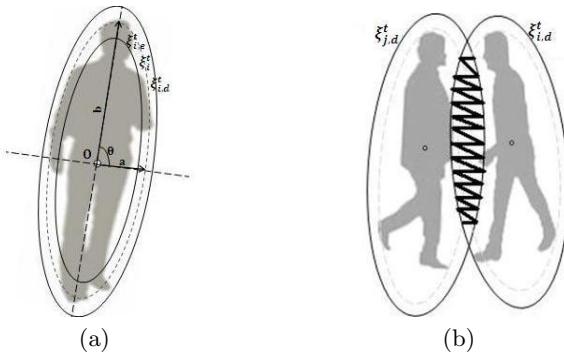


Fig. 2. (a)Appearance model,(b)Occlusion detection

2.2 People Detection

First, we perform a simple background subtraction algorithm based on frame difference followed by the application of a median filter in order to remove well the impulse noise and to preserve edges. This provides us a set of silhouettes. Then, we apply erosion on the extracted silhouettes followed by a thresholding step to remove small regions from the tracking process. Hence, we obtain a set of N silhouettes at time $t=0$ denoted $Silh_i^0$ with $i \in \{1 \dots N\}$ associated to the people in the first frame. Those extracted silhouettes will be modeled and tracked throughout the video sequence. We predict the location of each silhouette at the following frame by a forward projection. In fact, we use each ellipse ξ_i^t associated to the silhouette $Silh_i^t$ for the estimation of a predicted one denoted ξ_i^{t+1} . We denote by X_i^t the set of pixels belonging to ξ_i^t . We translate X_i^t by adding the average $\bar{v}_i^{t,t+1}$ of the optical flow vectors $v_j^{t,t+1}$ of pixels j belonging to the person i . If a person is totally occluded, the average of its optical flow vectors is null. In this case, we use the last non null average of optical flow vectors to predict ξ_i^{t+1} .

$$\xi_i^{t+1} = \{X_i^{t+1} = X_i^t + \bar{v}_i^{t,t+1} / X_i^t \in \xi_i^t\} \quad (1)$$

2.3 Occlusion Handling

Once we predict the set of silhouettes, we detect if there is a common area belonging to each pair of them (fig.2(b)). If it is the case, we can assume that an occlusion will handle on the associated frame. We denote this area by Δ .

$$\Delta(i, j) = \xi_{i,d}^{t+1} \cap \xi_{j,d}^{t+1} \quad (2)$$

where $(i, j) \in \{1 \dots N\}^2 \setminus \text{diag}$

Updating Interacting Models. We assign Δ to the occluding person by minimizing a distance between it and the N interacting persons. We define a distance function D which evaluates the similarity of two distributions of color histograms based on the Khi squared measure χ^2 . At each frame, we compute the color histogram associated to each person i based on $\xi_{i,d}^{t+1}$. The person that minimizes this distance is defined as the occluding person and Δ is assigned to it. Then we update the predicted silhouettes associated to those interacting persons.

$$D(\Delta, \xi_{i,d}^{t+1}) = \chi^2(h_\Delta, h_i), i \in \{1 \dots N\} \quad (3)$$

$$\chi^2(h_\Delta, h_i) = \sum_{j=1}^n \frac{(h_\Delta(j) - h_i(j))^2}{h_\Delta(j) + h_i(j)} \quad (4)$$

where n is the number of bins and h_i and h_Δ are the color histograms associated respectively to $\xi_{i,d}^{t+1}$ and Δ .

Departure and Entrance Events Detection. When the location of the predicted silhouettes seems to be outside the image plane, we assume that the corresponding persons will exit form the scene, so we take into consideration their departure. In the case of the entrance of new people, we can detect that by applying a subtraction algorithm on the border of the frame.

2.4 Segmentation with Graph-Cuts

In order to extract the silhouettes corresponding to tracked people, we define an energy function for each person i at the frame $t+1$ denoted by E_i^{t+1} . This energy is minimized by the min-cut/ max-flow algorithm [15]. The model prediction step allows us to compute a color distribution for each person that is used in the definition of this function.

Energy Function. First, we define the energy function for a labeling L that will be computed for each person i at time $t + 1$ based on the predicted $Silh_i^{t+1}$. This energy function $E_i^{t+1}(L)$ is composed of two terms as shown on equation 5.

$$E_i^{t+1}(L) = \sum_{p \in I} U_{i,p}^{t+1}(l_p) + \sum_{p \neq q, \{p,q\} \in Q} B_{i,(p,q)}^{t+1}(l_p, l_q). \quad (5)$$

where for a pixel k , the label l_k can be *Object* or *Background* while Q is the set of unordered pairs (p, q) of neighboring pixels belonging to the frame I . The first term of the energy function is a unary term that computes the cost of the assignment of a pixel to a label while the second term is a binary term which evaluates the assignment of neighboring pixels to different labels.

Unary Term. This term evaluates the cost of the assignment of a pixel to a label. We compute a color distribution for the background and the foreground relying on the predicted silhouettes. For each person i , we determine the color distribution which is a gaussian mixture model adjusted to the set of pixels belonging to $\xi_{i,e}^{t+1}$ at the frame $t + 1$ while the background color distribution is based on the pixels outside all the predicted ellipses $\xi_{i,d}^{t+1}$.

$$U_{i,p}^{t+1}(l_p) = -\log P(I_p | l_p). \quad (6)$$

where $P(I_p | l_p)$ denotes the probability at pixel p to belong to a label l_p and I_p is the grey level associated to the pixel p . For a label l_p , the global component of color model is represented by a Gaussian Mixture Model (GMM):

$$P(I_p | l_p) = \sum_{k=1}^K \omega_{ik} G(I_p | \mu_{ik}, \Sigma_{ik}). \quad (7)$$

where G is the Normal distribution and ω_{ik} , μ_{ik} and Σ_{ik} represent respectively the weight, the means and covariance matrix of the k^{th} component for label l_p , K is number of the components associated to the mixture model of label l_i .

Binary Term. This term aims to penalize labeling discontinuity of neighboring pixels even if they have the same intensity. It is based on the color gradient and it aims to smooth the segmentation by penalizing the assignment of pairs of neighboring pixels to different labels.

$$B_{i,(p,q)}^{t+1}(l_p, l_q) = \frac{1}{dist(p, q)} \exp\left(-\frac{\|I_p - I_q\|^2}{2\sigma^2}\right). \quad (8)$$

where $dist$ is the standard L_2 Euclidean norm. It evaluates the distance between neighboring pixels p and q while $\sigma^2 = \langle \|I_p - I_q\|^2 / \|p - q\|^2 \rangle$ is the average contrast over all $(p, q) \in Q$.

Energy Minimization. In order to obtain a final labeling of pixels L_i^{t+1} , we minimize the energy function defined below.

$$L_i^{t+1} = \operatorname{argmin} E_i^{t+1}(L). \quad (9)$$

This energy is modeled by a graph $G = \{E, N\}$ and is composed by a set of edges E and nodes N . Each pixel is considered as a graph node in addition to two nodes for the labels Background or Object. The Unary term $U_{i,p}^{t+1}$ is implemented by connecting each pixel p to the labels nodes. In fact, the edge weights represent the penalty of assigning each pixel p to a label l_p . The binary term is computed by connecting all combination of pairs of neighboring pixels (p, q) with an edge weight representing the cost of the assignment of those pixels to different labels. The min-cut of this weighted graph provides us the better segmentation of the N persons at time $t + 1$. Successive segmentations obtained by graph cuts that include the prediction step allow us to track each detected silhouette in the video sequence.

2.5 Updating Silhouettes

Once the energy function is minimized, we obtain a set of silhouettes associated to the tracked persons. In order to reduce the propagation error between successive frames, we update the predicted silhouettes provided by the prediction step.

$$Silh_i^{t+1} = L_i^{t+1}. \quad (10)$$

Finally, we present the outlines of the proposed method in Algorithm1.

Algorithm 1. Overview of the proposed method

People detection and modeling:

$$Silh_i^t, i = \{1..N\}, \xi_i^t, \xi_{i,d}^t, \xi_{i,e}^t$$

for each frame $t > 0$ do

1. Silhouette prediction: $\xi_i^{t+1}, \xi_{i,d}^{t+1}, \xi_{i,e}^{t+1}$

2. Occlusion Handling:

$$\Delta(i, j) = \xi_{i,d}^{t+1} \cap \xi_{j,d}^{t+1}, i \neq j, (i, j) \in \{1..N\}^2$$

3. Segmentation based Graph-cuts

(a) Graph construction and estimation of edges costs: E_i^{t+1}

(b) Energy minimization:

$$L_i^{t+1} = \operatorname{argmin} E_i^{t+1}(L)$$

4. Updating silhouettes:

$$Silh_i^{t+1} = L_i^{t+1}$$

end for

3 Experiments

The main goal behind the proposed algorithm is to track multiple persons in video sequences related to surveillance area. In fact, there are surveillance datasets including a large corpus of sequences recorded for more than a decade and used in various works to evaluate the related algorithms [4], [5].

3.1 Datasets

To evaluate our method, we use the following datasets.

- PETS 2006 dataset [16]: It is composed of five sets of frames captured from different views recorded in a train station. Persons on these sequences have different motions and can be under occlusion condition.
- PETS 2009 dataset [16]: This dataset consists of five separate sets of training and test sequences. Each set consists of one training sequence and one test sequence. All the datasets are multi-view and include challenges in terms of significant lighting variation, occlusion, scene activity and use of multi-view data.

3.2 Quantitative Evaluation

In order to prove the efficiency of the proposed algorithm, we propose an evaluation protocol. The reliability of a such protocol depends on three items. First, the adoption of evaluation scores that estimate the quality of tracking results. Then, the choice of appropriate data sets under various conditions which guarantee good results on real-world scenarios. Finally the use of a ground truth including desired tracking results.

Ground Truth. The ground truth at pixel level is often approximated with a bounding box or ellipse around the target area. In multi-target tracking, the evaluation can be cast onto multiple single-target evaluations using the scores defined below. The ground truth used to evaluate our algorithm is provided with the datasets. It provides us the bounding boxes and the centroid coordinates of the tracked persons on each frame.

Metrics. We use a selection of the metrics proposed in the Video Analysis and Content Extraction (VACE) protocol [17]. In order to evaluate the detection tasks of the proposed algorithm, we will use the MODA (Multiple Object Detection Accuracy) which evaluates the accuracy aspect of the system performance and the MODP (Multiple Object Detection Precision) that expresses the average overlap over the matching data. Otherwise, in order to evaluate the overall tracking performance, we use the MOTP (Multiple Object Tracking Precision) which evaluates the overall tracking precision and the MOTA (Multiple Object Tracking Accuracy) that evaluates the accuracy aspect of the system. This last metric is based on the false positive, the false negative and the number of switches in the system output for a given reference ground truth.

Statistical Results. In our experiments, the parameters are tuned as follow, we used a 8-neighborhood system. In fact the choice of neighborhood affects the smoothness of the segmentation, so smaller neighborhood introduces irregular segmentation. The number of components of the GMM is set to 5 and the experimentation are done on a standard workstation. We compare the tracking results of our proposed method with the works of [5] and the two methods proposed by [4] : the Sequential Dynamic Programming (SDP) and Linear Programming (LP). In table1, we evaluate the overall accuracy (MOTA) and precision (MOTP) of the tracking algorithm. We deduce that our algorithm provides an accuracy factor of 0.67 relatively less than those provided by [5] and LP [4] but greater than SDP [4]. For the precision factor, the proposed method provides a MOTP of 0,61 which is better than LP [4], SDP [4] and [5]. We also evaluate the accuracy factor of the detected silhouettes. The MODA term is based on the false positive and the false negative rates while the MODP term expresses the precision of the location associated to detected silhouettes in each frame. We deduce that the proposed approach provides results better than SDP [4] and close to [5] and LP [4].

Table 1. Evaluation

	LP[4]	SDP[4]	[5]	Proposed method
MOTA	0.82	0.11	0.89	0.67
MOTP	0.56	0.1	0.562	0.61
MODA	0.85	0.11	0.908	0.7
MODP	0.57	0.12	0.573	0.57
Frequence (fps)	-	-	0.5	1.32

Complexity of the Algorithm. The computational speed of the proposed method is related to the number of tracked targets. Our approach is implemented using Matlab and C++ Mex files. An evaluation of the computational time is shown in Table1. The average speed is up to 1.32 fps which is reduced comparing to [5].

3.3 Qualitative Evaluation

We show samples of the tracking results of the proposed method on PETS 2009 dataset (View 001) in fig.3. In fact, we notice that pedestrians are well detected and tracked. In fig.3(b-c), two persons who leaved the scene have been detected while in fig.3(d) and fig.3(i), our algorithm detected the entrance of new pedestrians in the sequence. On the another hand, the algorithm succeeded to track persons under occlusions (fig.3(h-k)).

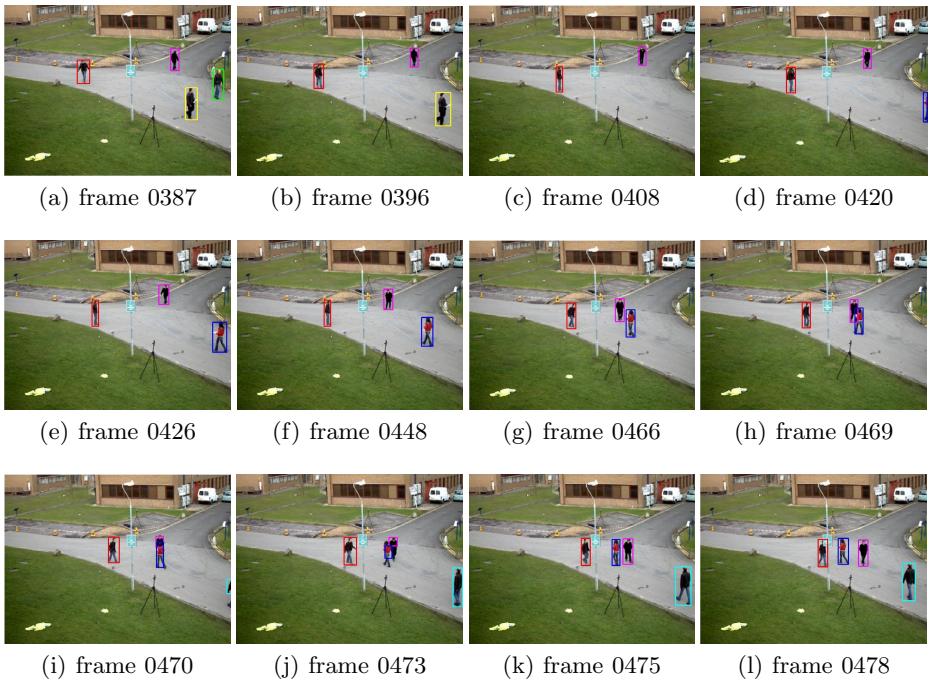


Fig. 3. Tracking results on PETS 2009 dataset (View1)

4 Conclusion

In this paper, we presented a multi-persons tracking approach based on segmentation with graph-cuts optimization. First, we proceed to a background subtraction algorithm which provides us initial silhouettes. Each one corresponds to a target that will be tracked along the sequence. Then, each silhouette is modeled by an ellipse and we predict the new location of the targets at the successive frame. In this step, we detect eventual occlusion and we update the models of interacting persons. Later, we proceed to a segmentation based on the minimization of an energy function via graph-cuts optimization technique. Experiments are performed on several video sequences. In fact, our method provides good tracking results with a reduced computational time. As future directions, we aim to deal with the case of initial detection including group of people. Also, we will try to apply our method on multi-view sequences and to improve the tracking process by join tracking to reconstruction.

References

1. Sato, K., Aggarwal, J.K.: Temporal spatio-velocity transform and its application to tracking and interaction. Computer Vision and Image Understanding 96(2), 100–128 (2004)

2. Shafique, K., Shah, M.: A non-iterative greedy algorithm for multi-frame point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 51–65 (2003)
3. Heili, A., Odobezi, J.-M.: Parameter estimation and contextual adaptation for a multi-object tracking crf model. In: *IEEE Workshop on Performance Evaluation of Tracking and Surveillance* (2013)
4. Berclaz, J., Fleuret, F., Fua, P.: Multiple object tracking using flow linear programming. In: *The 12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pp. 1–8 (2009), <http://fleuret.org/papers/berclaz-et-al-pets2009.pdf>
5. Andriyenko, A., Schindler, K., Roth, S.: Discrete-continuous optimization for multi-target tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1926–1933 (2012)
6. Criminisi, A., Cross, G., Blake, A., Kolmogorov, V.: Bilayer segmentation of live video. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 53–60 (2006)
7. Yilmaz, A., Li, X., Shah, M.: Contour based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1531–1536 (2004)
8. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 564–577 (2003)
9. Boltz, S., Debreuve, E., Barlaud, M.: High-dimensional statistical distance for region-of-interest tracking: Application to combining a soft geometric constraint with radiometry. In: *IEEE Conference on Computer Vision and Pattern Recognition*, p. 7 (2007)
10. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25, 1296–1311 (2003)
11. Nguyen, H.T., Smeulders, A.W.M.: Fast occluded object tracking by a robust appearance filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1099–1104 (2004)
12. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision* 29, 5–28 (1998)
13. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects, 572–578 (1999)
14. Amri, S., Barhoumi, W., Zagrouba, E.: A robust framework for joint background/foreground segmentation of complex video scenes filmed with freely moving camera. *Multimedia Tools and Applications* 46(2), 175–205 (2010)
15. Boykov, Y., Jolly, M.-P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In: *International Conference on Computer Vision*, pp. 105–112 (2001)
16. PETs, Performance evaluation of tracking and surveillance, <http://www.cvg.rdg.ac.uk/slides/pets.html>
17. Kasturi, R., et al.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(2), 319–336 (2009)

A Framework for Quick and Accurate Access of Interesting Visual Events in Surveillance Videos

Fei Yuan¹, Chu Tang¹, Shu Tian², and Hongwei Hao¹

¹ Institute of Automation, Chinese Academy of Sciences, Beijing, China 100190
`{fei.yuan,chu.tang,hongwei.hao}@ia.ac.cn`

² Department of Computer Science and Technology, University of Science
and Technology Beijing, China 100083
`tshu23@gmail.com`

Abstract. It is a challenge to quickly and accurately access interesting visual events in explosively increasing surveillance video data. In this paper, we develop a novel video summarization framework to accurately detect and quickly browse interesting visual events in large-scale surveillance videos. The method firstly detects interesting foreground objects by running a set of pre-trained object classifiers on foreground object candidates which are generated by a background subtraction algorithm. Then a Hungarian algorithm based multi-objects tracking program is followed to obtain accurate and complete motion trajectories of detected foreground objects. Finally, each interesting visual event is compactly represented by a synthetical snapshot, which makes it convenient to quickly access interesting visual events in long videos. Experiments on challenging surveillance videos show our framework outperforms existing video summarization systems in the detection accuracy of interesting visual events.

1 Introduction

Modern society is increasingly dependent on surveillance cameras. Everyday, widely distributed surveillance cameras generate an enormous amount of surveillance video data. It presents severe challenges to efficiently utilize these data. Among numerous applications, an important demand is to quickly and accurately access interesting visual events in massive video data.

Recently, video summarization techniques are developed to shorten long videos into short ones by merely preserving interesting visual contents and rearranging them in a compact way. To automatically extract interesting visual contents, the simplest way is to represent original videos by collecting key frames [1–4]. For example, in [2], Kim and Hwang proposed a key-frame based video abstraction that transforms an entire video clip to a small number of representative images. Even though key frames based methods can often generate compact video representation, they often face the risk of losing important information, such as the dynamic aspects of video contents. Moreover, it is difficult to extract proper key frames.

An alternative method is called video synopsis [5–9] or video condensation [10]. A classic framework of video synopsis consists in firstly extracting dynamic objects, such as moving pedestrians and vehicles, then rearranging them in a compact video by minimizing an energy function that optimally compresses the play time of such objects [6, 10]. The seminal work is proposed by Peleg and his colleagues [6]. In [6], dynamic objects are extracted by segmenting moving foreground objects. Specifically, background subtraction combined together with min-cut segmentation is exploited. Then the synopsis video is generated with a mapping that assigns each coordinate in the video synopsis the coordinates of a source pixel from the input video. The mapping is obtained by minimized a cost function which consists in a loss term in activity and a discontinuity cost. This method can achieve a very large compression ratio and enable viewers to browse a long video in several minutes.

While current video synopsis techniques do well in shortening long videos into compact representations, it is still an open problem to accurately detect and track interesting visual events in realistic situations. Surveillance videos involved in security applications often contain ambiguous objects, especially dim pedestrian objects and dense objects with mutual occlusion. Segmentation-based foreground object detection and tracking algorithms [6, 9, 10] often fail to deal with it. The detection precision of dim pedestrian objects with size smaller than 15×30 is extremely low, while dense objects with mutual occlusion are usually detected as a single object by these algorithms. Poor abilities of object/events detection and tracking make current video synopsis techniques impractical in realistic applications.

In this paper, we propose a novel video summarization framework to accurately detect and quickly browse interesting visual events in large-scale surveillance videos. Our method can accurately detect interesting visual events by combining state-of-the-art object detection techniques and multi-objects tracking techniques. The contributions of this paper are three-fold: Firstly, the method detects interesting foreground objects by running a set of pre-trained object classifiers on the foreground which is generated by a background subtraction algorithm. Our classifiers are well designed in order to successfully detect ambiguous objects, especially dim pedestrian objects and dense objects with mutual occlusion. Secondly, a Hungarian algorithm based multi-objects tracking program is followed to obtain accurate and complete motion trajectories, namely interesting visual events. Finally, each interesting visual event is compactly represented by a synthetical snapshot, which makes it convenient to quickly access interesting visual events in long videos. Furthermore, we experimentally demonstrate that our framework outperforms existing video summarization systems in the detection accuracy of interesting visual events on challenging surveillance videos.

2 Our Approach

In this work, we define an interesting visual event as the temporal evolution of a moving objects (noted as motion trajectory) and aim to accurately extract its

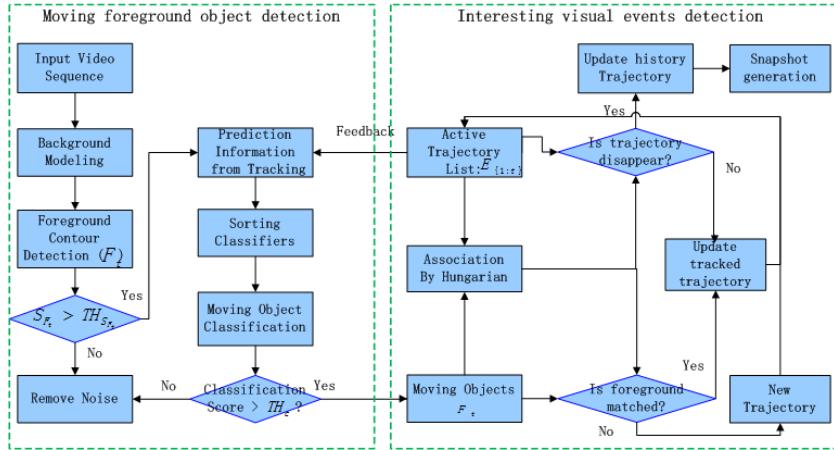


Fig. 1. A graphical illustration of our video summarization framework. Our framework mainly consists in two modules: a detection module and a tracking module. These two modules are mutually improved. The feedback information from tracking module contributes to improve the detection precision and accelerate detection speed.

complete lifecycle. We propose to extract interesting visual events in a tracking-by-detection framework, where detection and tracking are mutually improved. Figure 1 gives an overview of our method. In the first stage, we detects interesting foreground objects by running a set of pre-trained object classifiers on foreground object candidates which are generated by a background subtraction algorithm. In the second stage, we establish the temporal correspondence of moving objects by using a Hungarian algorithm based multi-objects tracking program. It is worth to note that we introduce a feedback between the tracking module and detection module, where tracking information is feeded back to improve detection. Finally, in order to quickly browse interesting visual events in long videos, each interesting visual event is compactly represented by a snapshot, which contains rich information of an event, such as motion trajectory and appearance of object.

2.1 Moving Object Detection

In this section, a set of pre-trained object classifiers are applied on the moving foreground produced by a background subtraction algorithm, in order to detect various moving objects of interest. The background subtraction, which is used as a preliminary procedure in the moving object detection stage, can remove the still background and generate foregrounds containing possible moving object candidates for each frame of a source video. The detection is then conducted on foregrounds using a set of classifiers with a sliding window approach.

This mechanism of detection avoids searching the whole frame with the object classifier which would demand a large amount of computation, since the

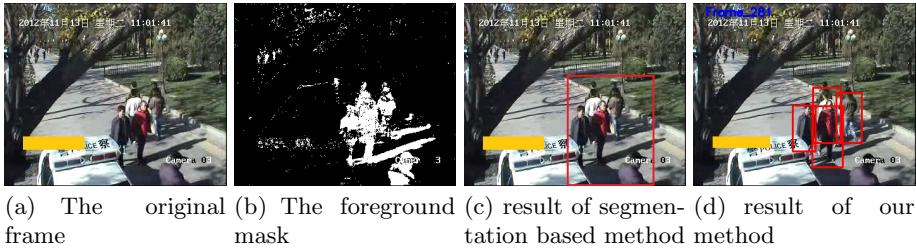


Fig. 2. Example visualization of ViBe background subtraction and detection results of segmentation based method and classification based method

background subtraction is able to remove pixels that are most unlikely to be moving before applying the object classifier. In this work, ViBe [11], a recently reported approach is employed in our background subtraction process. ViBe has shown its superiority in its modeling efficiency and its accuracy in segmenting the foreground compared with most background subtraction techniques.

Figure 2 gives an example of the applied ViBe for background subtraction. Figure 2(a) and Figure 2(b) show a frame chosen from a video surveillance sequence and its corresponding binary foreground mask produced by ViBe background subtraction approach, respectively. It is clear that in Figure 2(b), several mutual occluded pedestrians together make a large foreground region. Segmentation based method fails to extract particular pedestrians, see Figure 2(c). Therefore, well designed classifiers are necessary in order to resolve such a situation and Figure 2(d) illustrates the detection result produced by applying a pre-trained pedestrian classifier on the corresponding foreground.

In this work, we choose to utilize the classification based detection approach proposed by Dalal and Triggs [12], in which histograms of oriented gradients (HOG) feature is extracted as the representation of moving objects and support vector machine (SVM) is employed as the classifier. The effectiveness of this approach in detecting pedestrians and cars has been shown in applications. In our work, the pre-trained classifiers are applied with a sliding-window approach only on the moving object candidate regions of a frame, i.e. the regions indicated by the foreground mask generated by ViBe, using various scales, which are determined by the size of the foreground region as well as the feedback from the tracking module. Note the foreground may need some simple morphological operations for the de-noising purpose before conducting the classification in order to avoid running the detectors on small regions that are falsely segmented as foreground by ViBe.

Besides the overall mechanism of background subtraction and classifier described above, some strategies are also applied in this work to achieve accurate detections in different scenes of surveillance:

Intra Class Variation. It is known that in surveillance videos, the difference between objects not only exists in mutual classes, such as car and pedestrian, but also in a single intra class. For example, a sedan and a minivan both belong

to the car class, while they have different shapes and appearances; nevertheless, even the perspective and the viewing angle could make a same object quite different with regard to the HOG feature. Hence, a set of classifiers for different classes, various scales as well as viewing angle needs to be trained so as to widely adapt to different surveillance scenes and achieve good performances in them. In this work, for each of the three classes: car, bicycle (including motor cycle) and pedestrian, two classifiers of both front view and side view, are pre-trained, respectively. For a new object candidate appearing in the scene, each classifier is applied on the corresponding region of the frame to detect the potential object, and the one with the highest detection score among all the detection results is taken as a detection of this region, also the corresponding class of this detected object is recorded. The detection result together with its class label would be used as the input of the tracking module. Such an object is tracked and its predicted location in next frame as well as the class label is utilized for the detection of next frame. The use of such feedback information will be talked in the next paragraph.

Feedback from Tracking. For the purpose of efficiency, the predicted positions of different objects as well as their class labels from the tracking process could be well utilized. As described previously, for the detection of an object which has already been under tracking, a prior knowledge of its possible coordinates and the class label are both available in the current frame. Therefore, combined with the segmented foreground region by ViBe, the predicted location and the class label are able to help avoid conducting all kinds of classifiers repetitively on the same region for detection.

Sort of Classifiers. Since a set of different classifiers have been trained for the detection, a judicious sort of them is quite necessary so as to quickly detect the expected object instead of employing all the classifiers every time which would lead to rapid growth of computation amount concomitantly. For a specified region of the foreground, classifiers are ranked according to the class label of this region, which is produced by the tracking module, as discussed previously. For example, if the class label indicates a specified region to be a person, then the classifiers trained for pedestrian are ranked on the top, followed by other kind of classifiers. In the detection, classifiers are applied for detection sequentially according to the ranking until the object is detected. Nevertheless, a parameter that controls the detection scale is attached to the corresponding classifier at the same time. This parameter is determined by the classification result of last frame together with the prediction information for the corresponding foreground region.

Classification Score. If the detection score of a classifier is greater than a threshold, a detection of the corresponding class is defined. In order to determine the threshold for a classifier, the classifier is applied on each of the positive samples used during its training process, and the lowest positive score is set as its threshold.

2.2 Detection of Interesting Visual Events

In this section, detection responses of foreground objects across video frames are partitioned into trajectories. We regard each associated trajectory as an interesting visual event. Instead of tracking each single object in an image sequences by a recursive Bayesian filter [10], we formulate the problem in a multi-object tracking framework where observations are associated across video frames. Hungarian method is applied to compute the assignments between the object hypotheses across video frames based on the affinity of the responses in position and appearance.

Consider $E_{1:t} = \{e_{1:t}^1, \dots, e_{1:t}^m\}$ as all the active object trajectories up to frame t , and $F_t = \{f_t^1, \dots, f_t^n\}$ as all the detection responses at frame t . The pairwise association likelihoods are computed based on a color model and a dynamic model. The color model is an object-specific color histogram. Specifically, for each observation f_t^i , we compute an $8 \times 8 \times 8$ histogram in RGB color space over the foreground object area. The Hellinger distance, which is related to Bhattacharyya coefficient, is used to compare color models,

$$d(\mathbf{h}_1, \mathbf{h}_2) = \sqrt{1 - \frac{1}{\sqrt{\mathbf{h}_1 \mathbf{h}_2 N^2}} \sum_{q=1}^N \sqrt{\mathbf{h}_1(q) \mathbf{h}_2(q)}} \quad (1)$$

where, \mathbf{h}_1 and \mathbf{h}_2 are two color histograms, N is the number of bins(here is 512), and $\bar{\mathbf{h}}_k = \frac{1}{N} \sum_{j=1}^N \mathbf{h}_k(j)$.

As for the dynamic model, for each active object trajectory $e_{1:t}^k$, we first compute its predicted position (x'_{t+1}^k, y'_{t+1}^k) which is approximated by a constant velocity model: $x'_{t+1}^k = x_t^k + u_t^k, y'_{t+1}^k = y_t^k + v_t^k$, where, u_t^k and v_t^k are estimated velocity of $e_{1:t}^k$ at frame t . The metric to compare dynamic model is Euclidean distance. The uncertainty of color model and dynamic model are both modeled as Gaussian distribution,

$$\begin{aligned} p(\mathbf{h}_{t+1}^i | \mathbf{h}_t^i) &\sim N(\mathbf{h}_t^i, \Sigma_c) \\ p(\mathbf{x}_{t+1}^i | \mathbf{x}_t^i) &\sim N(\mathbf{x}_{t+1}^i, \Sigma_l) \end{aligned} \quad (2)$$

where, \mathbf{h}_t^i is the color histogram of i_{th} object at t_{th} frame, \mathbf{x}_t^i is the location of i_{th} object at t_{th} frame, \mathbf{x}_{t+1}^i is the prediction location of i_{th} object at $(t+1)_{th}$ frame, and Σ_c and Σ_l are the covariance matrices for color histogram and location respectively.

Then the similarity between a detection response and an active object trajectory is evaluated yielding:

$$simi_{i,j}^t = p(\mathbf{h}_{t+1}^i | \mathbf{H}_t^j) p(\mathbf{x}_{t+1}^i | \mathbf{X}_t^j) \quad (3)$$

where, $simi_{i,j}^t$ is the similarity between the i_{th} detection response at $(t+1)_{th}$ frame and the j_{th} active object trajectory at t_{th} frame, \mathbf{h}_{t+1}^i is the color histogram of the i_{th} detection response at $(t+1)_{th}$ frame, \mathbf{H}_t^j is the color histogram of the j_{th} active object trajectory at t_{th} frame, \mathbf{x}_{t+1}^i is the location of the i_{th} detection response at $(t+1)_{th}$ frame, and \mathbf{X}_t^j is the location of the j_{th} active object trajectory at t_{th} frame.

Figure 1 shows the framework of our multi-object tracking algorithm. Consider F_t as the foreground objects set extracted by classification based detector in frame t , and $E_{1:t-1}$ as active trajectory list up to frame $t-1$. Similarities between F_t and $E_{1:t-1}$ are calculated according to formula 3. Then to match the foregrounds and tracked trajectories, we maximize the similarities via Hungarian algorithm. Finally, the matched pairs generated by Hungarian algorithm are further checked. If the similarity of a pair (an active trajectory and a foreground candidate) is smaller than a threshold, they are removed from the matched pairs. This step is useful to avoid the mismatching derived from the greedy matching.

We maintain two trajectory lists: active trajectory list and history trajectory list. The active trajectory list includes all active trajectories, while history trajectory list includes dead trajectories. Foreground objects which are not matched to any active trajectories are treated as a new trajectory candidate. If it can be constantly tracked for T_{new} frames, this trajectory candidate is added to the active trajectory list. Otherwise, it is considered as noisy. When a trajectory is dead, we add it into the history trajectory list. In order to improve the robustness of our tracking system, only if a trajectory is not matched to any foreground objects for T_{lost} frames, it is sentenced to be dead. In our work, we empirically set $T_{new} = 3$ and $T_{lost} = 3$.

2.3 Quick Browse of Interesting Visual Events

The trajectories detected in above sections reflect most, if not all, of interesting visual contents in long videos. To quickly browse long videos, we only show these trajectories and discard the remaining redundant contents. Similar to [2, 4], We compact a long video by a set of pictures. However, instead of using key frames as essential building blocks [2, 4], we exploit a synthetical snapshot to represent each interesting visual event. Compared with key frames, our synthetical snapshots can record rich information of interesting visual events: not only appearance of objects but also the dynamic property of events, such as the motion trajectories, the starting/ending time and location of events.

Figure 3 shows two examples of synthetical snapshots extracted in realistic surveillance video data. The red rectangular box indicates the object involved in an interesting visual event. Cross-over points show the motion trajectory of the object, where points with blue color represent the beginning locations of the event and points with red color represent the ending. Furthermore, the beginning/ending times of the life cycle of the event are also displayed.



Fig. 3. Example visualization of video snapshots extracted in realistic surveillance videos. Each interesting visual event is displayed by a snapshot which records the involved object as well as its motion trajectory.

3 Experimental Results

We evaluate the performance of the proposed method on dozens of realistic surveillance videos. Figure 4 and Figure 5 show two typical video clips collected at traffic intersection. These videos often contain ambiguous objects, such as dim pedestrians and dense pedestrians with mutual occlusion.

We compare our method with state-of-the-art tracking-by-segmentation event detection methods [6, 9]. We implemented a video summarization system based on tracking-by-segmentation method as a baseline, where min-cut segmentation is performed on foregrounds and recursive Bayesian filters are used independently to track objects. To equally compare performances of different methods, we use the same experimental details, including background modeling, features on tracking and so on.

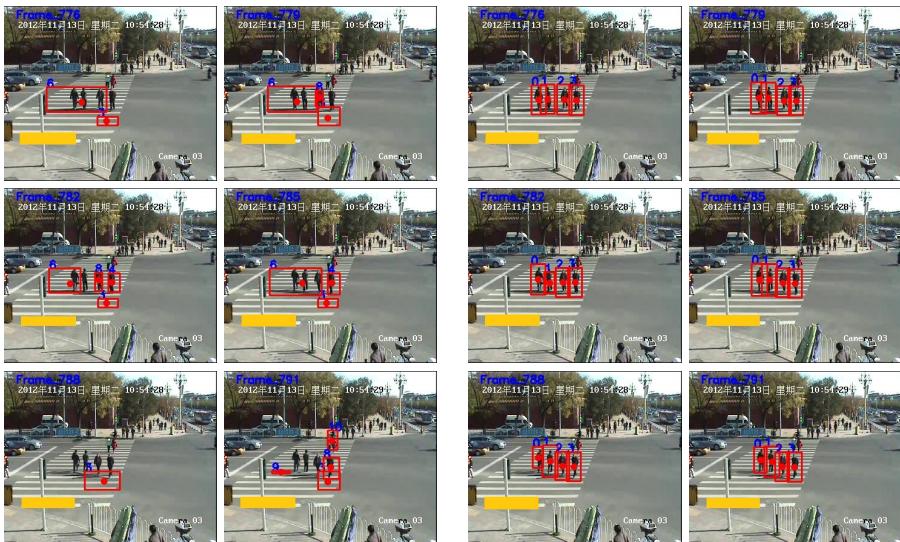
Figure 4 shows results of event detection and tracking generated by different methods in a situation with dense pedestrians with mutual occlusion. From Figure 4, we can see that our method can detect dense pedestrian objects in clutter, while tracking-by-segmentation method takes the crowd as a whole foreground and fails to detect most of pedestrians. Our method can also generate accurate motion trajectories of pedestrians in crowd, while the baseline method can only provide holistic motions of crowds. Furthermore, foreground objects generated by the baseline method are not stable because of shadows, see Figure 4(a), however, our method can handle effects of shadows.

Figure 5 shows results of event detection and tracking generated by different methods in situation with dim pedestrians. In this video clip, pedestrians are much small, typically with the width of 5-10 pixels. From Figure 5, we can see that both methods fail to detect extremely small pedestrians (pedestrians with width smaller than 6 pixels). Our method, however, performs better than tracking-by-segmentation method for pedestrians with the width of 10 pixels. Our method can detect this kind of pedestrians and generate good motion trajectories. In contrast, tracking-by-segmentation method is more sensitive to noise.



(a) Results of segmentation-based event detection and tracking method (b) Results of event detection and tracking by our method

Fig. 4. Comparison of event detection and tracking results in complicated video with dense pedestrians with mutual occlusion. Images in each column show tracking results of detected events.



(a) Results of segmentation-based event detection and tracking method (b) Results of event detection and tracking by our method

Fig. 5. Comparison of event detection and tracking results in complicated video with dim pedestrians. Images in each column show tracking results of detected events.

4 Conclusions

We have presented a novel video summarization framework for quick and accurate access of interesting visual events in surveillance videos. We propose to detect interesting visual events by combining state-of-the-art object detection techniques and multi-objects tracking techniques. For detecting interesting foreground objects, we run a set of pre-trained object classifiers on the foreground, which are well designed in order to detect ambiguous objects, especially dim pedestrian objects and dense objects with mutual occlusion. Then a Hungarian algorithm based multi-objects tracking program is followed to eliminate conflicts between trajectories and observations. We also propose to compactly represent long videos by a set of interesting visual event, each of which is represented by a synthetical snapshot. Experimental results demonstrate that our framework outperforms existing video summarization systems in the detection accuracy of interesting visual events on challenging surveillance videos.

References

1. Li, Y., Zhang, T., Tretter, D.: An overview of video abstraction techniques. Technical report, HP Laboratories Palo Alto (2001)
2. Kim, C., Hwang, J.N.: An integrated scheme for object-based video abstraction. In: Proceedings of the Eighth ACM International Conference on Multimedia (2000)
3. Petrovic, N., Jojic, N., Huang, T.: Adaptive video fast forward. *Multimedia Tools and Applications Journal* 1, 108–121 (2005)
4. Höferlin, B., Höferlin, M., Weiskopf, D., Heidemann, G.: Information-based adaptive fast-forward for visual surveillance. *Multimedia Tools Appl.* 55, 127–150 (2011)
5. Pritch, Y., Rav-Acha, A., Gutman, A., Peleg, S.: Webcam synopsis: Peeking around the world. In: ICCV (2007)
6. Pritch, Y., Rav-Acha, A., Peleg, S.: Nonchronological video synopsis and indexing. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 1971–1984 (2008)
7. Rav-Acha, A., Pritch, Y., Peleg, S.: Making a long video short: Dynamic video synopsis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2006)
8. Pritch, Y., Ratovitch, S., Hendel, A., Peleg, S.: Clustered synopsis of surveillance video. In: Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (2009)
9. Wang, S., Yang, J., Zhao, Y., Cai, A., Li, S.Z.: A surveillance video analysis and storage scheme for scalable synopsis browsing. In: Proceedings of Computational Methods for the Innovative Design of Electrical Devices (2011)
10. Feng, S., Lei, Z., Yi, D.: Online content-aware video condensation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2012)
11. Barnich, O., Droogenbroeck, M.V.: Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 1709–1724 (2011)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)

Detecting and Tracking Unknown Number of Objects with Dirichlet Process Mixture Models and Markov Random Fields

Ibrahim Saygin Topkaya¹, Hakan Erdogan¹, and Fatih Porikli²

¹ Sabanci University VPALAB, Istanbul, Turkey

{isaygint, haerdogan}@sabanciuniv.edu

² Mitsubishi Electric Research Laboratories, Cambridge MA, USA

fatihporikli@ieee.com

Abstract. We present an object tracking framework that employs Dirichlet Process Mixture Models (DPMMs) in a multiple hypothesis tracker. DPMMs enable joint detection and tracking of an unknown and variable number of objects in a fully automatic fashion without any initial labeling. At each frame, we extract foreground superpixels and cluster them into objects by propagating clusters across consecutive frames. Since no constraint on the number of clusters is required, we can track multiple cluster hypotheses at the same time. By incorporating superpixels and an efficient pruning scheme, we keep the total number of hypotheses low and tractable. We refine object boundaries with Markov random fields and connectivity analysis of the tracked clusters. Finally, we group tracked hypotheses to combine possible parts of an object as one.

1 Introduction

Conventional object tracking methods often use multiple hypothesis tracking, which can establish correspondence between many hypotheses in parallel [1], or joint probabilistic data association, which can utilize weighted contributions of each observation to update target states [2]. These hypotheses are usually sampled in a particle filtering framework [3] that results in a probabilistic update of the object states. Recently, Dirichlet Process Mixture Models (DPMMs) received increasing attention for tracking applications. For example, DPMMs are adopted for tracking of acoustic energy around a particular frequency in the speech wave for speech recognition [4] and employed a Sequential Monte Carlo (SMC) inference and Gibbs sampler in visual tracking [5].

In this work, we propose a nonparametric model based multiple object tracker for unknown number of targets. Instead of applying batch inference by a Gibbs sampler as in [5], we explore a full-association space inspired by [4] and aim to track all feasible associations. In addition, we select observations using a Gaussian Mixture Model (GMM) based change detection algorithm as opposed to arbitrarily use all image pixels. To reduce the number of tracked association hypotheses, we employ superpixels as atomic observations. After obtaining tracking hypotheses for unknown number of targets, we refine pixel-level boundaries using Markov random fields (MRF) that incorporates the clusters obtained during the previous tracking stages into the refinement process. Finally, we carry out a temporal grouping of clusters to combine different parts of a tracked object into one.

Within the scope of this paper we refer *observations* for any atomic observation to be associated to a target, *targets* for *clusters* (used interchangeably) of observations obtained by DPMM tracking, *objects* for tracked objects that are formed by one or more targets, and *hypotheses* for a tracking hypothesis that defines target states and observation to target associations.

In the next section we give a brief review of DPMMs. In Section 3 we present our object tracking framework using DPMMs followed by the refinement of target boundaries in Section 4 and the grouping of targets in Section 5.

2 Dirichlet Process Mixture Models

DPMMs provide a way to model data as a mixture model having unknown number of mixture components or clusters [6]. Let $X_n; n = 1..N$ be the observed data that is to be modeled as a mixture of distributions having the form $F(\theta)$. If θ_k denotes parameters of the k th mixture component, then $X_n \sim F(\theta_k)$ if $X_n \in k$. Let c_n be the latent indicator variable such that $c_n = k$ indicates $X_n \in k$. The discrete probability distribution $p(c_n = k)$ has Dirichlet distribution as conjugate prior and taking the number of mixture components to infinity results in the Dirichlet process.

As opposed to the GMM and hidden Markov model (HMM) that require the number of habitats to be specified prior to training, DPMM has the appealing property that the number of mixtures or clusters does not need to be known a priori. It assumes that there are infinite number of mixture components $k = 1..\infty$ yet only a finite number of these components have observations assigned to them. Modeling the data with DPMMs consists of finding the parameters of those finite and unknown number of mixture components, i.e. clusters. A detailed review of Markov chain sampling methods (like Gibbs sampling) to estimate the cluster parameters can be found in [7]. These methods iterate over all observations and for each observation calculate probabilities of belonging to an existing or a new cluster controlled by an aggregation parameter α . For higher values of α more clusters are generated during the clustering. The probability that an observation belongs to a mixture component is given as

$$p(c_n; \alpha) = \begin{cases} \frac{N_k}{N+\alpha-1} p(X_n|\theta_k) & \text{existing } k, \\ \frac{\alpha}{N+\alpha-1} \int_{\theta} p(X_n|\theta) d\theta & \text{new cluster,} \end{cases} \quad (1)$$

where N_k is the number of assignments to cluster k and N is the number of all observations. The graphical model for the DPMMs is depicted in Fig. 1.

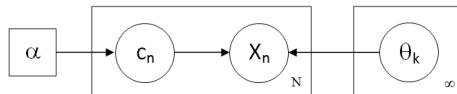


Fig. 1. Graphical model for the DPMMs: The observation (X_n) depends on one of the infinite number of cluster parameters (θ_k), assignment (c_n) of which is controlled by α

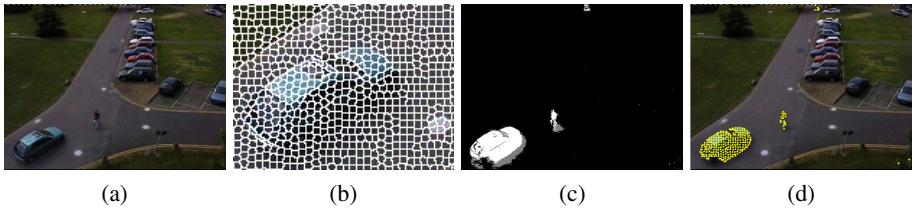


Fig. 2. A sample frame (a), superpixel borders for the bottom-left part of the frame (b), the foreground probability map for each pixel (c) and centers of superpixels that contain foreground pixels (d)

3 Object Tracking with DPMMs

3.1 Extracting Observations for Tracking

To reduce the number of observations and decrease the DPMM association time, we incorporate superpixels that segment an image into small, compact and almost-regular regions while keeping color variation within regions low. With a small computational overload, a rough superpixel extraction [8] significantly decreases the number of observations by around 95%, an example of which can be seen in Fig. 2.

We extract foreground regions on the frame using GMM based background representation [9] that models the previous color changes of each pixel using a mixture of Gaussians by applying an expectation maximization update. Here, we select superpixels that contain pixels whose foreground probability is high. For the sample frame shown in Fig. 2, the number of observations decrease from $\sim 160,000$ to $\sim 4,500$ foreground pixels and then finally to around 200 foreground superpixels.

3.2 Observation and Target Models

We model each observation using the spatial center of the superpixel (i.e. x and y pixel coordinates) and the mean value of the a and b pixel color components in the *Lab* color space. Similarly, we model each cluster/target using the mean and variance of the same components along with the spatial covariance.

For computational ease, we do not model Gaussian models with full covariance matrices for targets. Instead, we analyze the covariance between spatial components since it is strongly related to the appearance of the target on the image by approximating the target appearance with a rotated ellipse (for human tracking). Thus, each observation is defined with four parameters; $X : (\mu_x, \mu_y, \mu_a, \mu_b)$ and each target with six parameters; $\theta : (\mu_{xy}, \Sigma_{xy}, \mu_a, \sigma_a, \mu_b, \sigma_b)$. Under this model, the likelihood that an observation X_n is generated by a target k with parameters θ_k is

$$p(X_n|\theta_k) = \mathcal{N}(X_{xy}|\mu_{xy}^k, \Sigma_{xy}^k) \quad \mathcal{N}(X_a|\mu_a^k, \sigma_a^k) \quad \mathcal{N}(X_b|\mu_b^k, \sigma_b^k) \quad (2)$$

where the parameters of the Gaussians in Eq. 2 are estimated from the observations that are assigned to the targets. Here, μ_{xy} models the spatial center and μ_a and μ_b the average color values for the targets. The spatial covariance Σ_{xy} models the size and orientation of the target in the image. The larger the covariance the more spread in that direction the target becomes.

Eq. 1 and Eq. 2 together define the assignment probability of an observation to an existing or a new target. For a new cluster, the integral in Eq. 1 is calculated over the whole prior distribution. The prior for Gaussian distribution is Normal-Inverse Wishart distribution and integrating over it gives a t-distribution [10]. However, [10] shows that this can be approximated by a Gaussian with properly chosen parameters. We choose it as a Gaussian that is centered on the frame and having a variance that covers the whole frame. The color components have a similar coverage.

3.3 Target Assignment and Tracking with DPMM Clustering

In [5] two inference methods for tracking are defined; one depends on Markov Chain Monte Carlo (MCMC) to perform batch inference and the other uses SMC in a particle filtering framework. At each frame, iterative Gibbs sampling is performed and assignments are selected for each observation. Then, parameters of each cluster are again sampled from the current and past assignments. In [4], at each time and for each observation, a whole exploration of the assignment space is done in a Rao-Blackwellized [11] fashion and new hypotheses are generated for each assignment instead of Gibbs sampling.

We follow a similar approach. After initially estimating the positions of the targets using past motion dynamics (i.e. *Rao-Blackwellization*), we evaluate the observations one by one and calculate association probabilities of observations to an existing or a new target with Eq. 1. Each association represents a new hypothesis with a calculated weight.

For each frame f , the DPMM clustering inherits K_{f-1} number of clusters from the previous frame and performs clustering of the new observations to those existing (or new) clusters. Note that, some of the inherited clusters may be kept with new observation assignments, some of them may be dropped if no observations are assigned to them, and some new clusters may be generated. Altogether, they form K_f number of clusters as the tracking result for frame f . These clusters are projected to the next frame $f+1$ later.

We prune association hypotheses with very low weights after evaluating each observation, which prevents the number of the hypotheses to grow. In our experiments we have observed that much less than 10 hypotheses are kept between frames.

Our association scheme differs from [5] in the sense that the whole assignment space is explored and cluster parameters are updated deterministically instead of random sampling of the assignments and cluster parameters.

A difference between [4] and our method is the weight update rule of the hypotheses. We consider transition probabilities of the clusters while updating the weights of the hypotheses, which allows us after evaluating all observations in a frame, to remove the hypotheses that have unusual change in the states of the clusters. In [4], after each

observation (X_n) is assigned to a target, a new hypothesis is derived weight of which (w_h) is updated as:

$$w_h = w_h p(c_n = k; \alpha). \quad (3)$$

We also perform the same update rule while generating new hypotheses during handling the observations. In addition, for one frame, after all observations are assigned to targets and target parameters are updated, we calculate the following transition probabilities for each cluster and update the weight of the hypothesis:

$$w_h = w_h \prod_{k \in h} p(\theta_k^f | \theta_k^{f-1}). \quad (4)$$

The transitions are calculated for clusters inherited from previous frame ($f - 1$) and kept in current frame (f). Addition of new clusters is controlled by the parameter α in Eq. 1, and at this stage there is no special handling for the deletion of the clusters, which we leave as a future work. The transition probability in Eq. 4 is taken as:

$$\begin{aligned} p(\theta_k^f | \theta_k^{f-1}) &= \mathcal{N}(\mu_{xy}^f | \mu_{\hat{x}\hat{y}}^f, \Sigma_{xy}^{f-1}) \times \\ &\quad \mathcal{N}(\sigma_x^f | \sigma_x^{f-1}, 0.1 \sigma_x^{f-1}) \times \\ &\quad \mathcal{N}(\sigma_y^f | \sigma_y^{f-1}, 0.1 \sigma_y^{f-1}), \end{aligned} \quad (5)$$

where \hat{x} and \hat{y} denote the initial spatial estimates of the positions of the targets estimated from their previous motions with the Rao-Blackwellization. Considering also that the variance of a target is proportional to its size, the first probability in Eq. 5 represents the typical assumption that the position of the tracked target conforms with the past dynamics with an uncertainty proportional to its size. The latter two probabilities represent the assumption that the size of the tracked target changes at most around 10% of its size between frames.

4 Refining Object Boundaries

Tracking by the DPMM clustering scheme presented in Section 3.3 generates the labeled foreground superpixels, where the labels correspond to tracked clusters/targets. To compensate border artifacts caused by the quick but rough superpixel extraction we apply a refinement step.

MRFs [12] are commonly used graphical models in image labeling tasks to obtain smooth maps. The labeling is considered as an optimization problem where energies for labeling of the image are defined for individual pixels within local neighborhoods on uniform grid.

The energies for neighborhoods are used to enforce smoothness in the local regions. Having a graphical model where nodes n correspond to pixels and vertices v to neighborhoods, the aim is to find the lowest overall energy E of a labeling \mathcal{L} for image \mathcal{I} , which is calculated as sum of unary and pair-wise energies as

$$E(\mathcal{L}) = \sum_{u \in n} E(\mathcal{L}_u) + \sum_{(u_1, u_2) \in v} E(\mathcal{L}_{u_1}, \mathcal{L}_{u_2}), \quad (6)$$

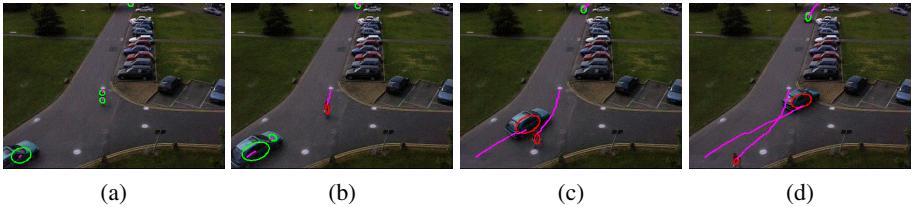


Fig. 3. Four sample frames from PETS 2001 dataset. Ellipses are the isocontours of the Gaussian distribution (for $\sigma=1$) that defines the target position (Section 3.2). Green ellipses correspond to single targets, red ellipses correspond to two or more merged/grouped targets (Section 5). Pink trajectories denote the past spatial centers of the relevant targets.

where $E(\mathcal{L}_u)$ is the cost of labeling individual pixels (unary term) and $E(\mathcal{L}_{u_1}, \mathcal{L}_{u_2})$ is the cost of labeling neighboring pixels (pair-wise term), which is used to enforce smoothness. In our work, we label each pixel in the frame as one of the targets obtained with the DPMM tracking and the background. During that process, we employ MRFs as with any other labeling task [12].

The DPMM clustering results in target clusters for a frame and for each pixel. Using position and color of the pixel, Eq. 2 can be used to calculate the likelihood of the pixel for each target. To impose this likelihood as the unary term for a single pixel X_n in Eq. 6, we take the negative log of Eq. 2:

$$E(\mathcal{L}_{X_n}) = -\log(p(X_n|\theta_k)), \quad (7)$$

For the background label, we again take negative log of the background probability obtained with the background subtraction results produced in Section 3.1 to detect foreground superpixels. For the pair-wise term in Eq. 6, we set 8-pixel neighborhoods and give a fixed energy value if the neighboring pixels have different labels where same labels incur zero penalty.

5 Grouping Targets into Objects

We empirically obtained results with different α values (of Eq. 1) and observed that even with suitable values there may be cases that parts of an object may be assigned to different targets because of color differences. Thus, we run a final step to detect and group those different parts of an object into one object, and present the final tracking output by representing objects as those grouped targets.

To decide whether any two targets can be merged into one, we analyze the historical motion of the targets by measuring the similarity of the motions of the pairs of targets. To measure the similarity, we use cross-correlation of historical spatial (x and y components) positions of the targets.



Fig. 4. Sample frames from PETS 2009 dataset. Ellipses are the isocontours of the Gaussian distribution (for $\sigma=1$) that defines the target position (Section 3.2). Green ellipses correspond to single targets, red correspond to two or more merged/grouped targets (Section 5). Pink trajectories denote the past spatial centers of the relevant targets.

Cross-correlation for x component of any two targets historical positions of which are known is calculated as

$$\rho_x = \frac{\sum_t (x_1(t) - \mu_1^x)(x_2(t) - \mu_2^x)}{\sqrt{\sum_t (x_1(t) - \mu_1^x)^2} \sqrt{\sum_t (x_2(t) - \mu_2^x)^2}}, \quad (8)$$

and similarly for ρ_y . At each frame, we calculate ρ_x and ρ_y using last t frames after tracking with DPMMS is achieved. In case their sum exceeds a threshold value for any two targets, we assume those targets move together, thus belong to the same object. Such targets are merged into the same object.

6 Experiments and Results

We implemented the proposed algorithm in C#. For superpixel extraction, we used SLIC superpixels [8] and integrated the implementation in VLFeat library [13]. For smoothing with MRFs, we used FastPD MRF optimization [14,15] library. To extract the foreground pixels, we applied the GMM implementation of OpenCV [16]. Object boundaries were obtained with α -shapes [17] implementation of CGAL library [18]. We run the experiments on a sequence of 200 frames in PETS 2001 [19] and 100 frames in PETS 2009 [20] datasets where α in Eq. 1 is fixed ($\alpha = 1$) for the proposed method.

6.1 Tracking and Target Grouping Results

Figures 3 and 4 present the tracking results by denoting the tracked targets as 2D Gaussian ellipses drawn on the image frame, as well as their past tracks superimposed onto the image.

The results demonstrate that the proposed tracking algorithm works accurately in complex situations where some tracked objects are partially occluded by others like in

the last two sample frames from the PETS 2009 sequence. In Fig. 4c, it can be seen that the pedestrian that entered the scene from right passes in front of the pedestrian that was walking in the middle of the scene from the beginning. The proposed tracker continued to track those two pedestrians in the following frames (Fig. 4d) without having any drift problems.

The figures also show the success of the proposed target grouping scheme. For example, initially at Fig. 3a, parts of the pedestrian in the middle of the scene are detected and being tracked as separate targets, as well as the car in Fig. 3b. After 10 frames, using the proposed target grouping scheme, parts of the targets that belong to the same object are grouped and assigned as one (Fig. 3c and Fig. 3d). In all figures, the red ellipses denote the targets that are obtained by grouping two or more targets.

Similarly, initially at Fig. 4a, parts of three pedestrians are detected and being tracked as separate targets. After 10 frames, parts of the pedestrians are grouped and represented as a single object (Fig. 4b). Again, the grouped targets are denoted by red ellipses in the figure.

6.2 Refining Object Boundaries

Figures 5 and 6 display results in which accurate object boundaries are drawn rather than ellipses. After target clusters and superpixel assignments are obtained, a finer labeling of pixels is obtained with MRFs.

The refinement step provides pixel-wise assignments for targets. Using those assignments, boundaries are calculated as sets of pixels that represent the border contains all pixels assigned to the target cluster. Note that, borders are not necessarily convex. While the boundaries are being determined, pixel assignments that are far from the target cluster are filtered out where the cut-off distance is determined by the standard deviation of the spatial components of the target cluster (i.e. σ_x and σ_y). The results show that even in complex situations that objects come together (e.g. Fig. 6c); the boundaries of them can be detected accurately by taking the grouping of the targets into consideration.

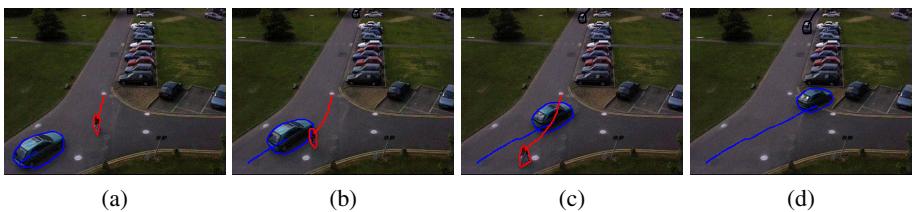


Fig. 5. Sample object boundaries (Section 4) from PETS 2001 dataset. The target borders are denoted with distinct colors across frames, as well as the historical centers as the superimposed trajectory having the same color with the target borders.

Table 1. Tracking accuracy ($100 \times \text{ATA}$) and running time (SPF) for PETS 2001 sequence comparing the proposed method and [5] (best results are in bold)

	Proposed	Neiswanger [5]
ATA	78.07	29.93
SPF	9.91	18.08

6.3 Quantitative Analysis

We report the *average tracking accuracy* (ATA) scores given as [21]:

$$\text{STDA} = \sum_{i=1}^{N_o} \frac{\sum_{f=1}^{N_f} \frac{G_i^f \cap D_i^f}{G_i^f \cup D_i^f}}{N_{(G_i^f \cup D_i^f) \neq 0}}, \quad \text{ATA} = \frac{\text{STDA}}{\left(\frac{N_G + N_D}{2}\right)} \quad (9)$$

which is the normalization of *sequence track detection accuracy* (STDA) by the number of detected (N_D) and ground truth objects (N_G). STDA is calculated using overlaps and unions pixels that belong to the ground truth (G_i^f) and detected (D_i^f) objects. This *overlap ratio* is aggregated over the sequence of N_f frames and normalized by the number of frames ($N_{(G_i^f \cup D_i^f) \neq 0}$) where a ground truth or detected object exists, and calculated for all N_o objects.

In Table 1, we report the ATA values of the proposed method for the PETS 2001 sequence that contains three distinct objects along with the accuracy of the method proposed in [5]. For both methods, we filter clutter by removing targets that appear less than five frames. For our method, we take the target merges into consideration. For [5], we repeat the experiments with different α and *covariance confidence*, which is a specific parameter used for output representation in [5], and report the best result. We also report the average running times to process one frame, i.e. *seconds per frame*–SPF in the same table. The results demonstrate that the proposed method significantly outperforms [5]. The primary reason is that our method calculates target parameters deterministically as opposed to probabilistic sampling approach in [5]. In addition, we

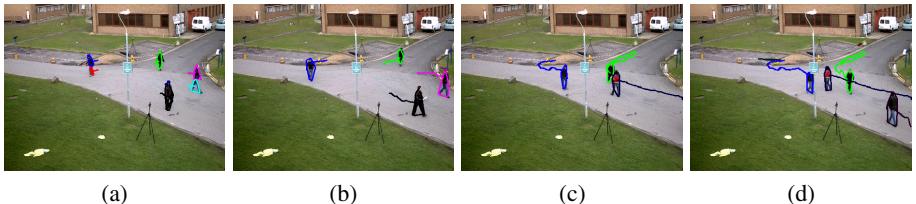


Fig. 6. Sample object boundaries (Section 4) from PETS 2009 dataset. The target borders are denoted with distinct colors across frames, as well as the historical centers as the superimposed trajectory having the same color with the target borders. Some targets in (a) merged (Section 5) into others in the following frames.

Table 2. Tracking accuracy ($100 \times \text{ATA}$) and running time (SPF) of the proposed method for PETS 2001 sequence; (a) comparing the different superpixel sizes, (b) comparing noisy observations

	SP size=9	SP size=25	SP size=100		Clean	Noisy
ATA	77.98	78.07	62.57	ATA	78.07	77.96
SPF	28.27	9.91	5.88	SPF	9.91	13.21

(a)

(b)

extract the foreground pixels by a more robust background generation method instead of the simple frame differences as in [5].

In Table 2-a, we give the tracking accuracy scores of our method for different average superpixel sizes. The results show that the superpixel size plays an important role for both tracking accuracy and running time.

As seen in the first two columns, increasing superpixel size significantly decreases the running time by reducing the number of observations. However, using large superpixels (as in the third column) may negatively impact the tracking accuracy too. The reason is that as superpixels get larger, there is risk of grouping some pixels of the background and nearby targets into same superpixels, thus losing the distinction between targets and deforming their boundaries at the observation level that causes tracking drift problems. For tracking accuracy, it is not always preferable to use smaller superpixels either as the optimal size for the ATA is around 25.

In Table 2-b, we report tracking accuracies when a detection noise is added by adding a random number of false observations. The number of false observations are controlled such that during foreground extraction, each background superpixel is chosen falsely as with some particular probability, which was set as 0.001 for the presented result. The effect of noise can be seen on the running time. Noisy observations introduces false targets, which involves more calculations in the process of the observation-to-target assignment during the generation of tracking hypotheses. Table 2-b shows that the tracking accuracy changes minimally for noisy observations, which indicates that the proposed method is very robust to observation errors.

7 Discussion and Future Work

We use DPMMs in visual object tracking in a deterministic multiple-hypotheses tracking framework. DPMMs allow us to detect and track unknown number of object in a fully automatic fashion, without any initial labeling required. Since our method is based on superpixels and incorporates an efficient pruning step, the number of hypotheses does not grow in memory and is tractable. It also achieves refinement of object boundaries with MRFs while employing a target grouping step to compensate clustering errors.

In the future we plan to extend MRF refinement by enforcing different higher order constraints such as shape and histograms to incorporate object specific priors into segmentation.

References

1. Reid, D.B.: An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control* 24, 843–854 (1979)
2. Fortmann, T., Bar-Shalom, Y., Scheffe, M.: Multi-target tracking using joint probabilistic data association, pp. 807–812 (1980)
3. Arulampalam, M.S., Maskell, S., Gordon, N.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 50, 174–188 (2002)
4. Özkan, E., Özbek, I.Y., Demirekler, M.: Dynamic speech spectrum representation and tracking variable number of vocal tract resonance frequencies with time-varying dirichlet process mixture models. *IEEE Transactions on Audio, Speech & Language Processing* 17, 1518–1532 (2009)
5. Neiswanger, W., Wood, F.: Unsupervised Detection and Tracking of Arbitrary Objects with Dependent Dirichlet Process Mixtures. *ArXiv e-prints* (2012)
6. Antoniak, C.E.: Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics* 2, 1152–1174 (1974)
7. Neal, R.M.: Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational And Graphical Statistics* 9, 249–265 (2000)
8. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1 (2012)
9. Stauffer, C., Grimson, E.: Adaptive background mixture models for real-time tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999 (2002)
10. Suderth, E.: Graphical Models for Visual Object Recognition and Tracking. PhD thesis, Massachusetts Institute of Technology (2006)
11. Casella, G., Robert, C.P.: Rao-blackwellisation of sampling schemes. *Biometrika* 83, 81–94 (1996)
12. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 2001 (2001)
13. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), <http://www.vlfeat.org/>
14. Komodakis, N., Tziritas, G.: Approximate labeling via graph cuts based on linear programming. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1436–1453 (2007)
15. Komodakis, N., Tziritas, G., Paragios, N.: Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies. *Comput. Vis. Image Underst.* 112, 14–29 (2008)
16. Bradski, G.: The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000)
17. Bernardini, F., Bajaj, C.L.: Sampling and reconstructing manifolds using alpha-shapes. In: *Proc. 9th Canad. Conf. Comput. Geom.* (1997)
18. CGAL: Computational Geometry Algorithms Library, <http://www.cgal.org>
19. PETS 2001: Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2001), <ftp://ftp.pets.rdg.ac.uk/pub/PETS2001/>
20. PETS 2009: Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2009), <ftp://ftp.pets.rdg.ac.uk/pub/PETS2009/>
21. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 319–336 (2009)

Grassmannian Spectral Regression for Action Recognition

Sherif Azary and Andreas Savakis

Computing and Information Sciences and Computer Engineering
Rochester Institute of Technology, Rochester, NY 14623

Abstract. Action recognition from multiple views and computational performance associated with high-dimensional data are common challenges for real-world action classification systems. Subspace learning has received considerable attention as a means of finding an efficient low-dimensional representation that leads to better classification and efficient processing. In this paper we propose Grassmannian Spectral Regression (GRASP), a novel subspace learning algorithm which combines the benefits of Grassmann manifolds and spectral regression for fast and accurate classification. A Grassmann manifold is a space that promotes smooth surfaces where points represent subspaces and the relationship between points is defined by a mapping of an orthogonal matrix. Spectral regression is a regularized subspace learning approach that overcomes the disadvantages of eigen-based approaches. We demonstrate the effectiveness of GRASP on computationally intensive, multi-view action classification using the INRIA IXMAS dataset and the i3DPost Multi-View dataset.

1 Introduction

Action classification is a challenging problem because of many factors including varying human sizes, shapes, poses, and action execution speed. Another difficulty that is not widely addressed is variation due to viewpoint. Real world applications such as autonomous surveillance and human computer interfacing would greatly benefit from action classification systems supporting viewpoint independence.

Weinland et al. [1] explain that viewpoint independence is commonly addressed by normalization, invariance, or exhaustive search. View normalization involves a view correction through a transformation to a canonical view. Ding et al. [2] present such a pose-normalization algorithm using random forest embedded active shape models to map 2D features into a 3D corresponding space. Haq et al. [3] apply view-invariant action classification that involves forming fusion tables from spatio-temporal cuboid features and feature fusion using Principal Component Analysis.

A common challenge for real-world action classification systems is the computational performance associated with processing high-dimensional data. Subspace learning and discriminative analysis address the issue of high dimensional data by finding an efficient low-dimensional representation. Principal Component Analysis (PCA) uses eigen-decomposition to find a low dimensional representation and has been employed for action classification systems in [4] and [5]. Manifold learning by

Locality Preserving Projections (LPP) preserves local neighborhood information and was first reported for action classification systems in the work of Wang and Suter [6]. Linear Discriminant Analysis (LDA) was applied to action classification in [7].

While eigen-based linear subspace approaches are effective at learning linear and non-linear representations of data, recent efforts have emerged towards least squares frameworks because of drawbacks associated with eigen-formulations. De la Torre [8] suggests that eigen-decomposition results in normalization factors and inaccuracies with rank deficient matrices, and proposes a least-squares weighted kernel reduced rank regression (LS-WKRRR). Cai et al. [9] encourage the avoidance of eigen-decomposition because of computational inefficiencies and introduce Spectral Regression (SR) for regularized subspace learning. The SR approach enables the benefits of regularization, which is not as simple to do with eigen-decomposition.

Another recent development is the representation of image sets as low-dimensional linear subspaces using Grassmann manifolds where distances between subspaces can be measured by principal angles. Shigenaka et al. [10] present the Grassmann Distance Mutual Subspace Method (GD-MSM) and Grassmann Kernel Support Vector Machines (GK-SVM). Park and Savvides [11] adopted Grassmann kernels into Kernel Principal Component Analysis (KPCA). Azary and Savakis [12] introduced Grassmannian Sparse Representations (GSR) for 3D action recognition.

In our work we propose Grassmannian Spectral Regression (GRASP) as a regression framework that supports regularization for improved class separability. Another important benefit is a drastic improvement on computational performance that is achieved by avoiding eigen-decomposition. The remainder of this paper is organized as follows. Section 2 describes Spectral Regression and Grassmann manifolds. Section 3 introduces Grassmannian Spectral Regression and Section 4 presents the experimental setup for multi-view action classification. We conclude in Section 5.

2 Subspace Learning

2.1 Spectral Regression

Spectral Regression (SR) [9] is a regularized subspace learning approach that overcomes the disadvantages of eigen-based approaches in terms of inefficiencies in execution time performance, memory allocation, and regularization. Assume n input samples x such that $\{x_i\}_{i=1}^n \in \mathbb{R}^m$, and $X = [x_1, \dots, x_n]$. The general graph embedding framework G represents each vertex n as a low dimensional vector that preserves similarities between the vertex pairs through weight assignments on an $n \times n$ weight adjacency matrix W . Let $B = [b_1, \dots, b_n]^T$ where b_i are the eigenvectors of the following eigen-problem:

$$WB = \lambda DB \quad (1)$$

where λ is the eigenvalue, W is the adjacency matrix, and D is a diagonal matrix. Let $X^T A = B$, where A are the eigenvectors of the following eigen-problem with the same eigenvalue of equation (1):

$$XWX^T A = \lambda XDX^T A \quad (2)$$

The goal is to solve for the optimal eigenvectors A^* corresponding to the maximum eigenvalue of this eigen-problem [9]. To do so efficiently, instead of solving the eigen-problem in equation (2) we use the SR approach based on a two-step iterative process outlined below:

1. Solve for B in equation (1)
2. Solve for the optimal eigenvectors A^* corresponding to the maximum eigenvalue of equation (2) that satisfies $X^T A = B$, using least squares regression, as in the equation below where b_i is the i^{th} element of B .

$$A^* = \arg \min_A \sum_{i=1}^n (A^T x_i - b_i)^2 \quad (3)$$

The minimization problem could be underdetermined with many possible solutions or overfitting may be occurring. To account for this we can use regularization, with parameter λ regulating the amount of shrinkage, and an applied penalty on the norm of A , where $\|A\|_1$ is an ℓ^1 norm:

$$A^* = \arg \min_A \left(\sum_{i=1}^n (A^T x_i - b_i)^2 + \lambda \|A\|_1 \right) \quad (4)$$

A class value is assigned by performing classification in the lower dimensional space using k-Nearest Neighbor (k-NN) or another classifier. Other types of regularizers can be incorporated, which demonstrates the flexibility of regularized subspace learning for adaptation to various applications.

2.2 Grassmann Manifolds

A manifold is a topological space embedded in a high dimensional Euclidean space R^D , such that each manifold point has a neighborhood homeomorphic to a Euclidean space of dimension $m < D$ [13]. A Grassmann manifold $G(m, D)$, is the set of m -dimensional linear subspaces of R^D [14]. It is a type of Riemannian manifold that is naturally smooth and curved, where each point represents a subspace. Given n training samples in R^D , we solve for m unit vector representations of each class where m is the number of samples of each class. Unit vector representations are determined through singular value decomposition (SVD), such that:

$$\begin{aligned} C_{D \times m} &= U_{D \times D} S_{D \times m} V'_{m \times m} \\ U'U &= I, \quad V'V = I \end{aligned} \quad (5)$$

where $U_{D \times D}$ is an orthogonal matrix whose columns are the eigenvectors of AA' and $V_{m \times m}$ is the transpose of an orthogonal matrix whose columns are the eigenvectors of $A'A$. The diagonal matrix $S_{D \times m}$ contains the singular values in descending order. With the orthogonal matrix $U_{D \times D}$ we define a unit vector $\vec{u}_{1 \times D}$ representation of each sample with an imposed orthogonal constraint. The unit vectors of each k -class are

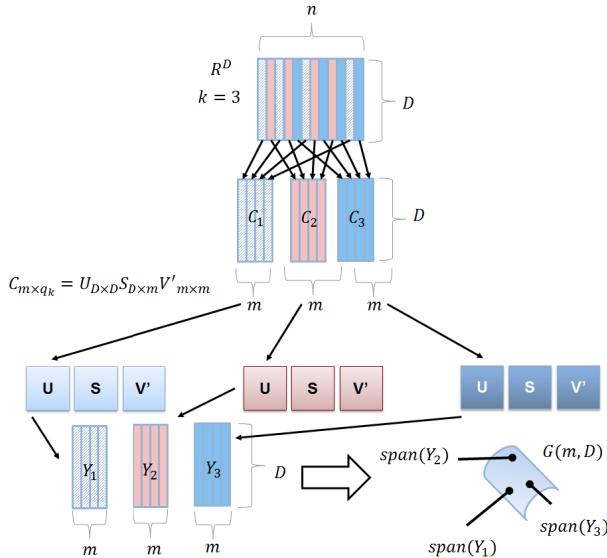


Fig. 1. This figure demonstrates the mapping of three classes from a high Euclidean space onto a Grassmann manifold. The span of the orthonormal matrix Y represents a subspace as a single point on a Grassmann manifold.

grouped into an orthonormal matrix $Y_{D \times m}$. The span of the orthonormal matrix $Y_{D \times m}$ represents a subspace of a class on a Grassmann manifold. If the columns of Y span a vector \vec{u} , then \vec{u} can be classified to that subspace.

There are benefits to using Grassmann manifolds. The span of orthonormal matrices embedded as single points promotes high between-class discrimination. It also allows for directly comparing two subspaces, which is computationally cheaper than measuring all distances between individual elements [14]. Additionally Grassmann manifolds fill in missing information through linear spans of subspaces.

The distance between any two points on a Grassmann manifold is geodesic, however, Grassmann kernels provide a means to simplify subspace metrics so that geodesic computations can be avoided. Two common Grassmann kernels are projection kernels and canonical correlations kernels. A projection kernel k_p maps an isometric embedding from the Grassmannian space to a projection space. The projection of two matrices Y_1 and Y_2 as defined by proposition 1 of Hamm and Lee [14] is:

$$K_p(Y_1, Y_2) = \text{tr}[(Y_1 Y_1')(Y_2 Y_2')] = \|Y_1' Y_2\|_F^2 \quad (6)$$

A canonical correlation kernel K_{cc} finds the relationship between linear composites of independent and dependent variables that have maximum correlation with each other based on their principal angles.

Grassmann kernels require kernel-based methods for classification and cannot be directly used with a classifier such as k-NN, because they do not define a direct linear relationship between subspaces. Thus, kernel-based methods such as PCA or LDA are needed for classification, as reported in Turaga et al. [15].

3 Grassmannian Spectral Regression

We now present the Grassmannian Spectral Regression (GRASP) framework which is a combination of Grassmannian kernels and spectral regression. There are two problems with eigen-decomposition subspace learning. First they add a level of computational complexity as suggested by De la Torre [8]. Secondly, such algorithms do not easily incorporate regularization. Our motivation is to combine computational efficiency and high between-class separability, promoted by Grassmann manifolds, with efficient and fast data representation promoted by spectral regression.

We begin by constructing training projection and canonical correlation kernels K_{p_train} and K_{cc_train} of sizes $m_1 \times m_1$, as a kernel mapping of all data elements, where m_1 is the number of training subspaces. Similarly we construct testing projection and canonical correlation kernels K_{p_test} and K_{cc_test} of size $m_1 \times m_2$, which map training subspaces to testing subspaces, where m_2 is the number of testing subspaces. These kernels map the Grassmannian space to a projective space. Given a general Grassmann training kernel K , we solve for the transformation matrix A of eigenvectors which maintains the linear relationship between the Grassmannian kernels and the lower dimensional representation B as in equation (8).

$$K^T A = B \quad (7)$$

The transformation A^* corresponding to the maximum eigenvalue can be solved using least squares regression by introducing Grassmann kernels into the least squares loss function with regularization such that:

$$A^* = \arg \min_A \left(\sum_{i=1}^P (A^T k_i - y_i)^2 + \lambda \|A\|_1 \right) \quad (8)$$

$$K = [k_1, \dots, k_P], \quad Y = [y_1, \dots, y_P]^T$$

where P is the number of subspaces on the Grassmann manifold, $\{k_i\}_{i=1}^P \in G(m, D)$. This formulation allows for least squares regularization of an isometric embedding in Grassmann space instead of a high dimensional Euclidean space.

K can either be a projection kernel or a canonical correlations kernel. A weighted representation of the two was proposed in [14], such that $K = \alpha K_p + \beta K_{cc}$, where α regulates the projection kernel and β regulates the canonical correlation kernel. The eigenvectors A^* gives a linear method of reducing our kernel data such that $K^T A = B$. We can now reduce the dimensions of the training and testing kernels following:

$$B_{train} = K_{train}^T A, \quad B_{test} = K_{test}^T A \quad (9)$$

With the reduced training and testing kernels, classification can be carried out using k-NN to classify a test subspace.

The benefits of the GRASP approach include the following. The SR component of the algorithm allows for regularization to account for outliers and noise while avoiding the computational burden of eigen-based approaches. Fast high dimensional data reduction is achieved through linear derivations of weighted isometric

embeddings and canonical correlations from a Grassmann space to a Euclidean space. The Grassmannian component of the algorithm supports high between class discrimination because these manifolds have smooth structure and can fill in missing data through linear spanning.

4 Experimental Setup

We conducted experiments on computationally intensive, multi-view action datasets to demonstrate the robustness and computational advantage of GRASP.

The INRIA Xmas Motion Acquisition Sequences (IXMAS) dataset was presented by Weinland et al. [16] and provides synchronized images of 5 views of 10 people executing 14 actions. The fifth view is a top view and was ignored for our experiments. Each individual is captured performing all actions in one scene. The dataset includes both silhouettes and the original color frames for the following actions: *Check Watch, Cross Arms, Scratch Head, Sit Down, Get Up, Turn Around, Walk, Wave, Punch, Kick, Point, Pick Up, Throw (over head), and Throw (from bottom up)*.

The i3DPost multi-view dataset, presented by [17], provides synchronized high-definition images of 8 views performed by 8 people executing 12 actions. Each action is performed in exactly 125 frames. The dataset includes (i) individual actions: *Walk, Run, Jump, Bend, Hand-Wave, and Jump in Place*; (ii) action combinations, where multiple actions are executed in the same sequence: *Sit-Stand Up, Run-Fall, Walk-Sit, and Run-Jump-Walk*; (iii) interactions between two individuals: *Handshake, and Pull*. The images are provided in a high-resolution color format (PNG) and include background images for image differencing, camera calibration parameters for 3D reconstruction, and 3D mesh models. Fig. 2 shows an example from each multi-view dataset.

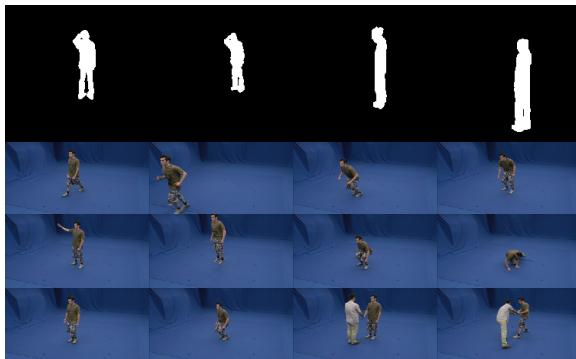


Fig. 2. A multi-view example of an individual’s silhouette in the IXMAS dataset performing the action “Scratch Head” is shown on top. The bottom shows a single frame of 12 action sequences executed by one subject from one view in the i3DPost Multi-View dataset.

Proposed by Davis and Bobick [18], motion history images (MHI’s) are temporal templates that describe where motion exists in a scene and how the motion is evolving over time. The MHI descriptor is expressed as follows.

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } \psi(x, y, t) = 1 \\ \max(0, H_\tau(x, y, t - 1) - \alpha) & \text{o.w.} \end{cases} \quad (10)$$

where τ describes the initial motion response, the decay operator is regulated by α , and $\psi(x, y, t)$ is an update function for background subtraction by subtracting frame t from a calibration frame. The main advantage of MHI is that it captures the direction of motion. We formulate our feature descriptors as MHI surfaces representing entire action sequences. Each action descriptor is based on the region of interest (ROI) of the MHI in each frame of the action scene, as shown at the top part of Fig. 3. The application of ROI-based MHI on a video sequence of an individual performing a walk and sit action captured by two cameras is presented at the bottom part of Fig. 3 in the form of multi-view spatio-temporal action surface descriptors using MHI.

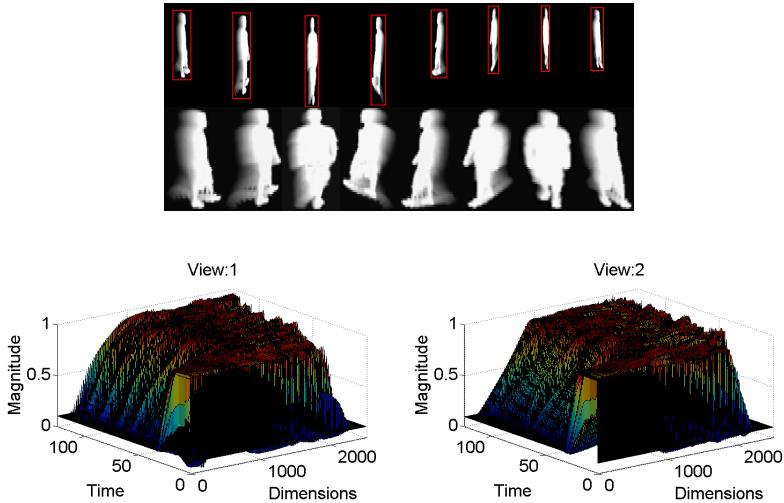


Fig. 3. The top image shows the ROI's and their corresponding MHI's based on multiple views of a walking sequence. The surfaces at the bottom show the spatio-temporal action descriptors of two views of that walking sequence.

4.1 Experimental Results

In our work we used the MHI feature descriptors for each experiment and applied PCA, LDA, LPP, k-NN based on Euclidean ℓ_2 norm, Spectral Regression (SR) GRASP (this paper), and GEDA [13]. We also included the computational processing time for each algorithm. All experiments were based on leave one subject out cross validation (LOOCV). We applied each algorithm on all actions for each database, except for the underhand throwing action in the IXMAS dataset since it only included 24 samples. This was because fewer people threw a ball underhand while most people threw the ball overhand. We did include overhand throwing in our

experiments. Thus, we trained and tested with 13 of the 14 actions in the IXMAS dataset and all 8 actions in the i3DPost dataset using LOOCV.

Table 1 presents results for classification accuracy and their corresponding processing times. These results demonstrate the strength of the GRASP technique. When comparing between the various subspace learning algorithms we discover that GRASP performs the best in terms of classification accuracy and processing times for large dictionaries of high dimensional data. We also observe that GRASP classifies better than SR alone and it does better than GEDA[13]. These results demonstrate the advantage of combining Grassmann Manifolds and SR.

The confusion matrices of GRASP classification for the IXMAS and i3DPost datasets are shown in Fig. 4. For the IXMAS dataset, the results demonstrate that dissimilar classes such as *Sit Down* and *Stand Up* are more easily identifiable, while similar actions such as *Point* and *Punch* or *Wave* and *Scratch Head* are confused more often. For the i3DPost dataset, we observe that grouped actions such as *run-jump-walk* can get misclassified with classes containing subset actions such as *walk-sit* and *run-fall*. The *jump* action, which are individuals hopping across a scene was misclassified (at a 7.8% error) with the *run* action of someone running across a scene.

The classification time results in Table 1 demonstrate that GRASP is much faster than eigen-based subspace methods. For the IXMAS and i3DPost datasets, GRASP is 5.4 to 9.9 times faster than GEDA and 5.8 to 18.8 times faster than SR. As expected, classification based on k-NN is the slowest since there is no subspace learning involved and the dimensionality of the data is high.

Table 1. Multi-view classification accuracy and processing time (in seconds) results comparing GRASP with existing subspace learning approaches

Algorithm	IXMAS Dataset		i3DPost Dataset	
	Classification Accuracy	Processing Time (sec)	Classification Accuracy	Processing Time (sec)
PCA	57.60%	202.58	60.27%	68.24
LDA	59.10%	315.62	78.78%	115.52
LPP	68.97%	268.30	82.81%	96.45
k-NN	66.17%	868.49	79.30%	439.20
SR alone	57.10%	88.66	77.99%	43.26
GRASP	80.28%	15.34	88.80%	2.30
GEDA	74.04%	152.11	86.98%	12.50

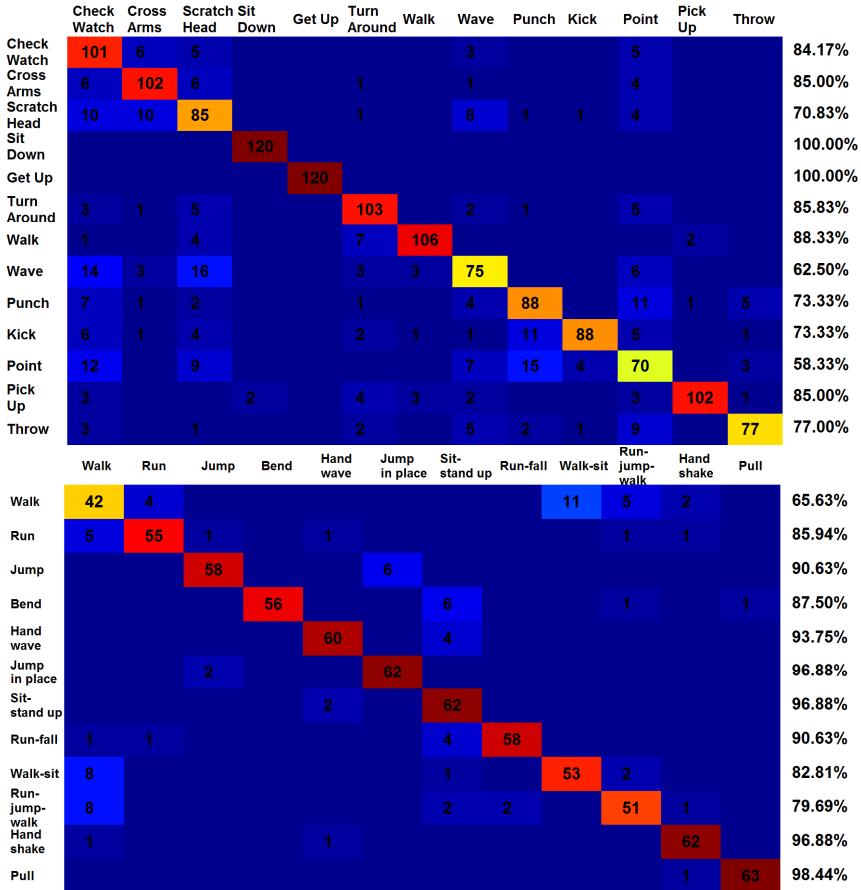


Fig. 4. Confusion matrices of multi-view classification results on the IXMAS dataset (left) and i3DPost dataset (right) using GRASP with overall classification accuracy of 80.28% and 88.80% respectively

5 Conclusion

In this paper we present Grassmannian Spectral Regression for optimized dimensionality reduction and classification. From a regression perspective, GRASP takes advantage of simplified regularization and improved computational performance by avoiding eigen-decomposition. From a class discrimination perspective, GRASP incorporates Grassmann manifolds to promote effective class separations. Our experiments with various subspace learning techniques on multi-view datasets demonstrate the classification accuracy and computational efficiency of the GRASP technique. We also demonstrate that projection kernels are more accurate than canonical correlation kernels for multi-view action classification.

References

1. Weinland, D., Ronfard, R., Boyer, E.: A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding* (2011)
2. Ding, L., Ding, X., Fang, C.: Continuous Pose Normalization for Pose-Robust Face Recognition. *IEEE Signal Processing Letters* (2012)
3. Haq, A., Gondal, I., Murshed, M.: On Temporal Order Invariance for View-invariant Action Recognition. *IEEE on Circuits and Systems for Video Technology* (2012)
4. Ali, S., Shah, M.: Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence* 32(2), 288–303 (2010)
5. Seo, H.J., Milanfar, P.: Action Recognition from One Example. *Pattern Analysis and Machine Intelligence* 33(5), 867–882 (2011)
6. Wang, L., Suter, D.: Learning and Matching of Dynamic Shape Manifolds for Human Action Recognition. *IEEE Transactions on Image Processing* (2007)
7. Liu, P., Wang, J., She, M., Liu, H.: Human action recognition based on 3D SIFT and LDA model. In: *IEEE Robotic Intelligence in Informationally Structured Space, RiiSS* (2011)
8. De la Torre, F.: A Least-Squares Framework for Component Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012)
9. Cai, D., He, X., Han, J.: Spectral Regression for Efficient Regularized Subspace Learning. In: *IEEE International Conference on Computer Vision, ICCV* (2007)
10. Shigenaka, R., Raytchev, B., Tamaki, T., Kaneda, K.: Face Sequence Recognition Using Grassmann Distances and Grassmann Kernels. In: *IEEE World Congress on Computational Intelligence* (2012)
11. Park, S.W., Savvides, M.: The Multifactor Extension of Grassmann Mani-folds for Face Recognition. In: *IEEE Automatic Face & Gesture Recognition* (2011)
12. Azary, S., Savakis, A.: Grassmannian Sparse Representations and Motion Depth Surfaces for 3D Action Recognition. In: *CVPR Workshop on Human Activity Understanding from 3D Data* (2013)
13. Harandi, M.T., Sanderson, C., Shirazi, S., Lovell, B.C.: Graph Embedding Discriminant Analysis on Grassmannian Manifolds for Improved Image Set Matching. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (2011)
14. Hamm, J., Lee, D.D.: Grassmann Discriminant Analysis: a Unifying View on Subspace-Based Learning. In: *Int. Conf. Machine Learning, ICML* (2008)
15. Turaga, P., Veeraraghavan, A., Chellappa, R.: Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (2008)
16. Weinland, D., Ronfard, R., Boyer, E.: Free Viewpoint Action Recognition using Motion History Volumes. *Computer Vision and Image Understanding* (2006)
17. Gkalelis, N., Kim, H., Hilton, A., Nikolaidis, N., Pitas, I.: The i3DPost multi-view and 3D human action/interaction. In: *Conference for Visual Media Production* (2009)
18. Davis, J.W., Bobick, A.F.: The Representation and Recognition of Action Using Temporal Templates. In: *IEEE Conference on Computer Vision and Pattern Recognition* (1997)

Layered RC Circuit Model for Background Subtraction

Karel Mozdřeň, Eduard Sojka, Radovan Fusek, and Milan Šurkala

Technical University of Ostrava, FEECS, Department of Computer Science,

17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

{karel.mozdren, eduard.sojka, radovan.fusek, milan.surkala}@vsb.cz

Abstract. The background subtraction is a technique widely used for video analysis, mainly moving object detection for surveillance systems. Such algorithms must be robust, fast and it has to be able to deal with dynamic backgrounds like water surface or moving tree branches. Also, they should be able to deal with illumination changes and objects casted shadows. Generally, in computer vision the algorithms with a physical background have the best performance. We propose an algorithm for background subtraction based on a model of layered RC circuits. We tested our method on video sequences acquired from level crossing and on commonly used datasets. Finally, we have compared the proposed method with other frequently used methods.

1 Introduction

The background subtraction is a common technique for moving object segmentation from video sequences. It is an alternative to the object detectors based on the knowledge of appearance of the objects. There are two popular methods: Ada-boost proposed by Yoav Freund and Robert E. Schapire [1], and support vector machine by Constantine Papageorgiou and Tomaso Poggio [2]. Both are trainable object detectors and are used in variety of applications.

In many cases, we are not able to predict the size, shape, or color of the objects, tahtin we are trying to detect. In such cases, we are ought to use the background subtraction algorithms. These, instead of training the object detector, model the background and subtract from it the “*moving*” objects in the foreground. The result of background subtraction is a binary image, where the pixels indicating “*moving*” objects are marked with white color and background pixels with black color. Then the connected components algorithm is used to find the individual objects in the segmentation. An example application of such an algorithm is the project Pfnder by Christopher Richard Wren et al. [3]. Their system tracks people and interprets their behavior. The tracking itself is done using the background subtraction algorithm. Generally, PDE based algorithms have the best performance in computer vision applications. Our method is inspired by the model proposed by Pietro Perona and Jitendra Malik [4]. They presented a model, using a grid of resistors and condensers organized in a grid, functioning as an image filter. In this paper we present our modification to this

model. We have modified the filter so it can be used for background modeling. The model is extended by excitation contacts used for connection of the input video sequence images with the model. Our paper has the following structure: in Section 2 we present works related to our matter, in Section 3 we describe our modified model, in Section 4 we conduct experiments and compare our method with other commonly used methods, and in Section 5 we conclude our work.

2 Related Works

The last two decades witnessed great improvement in background subtraction algorithms. Background subtraction techniques are based on learning of the background model from the video sequences. Each image in the video sequence is subtracted from the background model to find the differences and is also used for adaptation (update) of the background model. Big difference between the model and the actual image indicates “*moving*” objects in the foreground. One of the earliest method was proposed by Alan Lipton Hironobu et al. [5]. We refer to it as a temporal difference background subtraction (TDBS). This method uses one of the recent images as the background model and subtracts it with the most recent image. The difference is then thresholded and objects in the foreground are found. This method is not prone to fast illumination changes, but is prone to slow or temporally stationary objects. When the background model is too similar to the input image, there is too little difference and no foreground objects are indicated. This is exactly what happens when slow objects are moving in the images. Another method was proposed by Christopher Richard Wren et al. [3] and is known as temporal Gaussian background subtraction (TGBS). This method creates statistical model of the background. Each pixel of the background model is represented by one Gaussian defined by two values: μ for the mean value and σ for the standard deviation, both computed from N recent images. For better performance, those values are computed using the running Gaussian, which approximates these two values and is not that much demanding upon memory and computational time. This method deals with the problem of slow or temporally stationary objects detection, because it uses more than one image for background model construction. As a trade off, it is prone to the illumination changes. Sometimes, the μ value is represented by median (TMBS) [6], it makes the method more stable if the time between individual video sequence images vary, but the computational time and memory requirements rise accordingly. The problem with all these methods rises with dynamic backgrounds. Dynamic backgrounds are for example: moving tree branches, water surface waves, etc. . The method dealing with dynamic backgrounds is known as the mixture of Gaussians background subtraction (MoGBS). It was developed by C. Stauffer and W.E.L. Grimson [7]. This method models not only one, but K Gaussians for each pixel, and each Gaussian adapts to one background. For example, in the case of moving branches, one background represents the branch and the other the sky behind it. This also applies for interior video sequences, where the light is switched on and off. In this case, one Gaussian adapts to dark background and the other one to the illuminated background.

We describe our method in the next Section. First, we show our modification to the model presented by Pietro Perona and Jitendra Malik, then we propose a layered version of the model for dynamic backgrounds, and finally we provide the reader with information about additional improvements to our method.

3 Proposed Method

The inspiration to use a layered RC circuit model for background subtraction came from paper by Pietro Perona and Jitendra Malik [4]. In their work, they presented a method for scale-space image filtering based on heterogeneous isotropic diffusion. They also presented a model of diffusion process using electrical components, namely resistors and condensers (RC circuit). The scheme for filter they proposed can be seen in Figure 1. As we can see, the circuit is a grid of

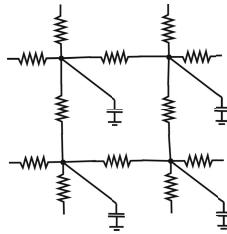


Fig. 1. The image filter model proposed by Pietro Perona and Jitendra Malik

condensers connected to neighboring condensers through the resistors. Each condenser represents one pixel in the image and the voltage is the pixel intensity (or color if we think of the voltage as a vector). If the condenser has lower voltage than its neighboring condenser, then it is being charged by that neighbor (the neighbor is being discharged by it) and vice versa. Charging speed depends on the resistance of the resistors. The greater the resistance is the slower is the charge/discharge rate.

Before we describe modification to the model, we have to understand the background itself and how it differs from foreground. Generally, the background consists of objects that are in most cases stationary and the foreground represents the “*moving*” objects. Basically, the background model represents the values occurring with higher probability and are not changing that much over time. The foreground objects are represented as values occurring less frequently and are significantly different from the background model. The background model can be statistically modeled using the Gaussian distribution. This method was presented by was proposed by Christopher Richard Wren et al. [3] and is known as temporal Gaussian background subtraction (TGBS). In this method, the mean value μ and standard deviation parameter σ are modeled for each pixel from last N images. Each new image is then compared with the model, and if new values

are 2.5σ further away from the mean value, then it is marked as foreground. The model of the background could also be viewed as a time dependent signal filter, which is often realized as a combination of resistors and condensers in electrical engineering. This is also the basic idea behind our method.

3.1 Simple RC Model

Our method uses for background modeling a simulation of diffusion using a grid of condensers and resistors as proposed by Pietro Perona and Jitendra Malik [4]. This model is modified in such a way, that each condenser representing one pixel of the background and the voltage over condenser is a representation of pixel intensity. This condenser is connected to excitation voltage representing the input image pixel values through the additional resistor. The one-dimensional example of one block of the original filter and in comparison the modified filter can be seen in Figure 2. This way the video sequence values are filtered and the voltage u_c over the condenser C represents the filtered value (modeled background), which is an equivalent of the mean μ value used in TGBS. The standard deviation parameter $u_{c\sigma}$ can be modeled similarly using the absolute difference between input excitation voltages (values) and modeled background u_c as an excitation voltage for u_σ circuit. The differential equation describing the background update has the following form

$$\begin{aligned} \frac{\partial u_{c,x,y}}{\partial t} = & \frac{1}{CR} (u_{c,x-1,y} + u_{c,x+1,y} + u_{c,x,y-1} + u_{c,x,y+1} - 4u_{c,x,y}) \\ & + \frac{1}{CR_e} (u_{e,x,y} - u_{c,x,y}) , \end{aligned} \quad (1)$$

where the u_c is the background model value (voltage over condenser), x and y represent the position in the circuit grid, C is the condenser capacity, R is the resistance value of the resistors connecting the condenser C with the neighboring condensers, and the resistor R_e connects the excitation voltage u_e (new value) to the background model condensers. The magnitude of the excitation resistance

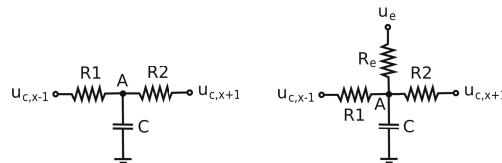


Fig. 2. The scheme of one block of the original (left) and modified filter (right)

R_e regulates the speed of background model adaptation. If the resistance is too small, the adaptation speed is fast, and if it is big, the adaptation speed is slow. Fast adaptation leads to imprint of slowly moving objects into the background

model. On the other hand, if the adaptation is too slow, some parts of the background image might be outdated.

This model is most similar to the TGBS method. The main difference between these two methods is in filtering. The TGBS filters the data only in time domain, because it filters the values for each input image pixel separately, but our method considers the background model as a whole, where all the pixels are connected to theirs nearest neighbors, which allows filtering also between background model pixels. The first experiment we propose compares the TGBS with our method. For the test, the level crossing dataset we have created is used. This dataset can be downloaded from our website <http://mrl.cs.vsb.cz/people/mozdren/levelcrossing/index.html>. It consists of high resolution images and ground truth images for randomly selected frames. As a quantitative measure the Matthews Correlation (Phi) Coefficient (MCC) [8] is used. It computes the rate between true positive (true foreground), true negative (true background), False Positive (false foreground), and false negative (false background) pixels. The MCC is computed as follows

$$\Phi = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (2)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. When the resulting coefficient is $+1$, the perfect prediction was measured, for -1 the inverse prediction, and for 0 the random prediction. The results can be seen in Table 1. As seen in the Table, our model performs better

Table 1. Performance comparison: RCBS: this model; TGBS: temporal gaussian

	TP	TN	FP	FN	Phi
RCBS	704612	19469974	80437	978641	0.59
TGBS [3]	630959	18605493	154090	1843122	0.41

than TGBS method. Further improvements for dynamic background adaptation, casted shadows removal, and segmentation filtering are presented in following parts of the text.

3.2 Layered RC Model

The simple layered RC model is already able to distinguish between foreground and background, but it still needs to be modified for adaptation to the dynamic backgrounds. Dynamic backgrounds are hard to adapt to. In many cases the background consists of moving objects like tree branches and water surface waves or objects frequently changing its color. The interiors, where the light are switched on and off are also considered as dynamic backgrounds. In this case, using only one Gaussian leads to high values of standard deviation σ . If the σ is high, most of the input values are marked as background, even the moving

objects in the foreground. This problem was solved by C. Stauffer and W.E.L. Grimson [7] in MoGBS. They developed a method, where the background is not modeled only by one Gaussian, but by a mixture of K Gaussians. There, each of the Gaussians adapts to one kind of the backgrounds emerging in the video sequence.

Inspired by this method, we introduce the layered RC circuit model (LRCBS). We added to our circuit (model) additional layers, that are used to represent multiple backgrounds, the *DEMUX* which is a demultiplex connecting the input voltages u_e (input values) to specific layer, which is selected by selector S . Each layer is also connected by the inter-layer resistor R_L providing inter-layer filtration. This helps mainly in initialization step, if the backgrounds are set randomly, and also when one background disappears. Layer, that does not represent any background moves towards to the nearest active background layer, where it helps to represent its background. The modified background modeler scheme can be seen in Figure 3. The adaptation of this model is driven by selector S . When

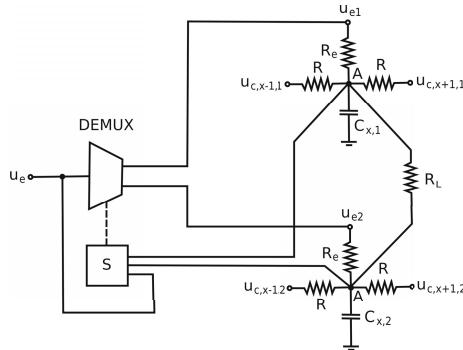


Fig. 3. The scheme of one one-dimensional block representing the layered RC model

the new excitation value u_e emerges, the selector S connects it using the demultiplex *DEMUX* to the layer, where the absolute difference between new value u_e and mean value $u_{c,l}$ is minimal. The other layers excitation values $u_{e,l}$ are leveled to theirs corresponding voltages $u_{c,l}$ (no excitation). This is performed for each block in the model and then the following difference equation is used for the update

$$\begin{aligned}
 u_{c,x,y,l}^{(t+1)} = & u_{c,x,y,l}^{(t)} + \frac{dt}{RC} (u_{c,x-1,y,l}^{(t)} + u_{c,x+1,y,l,t}^{(t)} + u_{c,x,y-1,l}^{(t)} \\
 & + u_{c,x,y+1,l}^{(t)} - 4I_{c,x,y,l}^{(t)}) + \frac{dt}{R_L C} (u_{e,x,y,l}^{(t)} - u_{c,x,y,l}^{(t)}) \\
 & + \frac{dt}{R_L C} (u_{c,x,y,l-1}^{(t)} + u_{c,x,y,l+1}^{(t)} - 2u_{c,x,y,l}^{(t)}) ,
 \end{aligned} \tag{3}$$

where $u_{c,x,y,l}^{(t)}$ is the value representing the background for pixels at position x, y in the layer l at time t . The dt is a time difference constant, and R_L is the resistance of the resistor connecting individual layers. Similarly, the u_σ is modeled. The pixel is marked as foreground, if the input value is not within the distance of 2.5σ from the most similar layer value.

3.3 Selectivity

In some cases, the selectivity is introduced to background subtraction algorithms. It slows or stops the adaptation of the background for the input pixels marked as the foreground. This way, the moving objects do not imprint into the background that much or not at all. The selectivity is driven by the resistance of the resistor R_e in our method. If the resistance R_e is high, then the background adaptation is slow and vice versa. This implies that the resistance of the resistor should be driven by the difference between voltages u_c and u_e . We have found the inspiration in a model of perceptron used in artificial neural network. Namely, the perceptron model used for back propagation neural network developed by David E. Rumelhart et al. [9]. The output of the perceptron y is computed from the total input x using the equation

$$y = \frac{1}{1 + e^{-\lambda(x-t)}}, \quad (4)$$

where λ drives the slope of the function, and t represents the threshold (the point, where the function moves rapidly from zero to one). Shape of the function can be seen in Figure 4. To use this function with our method, we have to

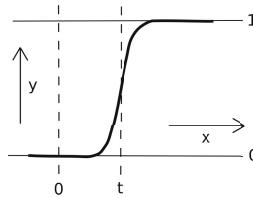


Fig. 4. The perceptron output function, used in back-propagation algorithm

define the maximal resistance R_{max} (slow adaptation of the background) and minimal resistance R_{min} (fast adaptation of the background). The function is then shifted up by R_{min} and stretched by the difference between maximal and minimal resistances. The input value is given by the absolute difference (distance) between the excitation voltage u_e and the condenser voltage u_c , and the threshold is given by modeled standard deviation $u_{c\sigma}$ multiplied by T , which is often set to 2.5 (represents 99 % of possible background values). This gives us

a function with fluent transition between minimal resistance for values within the range $Tu_{c\sigma}$ and maximal resistance for values exceeding this range. This equation has the following form:

$$R_e = R_{min} + \frac{R_{max} - R_{min}}{1 + e^{-\lambda(\|u_e - u_c\| - Tu_{c\sigma})}} , \quad (5)$$

and the graph of the function can be seen in Figure 5.

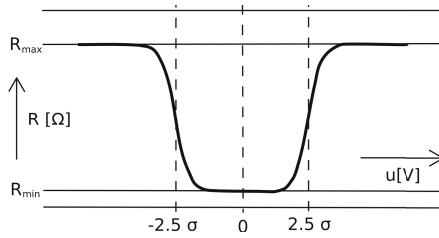


Fig. 5. Graph of a function for resistance regulation

3.4 Casted Shadows

Another problem is casted shadow. Objects moving in the video sequences often cast shadows and these are failingly marked as part of the object, which is not a wanted effect. In our method, the background/foreground segmented images are further processed by algorithm proposed by Thanarat Horprasert et al. [10]. They separate the brightness from chromaticity, and compute the brightness and chromaticity distortion using input pixel values and background model values. Those values are then thresholded and if those are within selected threshold, then the pixel is marked as shadow.

3.5 Segmentation Filtering

The background subtraction segmentations are often post processed by morphological operators. We have decided to use a more sophisticated filter. Our filter computes local histogram of segmentation affiliations for each pixel in segmentation and the affiliation for current pixel is substituted by the affiliation of the most frequent affiliation value (most probable value). The size of the area around the pixel directly affects the strength of the filter. The greater the area is, the stronger the filter is, and the more details are ignored. This filter can be used not only for binary segmentations, but also for ternary segmentations such as segmentation of moving objects, shadows and background. We have created an artificial example of segmentation to show ours filter abilities. The input and the output of the filter can be seen in Figure 6.

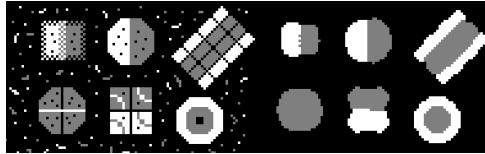


Fig. 6. Artificial segmentation before and after filtering

3.6 Frequency Sensitive Background Modeling

We have encountered one more problem. Our adaptation algorithm is similar to simple competitive learning [11], which in many cases becomes stuck in poor local solution (background). This also rises the problem, where some layers represent only small part of the background or none at all. We need each layer to represent approximately same quantity of the background. This is solved by introduction of a frequency sensitivity (frequency sensitive competitive learning) [12]. The basic idea is to store information about frequency of excitation of each layer in the block. In our method, we monitor the frequency of excitation, and if the pixel is marked as the foreground (new potential background), we excite the layer with the least frequency of the excitation. Consequently, this makes the least used layer to represent a new background.

4 Experiments

In this section we experiment with the complete method with all presented improvements and we compare it to other commonly used methods. The experiments were conducted on real video sequences captured from IP cameras installed on a level crossing, that we use for tests of obstacle detection and obstacle behavior monitoring. Furthermore, we use standard datasets, often used for testing and comparison of background subtraction algorithms. The level crossing dataset and ground truth images can be downloaded from <http://mrl.cs.vsb.cz/people/mozdren/levelcrossing/index.html>, and standard datasets with ground truth images created by L. Li [13] has been downloaded from http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.

The first experiment that we have conducted was a test of the abilities of the proposed method under different conditions like dynamic backgrounds and changing illumination. Resulting segmentations in Figure 7 clearly show that our method is able to adapt under many difficult conditions. The tests were conducted using the following configuration: $R = 2 M\Omega$, $C = 1 \mu F$, $dt = 0.001$, $R_L = 2 M\Omega$, $R_{min} = 20 k\Omega$, $R_{max} = 200 k\Omega$, $\lambda = 5.0$, $K = 5$.

Our second experiment is quantitative. We compared our method with other commonly used methods. In this test, the level crossing dataset has been used. This dataset consists of high resolution images, and therefore, moving object detection can be measured more accurately. As a quantitative measure the previously presented Matthews Correlation (Phi) Coefficient (MCC) [8] is used.

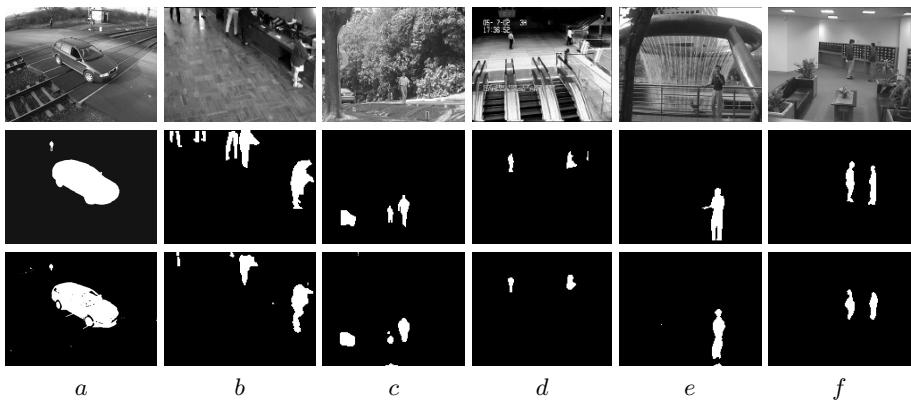


Fig. 7. Testing images their foreground segmentations and ground truths. First row) input images; Second row) ground truth; Third row) Our segmentations, a) Level Crossing - exterior with shadows; b) Bootstrap - high movement, changing illumination; c) Campus - dynamic background, moving tree branches; d) Escalator - dynamic background; e) Fountain - dynamic background, water; f) Lobby - Strong change in illumination, switching lights on and off

Table 2. Algorithms comparison: LRCBS: layered RC; MoGBS: mixture of gaussians; RCBS: Simple RC; TDBS: temporal difference; TGGS: temporal gaussian; TMBS: temporal median

	TP	TN	FP	FN	Phi
LRCBS	681225	20385616	103824	62999	0.89
MoGBS [7]	489913	20329902	295136	118713	0.7
RCBS	704612	19469974	80437	978641	0.59
TDBS [5]	461445	19816336	323604	632279	0.48
TGGS [3]	630959	18605493	154090	1843122	0.41
TMBS [6]	535682	17712377	249367	2736238	0.29

The results can be seen in Table 2. There, you can see that our method performs better than other commonly used methods. The simple RC circuit model (RCBS) performs better, than TDBS, TGGS, TMBS, which are the methods, that are not able to adapt to dynamic background as well as RCBS. The layered version LRCBS is in addition able to deal with dynamic backgrounds. It is clear from the experiments that it outperforms the MoGBS.

5 Conclusion

We have developed a novel algorithm for background subtraction using a layered RC circuit for background modeling. We have shown, that our simple RC model performs better than TGGS method that is similar to. This gave us

good foundations for further improvements. The first improvement was introduction of additional layers, which allowed our method to represent dynamic backgrounds. We also dealt with casted shadows and output segmentation filtering. We have shown in the experiments that our algorithm is able to adapt to many difficult conditions like strong illumination changes, casting shadows, and dynamic backgrounds. Furthermore, our method performs better than other commonly used methods for background subtraction.

Acknowledgement. This work was supported by the SGS in VSB Technical University of Ostrava, Czech Republic, under the grant No. SP2013/185, and Ministry of Industry and Trade of the Czech Republic - project TIP No. FR-TII/027 .

References

1. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting (1995)
2. Papageorgiou, C., Poggio, T.: A trainable system for object detection. *International Journal of Computer Vision* 38, 15–33 (2000)
3. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfnder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 780–785 (1997)
4. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, 629–639 (1990)
5. Hironobu, A.L., Lipton, A.J., Fujiyoshi, H., Patil, R.S.: Moving target classification and tracking from real-time video, pp. 8–14 (1998)
6. Lo, B., Velastin, S.: Automatic congestion detection system for underground platforms. In: *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 158–161 (2001)
7. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, p. xxiii+637+663 (1999)
8. Powers, D.M.W.: Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia (2007)
9. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Cognitive Modeling* 1, 213 (2002)
10. Horprasert, T., Harwood, D., Davis, L.S.: A statistical approach for real-time robust background subtraction and shadow detection, pp. 1–19 (1999)
11. Rumelhart, D.E., Zipser, D.: Feature discovery by competitive learning, vol. 1, ch. 5, pp. 151–193. MIT Press, Cambridge (1986)
12. Ahalt, S.C., Krishnamurthy, A.K., Chen, P., Melton, D.E.: Competitive learning algorithms for vector quantization. *Neural Networks* 3, 277–290 (1990)
13. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing* 13, 1459–1472 (2004)

Pairwise Kernels for Human Interaction Recognition

Saeid Motian, Ke Feng, Harika Bharthavarapu,
Sajid Sharlemin, and Gianfranco Doretto

Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV 26506, USA

{samotian, kfeng, habharthavarapu, sasharlemin}@mix.wvu.edu,
gianfranco.doretto@mail.wvu.edu

Abstract. In this paper we model binary people interactions by forming temporal interaction trajectories, under the form of a time series, coupling together the body motion of each individual as well as their proximity relationships. Such trajectories are modeled with a non-linear dynamical system (NLDS). We develop a framework that entails the use of so-called pairwise kernels, able to compare interaction trajectories in the space of NLDS. To do so we address the problem of modeling the Riemannian structure of the trajectory space, and we also prove that kernels have to satisfy certain symmetry properties, which are peculiar of this interaction modeling framework. Experiment results show that this approach is quite promising, as it is able to match and improve state-of-the-art classification and retrieval accuracies on two human interaction datasets.

1 Introduction

Recognizing human interactions from video is an important step forward towards the long-term goal of performing scene understanding fully automatically. Recent years have seen a concentration of works revolving around the problem of recognizing single-person actions, as well as group activities (see [1] and references therein). On the other hand, the area of modeling the interactions between two people is still relatively unexplored. Only recently, more realistic interaction datasets [2,3] have become available, and triggered the development of more sophisticated approaches [4,5,6,7].

In this paper we aim at developing a modeling framework leading to an approach that is fast, and that could become a building block for analyzing the behavior of a larger crowd in a scene, monitored by a network of cameras. We make the assumption that people in the scene are been tracked. This allows to analyze the spatiotemporal volume around each person and to extract relevant motion features. At the same time, the tracking information of a pair of individuals enables the extraction of a set of proxemics cues, which coupled with the motion cues form *interaction trajectories*.

We model interaction trajectories as the output of non-linear dynamical systems (NLDS), and reduce the problem of recognizing human interactions to the problem of discriminating between NLDSs. This requires designing special kernels that satisfy certain properties. In particular, (a) they have to take into account the geometry of the space where the interaction trajectories are defined, and (b) they have to satisfy certain symmetry properties, which are induced by the fact that we are modeling people interactions. We address (a) and (b) by carefully exploiting kernel construction techniques,

and by clearly showing that kernels for recognizing interaction trajectories should belong to a subcategory of the so-called *pairwise kernels*, and in particular they should satisfy the *balanced* property. A positive side effect of this framework is that by using pairwise symmetric and balanced kernels not only one can boost performance, but also is possible to significantly reduce the training time, since there is no need to use a symmetric training dataset, which has double the size of a regular one.

In § 2 we describe how human interactions can be represented by interaction trajectories, and introduce a new efficient motion feature, called motion histogram. In § 3 we pose the human interaction recognition problem and identify the challenges it implies. In § 4 we explain how interaction trajectories are represented by NLDSs. In § 5 we explain how to design kernels for comparing interaction trajectories, while addressing the challenges outlined in § 3. § 6 shows classification, and retrieval experiments where several proposed kernels are tested, validating the framework from the theoretical perspective, as well as practical by achieving very promising results.

2 Representation of Human Interactions

Given a sequence of images $\{I_t\}_{t=1}^T$, depicting two, or more people, we are interested in defining a representation for describing a potential interaction between two individuals. At every frame the bounding box delimiting the region of each person is assumed to be given (e.g., through the use of a person tracker [8], as it is typically done in video surveillance settings). For the i -th person in the video sequence, at every time t the bounding box is used to extract features aiming at describing the body motion.

From each bounding box two features are computed. The first one is the *histogram of oriented optical flow* (HOOF) [9], $\mathbf{h}_{i,t}$. It captures the motion between two consecutive frames. In addition, we introduce a feature called *motion histogram* (MH), which summarizes the motion trajectory of the past $\tau - 1$ frames (where $\tau > 1$). It requires the computation of the *motion*, or *frequency image* [10], $M_t \doteq \sum_{k=1}^{\tau-1} \eta(I_t - I_{t-k})$, where $\eta(z) = 1$ if $|z| > \delta$, otherwise $\eta(z) = 0$. Here δ is a threshold parameter to be set. Therefore, the motion histogram of person i at frame t , $\mathbf{m}_{i,t}$, is computed by binning the motion image inside the bounding box of the person. Both histograms are scale and direction invariant, as well as fairly robust to background noise, besides being fast to compute. Figure 1 shows a couple of examples of motion images with the corresponding MH features.

Eventually, the i -th person is represented by the sequence of HOOF and MH features $\mathbf{h}_i \doteq \{\mathbf{h}_{i,t}\}_{t=1}^T$, and $\mathbf{m}_i \doteq \{\mathbf{m}_{i,t}\}_{t=1}^T$, respectively, where $\mathbf{h}_{i,t}$ and $\mathbf{m}_{i,t}$ are normalized histograms made of b bins, $\mathbf{h}_{i,t} \doteq [h_{i,t;1}, \dots, h_{i,t;b}]^\top$, and made of τ bins, $\mathbf{m}_{i,t} \doteq [m_{i,t;0}, m_{i,t;1}, \dots, m_{i,t;\tau-1}]^\top$, where bin 0 has been added to account for the case of absence of motion.

In order to analyze the interaction between person i and person j , proxemics cues play an important discriminative role (e.g., person i cannot be hugging person j if they are far enough apart). That information here is captured by the Euclidean distance between the position $\mathbf{p}_{i,t}$ of person i , and the position $\mathbf{p}_{j,t}$ of person j , given by $d_{ij,t} \doteq \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2$. When the camera calibration is known and people tracking is performed on the ground-plane, the person position and velocity are readily available.

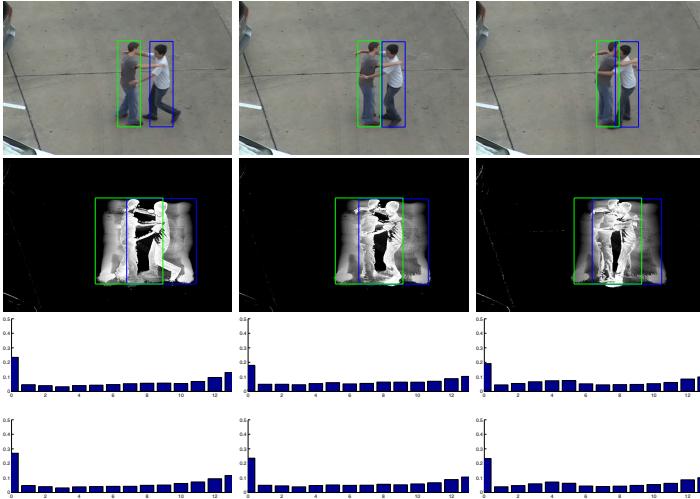


Fig. 1. From top to bottom: frames, motion images, and corresponding motion histograms of the green (top) and blue (bottom) boxes, for the UT-Interaction dataset [2].

If this is not the case, one can characterize proximity by computing the distance in the image domain, and performing a normalization based on the people size. Even if doing so is not view invariant, § 6 shows that this information still significantly increases the classification accuracy for the tested datasets. Other important cues include relative velocity and gaze direction between person i and j . We defer the use of those to future work.

Given the motion, described by $(\mathbf{h}_i, \mathbf{m}_i)$ and $(\mathbf{h}_j, \mathbf{m}_j)$, of person i and j , and their proximity described by $d_{ij} \doteq \{d_{ij,t}\}$, their *interaction trajectory* is the temporal sequence $\mathbf{y}_{ij} \doteq \{\mathbf{y}_{ij,t}\}_{t=1}^T$, where $\mathbf{y}_{ij,t} \doteq [\mathbf{h}_{i,t}^\top, \mathbf{m}_{i,t}^\top, \mathbf{h}_{j,t}^\top, \mathbf{m}_{j,t}^\top, d_{ij,t}]^\top$.

3 Recognizing Human Interactions

The interaction trajectory \mathbf{y}_{ij} is a temporal sequence, and can be seen as a section of the realization of a stochastic process, which fully describes the dynamics of an interaction. Therefore, recognizing interactions is cast as a problem of recognizing stochastic processes. Under mild conditions, when $\mathbf{y}_{ij,t}$ is defined on an Euclidean space, it can be modeled as the output of a linear dynamical system (LDS), and several distances and kernels for comparing them have been proposed [11,12].

The problem we set out to address presents a couple of unique challenges. First, $\mathbf{y}_{ij,t}$ does not assume values in an Euclidean space but in a Riemannian manifold with a nontrivial structure, which is $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+$. In particular, \mathbb{H}_b is the space of normalized histograms, which are probability mass functions satisfying the constraints $\sum_{k=1}^b h_{t;k} = 1$, and $h_{t;k} \geq 0, \forall i \in \{1, \dots, b\}$; and similarly, \mathbb{H}_τ is the space of normalized histograms with τ bins.

The second challenge relates to the symmetry of the input feature space, which is peculiar to modeling interactions. In particular, a recognition schema entails the definition of a decision function $f : \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+ \rightarrow \mathbb{R}$, which will predict whether person i and j are engaging in a certain interaction (i.e., $f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) > 0$), or not (i.e., $f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) < 0$). Therefore, given that no person ordering is imposed a priori, the decision function is expected to be symmetric with respect to i and j , i.e.,

$$f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) = f(\mathbf{h}_j, \mathbf{m}_j, \mathbf{h}_i, \mathbf{m}_i, d_{ji}). \quad (1)$$

In § 4 we summarize a general framework for modeling and comparing temporal sequences that do not assume values in an Euclidean space, which is based on kernelizing the output of an LDS, giving rise to a non-linear dynamical system (NLDS) representation. In § 5 we propose a family of so-called *pairwise kernels* that takes into account the Riemannian structure of the input feature space as well as its symmetry.

4 Modeling Temporal Sequences with Kernel NLDSs

A stochastic process can be modeled as the output of a dynamical system. Therefore, one can compare processes by comparing dynamical system models. For second-order stationary processes assuming values in an Euclidean space, the model of choice is an LDS [13], and methods for comparing LDSs include geometric distances [14], algebraic kernels [12], and information theoretic metrics [15]. For stationary processes assuming values in a non-Euclidean space like \mathbb{H}_b , or the space of binary values, suitable extensions have been proposed in [9], and [16], respectively, which are based on the ideas summarized in this section.

Given a temporal sequence $\{\mathbf{y}_t\}_{t=1}^T$, assuming values in a non-Euclidean space \mathcal{S} , let us consider the Mercer kernel $K(\mathbf{y}_t, \mathbf{y}'_t) = \Phi(\mathbf{y}_t)^\top \Phi(\mathbf{y}'_t)$, where $\Phi(\cdot)$ is mapping \mathcal{S} to \mathcal{H} , a Reproducing Kernel Hilbert Space (RKHS) [17]. We assume that $\{\mathbf{y}_t\}$ is mapped to a sequence $\{\Phi(\mathbf{y}_t)\}$, which can be modeled as the output of an LDS, given by

$$\begin{cases} \mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{v}_t, \\ \Phi(\mathbf{y}_t) = C\mathbf{x}_t + \mathbf{w}_t. \end{cases} \quad (2)$$

Here $\mathbf{x}_t \in \mathbb{R}^n$ is the state of the LDS at time t , $A \in \mathbb{R}^{n \times n}$ describes the dynamics of the state evolution, and the system noise \mathbf{v}_t is zero-mean i.i.d. Gaussian distributed with appropriate covariance. In addition, (2) differs from a traditional LDS in that C may not be a matrix but a linear operator $C : \mathbb{R}^n \rightarrow \mathcal{H}$ to account that \mathcal{H} could be an infinite dimensional space. The observation noise \mathbf{w}_t is also modeled as a zero-mean i.i.d. Gaussian process with appropriate dimension and covariance, which is independent from \mathbf{v}_t .

It is possible to extend the procedure developed in [18] for estimating the parameters of LDSs to the case of NLDS models like (2). This is done by substituting the PCA applied to the temporal sequence with a Kernel PCA (KPCA) [17], as it is shown in [19]. In particular, given the sequence $\{\mathbf{y}_t\}$ and the kernel K , [19] shows how to estimate the matrix A , and a representation of the linear operator C , under the form of kernel principal components. The c -th component is defined by the map $\Phi(\cdot)$, and by the

KPCA weight vector $\alpha_c \doteq v_c / \sqrt{\lambda_c}$, where λ_c and v_c are the c -th largest eigenvalue and eigenvector of the kernel matrix between the zero-mean data in the high-dimensional space, computed as $(I - \frac{1}{T}\mathbf{e}\mathbf{e}^\top)K(I - \frac{1}{T}\mathbf{e}\mathbf{e}^\top)$, where $\mathbf{e} = [1, \dots, 1]^\top \in \mathbb{R}^T$, and $[K]_{st} = K(\mathbf{y}_s, \mathbf{y}_t)$. It turns out that this representation of C is enough to define kernels for comparing NLDSSs of the type in (2).

Recently, a family of Binet-Cauchy kernels for LDSs has been introduced in [12], and in [9] it has been extended for NLDSSs like (2). In particular, the Binet-Cauchy trace kernel for NLDS is the expected value of an infinite series of weighted inner products between the outputs after embedding them into the high-dimensional (possibly infinite) space using the map $\Phi(\cdot)$. More precisely

$$K_{NLDS}(\{\mathbf{y}_t\}_{t=1}^\infty, \{\mathbf{y}'_t\}_{t=1}^\infty) \doteq E \left[\sum_{t=1}^{\infty} \lambda^t \Phi(\mathbf{y}_t)^\top \Phi(\mathbf{y}'_t) \right] = E \left[\sum_{t=1}^{\infty} \lambda^t K(\mathbf{y}_t, \mathbf{y}'_t) \right], \quad (3)$$

where $0 < \lambda < 1$, and the expectation of the infinite sum of the inner products is taken w.r.t. the joint probability distribution of \mathbf{v}_t and \mathbf{w}_t . The kernel (3) can be computed in closed form, and it requires the computation of the infinite sum $P = \sum_{t=1}^{\infty} \lambda^t (A^T)^\top F A'^\top$, where $F = \tilde{\alpha} S \tilde{\alpha}'$, and the columns of $\tilde{\alpha}$ and $\tilde{\alpha}'$ are the centered KPCA weight vectors of $\{\mathbf{y}_t\}$ and $\{\mathbf{y}'_t\}$, given by $\tilde{\alpha}_c = \alpha_c - \frac{\mathbf{e}^\top \alpha_c}{T} \mathbf{e}$, and $\tilde{\alpha}'_d = \alpha'_d - \frac{\mathbf{e}^\top \alpha'_d}{T'} \mathbf{e}$, respectively. S instead is such that $[S]_{st} = K(\mathbf{y}_s, \mathbf{y}'_t)$, where $s \in \{1, \dots, T\}$, and $t \in \{1, \dots, T'\}$. If $\lambda \|A\| \|A'\| < 1$, where $\|\cdot\|$ is a matrix norm, then P can be computed by solving the corresponding Sylvester equation $P = \lambda A^\top P A' + F$.

Given P , kernel (3) can be computed in closed form provided that the covariances of the system noise, the observation noise, and the initial state are available. On the other hand, as [9] points out, for recognition of phenomena that are assumed to be made by one or multiple cycles of a temporal sequence, we want to use a kernel that is independent from the initial state and the noise processes. Therefore, the original kernel (3) is simplified to K_{NLDS}^σ , which is a kernel only on the dynamics of the NLDS, and is given by the maximum singular value of P , i.e.,

$$K_{NLDS}^\sigma = \max \sigma(P). \quad (4)$$

For more details about the estimation of the NLDS model parameters, and about the derivation of kernel (4) the reader is referred to [18,19,9].

5 Pairwise Kernels for Recognizing Interaction Trajectories

In this section we intend to use the framework described in § 4 to model, compare, classify, and rank interaction trajectories. § 3 has pointed out that trajectories live in a non-Euclidean space with a special symmetry. At the same time, the effectiveness of modeling them with (2) depends upon how well the kernel $K(\mathbf{y}_{ij,t}, \mathbf{y}'_{ij,t})$ is able to “map” the non-Euclidean input feature space \mathcal{S} to a RKHS \mathcal{H} . Therefore, here we propose a few strategies for designing the kernel K .

Table 1. Classification accuracy for the UT-Interaction dataset [2]. For Set 1 MH features are computed with $\tau = 14$, and $\delta = 2$; HOOF features are computed with $b = 18$; NLDS order is set to $n = 8$. For Set 2 MH features are computed with $\tau = 22$, and $\delta = 5$; HOOF features are computed with $b = 24$; NLDS order is set to $n = 10$.

CLASSIFICATION ACCURACY - SET 1							CLASSIFICATION ACCURACY - SET 2						
Kernel/Class	Hug	Kick	Push	Punch	Hand Shake	Avg	Kernel/Class	Hug	Kick	Push	Punch	Hand Shake	Avg
No Proximity													
k_S	75.00	75.00	46.15	33.33	75.00	60.65	k_S	72.72	36.36	37.5	16.66	87.5	51.51
$K_H^{TL}(k_S)$	83.33	75.00	61.53	41.66	91.66	70.49	$K_H^{TL}(k_S)$	54.54	54.54	62.5	16.66	87.5	57.57
$K_H^{DS}(k_S)$	83.33	75.00	38.46	33.33	91.66	63.93	$K_H^{DS}(k_S)$	54.54	54.54	25.00	8.33	93.75	48.48
$K_H^{TL}(k_h k_m)$	83.33	83.33	84.61	8.33	100	70.49	$K_H^{TL}(k_h k_m)$	72.72	63.63	50.00	50.00	62.50	59.09
$K_H^{DS}(k_h k_m)$	83.33	75.00	38.46	33.33	91.66	63.93	$K_H^{DS}(k_h k_m)$	45.45	27.27	43.75	16.16	87.50	46.96
With Proximity													
RBF	100	100	76.92	50.00	83.33	81.96	RBF	100	45.45	87.50	41.66	81.25	72.72
$k_S k_d$	83.33	83.33	61.53	41.66	83.33	70.49	$k_S k_d$	100	54.54	81.25	41.66	83.33	72.72
$K_H^{TL}(k_S) k_d$	91.66	83.33	76.92	91.66	100	88.52	$K_H^{TL}(k_S) k_d$	81.81	72.72	50.00	16.16	75.00	59.09
$K_H^{DS}(k_S) k_d$	100	83.33	69.23	50	91.66	78.68	$K_H^{DS}(k_S) k_d$	90.90	27.27	50.00	33.33	93.75	60.60
$K_H^{TL}(k_h k_m) k_d$	100	100	69.23	91.66	100	91.80	$K_H^{TL}(k_h k_m) k_d$	100	72.72	87.50	75.00	100	87.87
$K_H^{DS}(k_h k_m) k_d$	100	83.33	69.23	66.66	91.66	81.96	$K_H^{DS}(k_h k_m) k_d$	100	36.36	87.50	41.66	100	75.75

Since the input feature space $\mathcal{S} \doteq \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+$ is a non-Euclidean space which is a Riemannian manifold, defining K to be a linear kernel would clearly be sub-optimal. A typical approach for improving the map to a RKHS is to use a generic, top-performing non-linear kernel, such as the Gaussian radial basis function (RBF) kernel with Euclidean distance. However, in this way we do not take advantage of the known Riemannian structure of \mathcal{S} . One way to do so is to replace the Euclidean distance with a proper distance for the manifold \mathcal{S} . Unfortunately, to the authors' knowledge defining a distance on \mathcal{S} is still an open problem, although for \mathbb{H}_b (or \mathbb{H}_τ) alone a theoretical solution exists, which is the *Fisher-Rao* metric [20]. Therefore, whenever it cannot be done otherwise, we advocate the use of kernel construction techniques [17], which take into account the fact that \mathcal{S} is given by the Cartesian product of subspaces. This way allows to concentrate on each subspace separately, and exploit the known subspace geometry to the full extent.

To start, we notice that the input feature space \mathcal{S} is given by the Cartesian product of the subspaces $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau$, and \mathbb{R}_+ . Therefore, we can focus on designing a kernel for histograms K_H on the first subspace, and a kernel K_d for people distances on the second subspace. K_H and K_d can then be combined by computing their *tensor product kernel* [17], leading to

$$K \doteq (K_H \otimes K_d)(\mathbf{y}_{ij}, \mathbf{y}'_{ij}) = K_H((\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j), (\mathbf{h}'_i, \mathbf{m}'_i, \mathbf{h}'_j, \mathbf{m}'_j)) K_d(d_{ij}, d'_{ij}), \quad (5)$$

where we have dropped the time subscript t to lighten the notation. Intuitively, a kernel defines similarity in an input space. Kernel (5) yields a high value only if the instances in each subspace have high similarity with the corresponding instances in the same subspace. This is desirable because the classification of interactions should be based on the similarity across not only the motion features, but also the proximity cues, as it is explained in § 2.

Table 2. Classification accuracy, and video retrieval average precision for the TVHI dataset [3]. MH features are computed with $\tau = 5$, and $\delta = 3$; HOOF features are computed with $b = 10$; NLDS order is set to $n = 10$.

CLASSIFICATION ACCURACY							RETRIEVAL PRECISION					
Kernel/Class	HS	HF	HG	KS	NG	AVG	Kernel/Class	HS	HF	HG	KS	AVG
No Proximity							No Proximity					
k_S	33.33	51.72	36.36	30.43	52.00	40.77	k_S	0.239	0.316	0.293	0.402	0.314
$K_H^{TL}(k_S)$	19.05	58.62	18.18	47.83	60.00	40.74	$K_H^{TL}(k_S)$	0.208	0.335	0.265	0.498	0.330
$K_H^{DS}(k_S)$	0	62.07	31.82	30.43	88.00	42.46	$K_H^{DS}(k_S)$	0.199	0.300	0.267	0.424	0.300
$K_H^{TL}(k_h k_m)$	38.10	44.83	31.82	30.43	44.00	37.84	$K_H^{TL}(k_h k_m)$	0.267	0.264	0.277	0.523	0.330
$K_H^{DS}(k_h k_m)$	9.52	41.38	31.82	43.48	68.00	38.84	$K_H^{DS}(k_h k_m)$	0.222	0.296	0.263	0.422	0.302
With Proximity							With Proximity					
RBF	4.76	65.52	59.09	73.91	60.00	52.66	$k_S k_d$	0.310	0.319	0.559	0.541	0.427
$k_S k_d$	19.05	62.07	86.36	73.91	56.00	59.48	$K_H^{TL}(k_S) k_d$	0.334	0.333	0.485	0.482	0.412
$K_H^{TL}(k_S) k_d$	19.05	79.31	81.82	65.22	64.00	61.88	$K_H^{DS}(k_S) k_d$	0.339	0.351	0.538	0.483	0.424
$K_H^{DS}(k_S) k_d$	23.81	51.72	90.91	78.26	64.00	61.74	$K_H^{TL}(k_h k_m) k_d$	0.342	0.357	0.551	0.525	0.439
$K_H^{TL}(k_h k_m) k_d$	28.57	79.31	86.36	65.22	64.00	64.69	$K_H^{DS}(k_h k_m) k_d$	0.351	0.338	0.554	0.540	0.440
$K_H^{DS}(k_h k_m) k_d$	38.10	51.72	81.82	73.91	72.00	63.51						

For kernel K_d we observe that d_{ij} belongs to \mathbb{R}_+ , and therefore we simply chose a Gaussian RBF kernel, given by

$$K_d(d_{ij}, d'_{ij}) \doteq \exp(-\gamma|d_{ij} - d'_{ij}|^2). \quad (6)$$

For kernel K_H , we note that it is a so-called *pairwise kernel* [21], because it is such that $K_H : (\mathcal{X}_H \times \mathcal{X}_H) \times (\mathcal{X}_H \times \mathcal{X}_H) \rightarrow \mathbb{R}$, where $\mathcal{X}_H \doteq \mathbb{H}_b \times \mathbb{H}_\tau$, and it could be used to support *pairwise classification*, which aims at deciding whether the examples of a pair $(a, b) \in \mathcal{X}_H \times \mathcal{X}_H$ belong to the same class or not. The requirement of being positive semidefinite implies that K_H satisfies the following *symmetry* property

$$K_H((a, b), (a', b')) = K_H((a', b'), (a, b)), \quad (7)$$

for all $a, b, a', b' \in \mathcal{X}_H$. By using kernel construction techniques based on direct sum and tensor product of kernels, given the kernel $k_H : \mathcal{X}_H \times \mathcal{X}_H \rightarrow \mathbb{R}$, one can build the following pairwise versions of K_H

$$K_H^D = (k_H \oplus k_H)(a, b, a', b') = k_H(a, a') + k_H(b, b'), \quad (8)$$

$$K_H^T = (k_H \otimes k_H)(a, b, a', b') = k_H(a, a')k_H(b, b'), \quad (9)$$

which obviously satisfy the symmetric property.

We now verify whether by using the kernels defined in (8) and (9) it is possible to construct decision functions for interaction trajectories, which are supposed to satisfy the symmetry property (1). We plan to learn decision functions f with a SVM that exploits the general kernel (3). Therefore, they will assume the form

$$f(\{a_{i,t}, a_{j,t}, d_{ij,t}\}) \doteq \sum_{u,v} \alpha_{uv} \ell_{uv} K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) + \beta, \quad (10)$$

where α_{uv} , ℓ_{uv} , and β are the usual SVM parameters [17], and $a_{i,t} = (\mathbf{h}_{i,t}, \mathbf{m}_{i,t}) \in \mathcal{X}_H$, and $a_{j,t} = (\mathbf{h}_{j,t}, \mathbf{m}_{j,t}) \in \mathcal{X}_H$. More importantly, (10) tells us that the symmetry property (1) imposes that

$$\begin{aligned} K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) = \\ K_{NLDS}(\{a_{j,t}, a_{i,t}, d_{ji,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}), \end{aligned} \quad (11)$$

for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$, and $d_{ij,t}, d'_{uv,t} \in \mathbb{R}_+$. In turn, (11) induces a symmetry property on the kernel (5) through (3), which is given by

$$K((a_{i,t}, a_{j,t}, d_{ij,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) = K((a_{j,t}, a_{i,t}, d_{ji,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) , \quad (12)$$

and finally, since $d_{ij,t} = d_{ji,t}$ and $d_{uv,t} = d_{vu,t}$, (12) imposes on K_H the following relationship

$$K_H((a_{i,t}, a_{j,t}), (a'_{u,t}, a'_{v,t})) = K_H((a_{j,t}, a_{i,t}), (a'_{u,t}, a'_{v,t})) , \quad (13)$$

to be valid for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$. Note that the relationship (13) is different than the symmetry relationship (7), and kernels that satisfy (13) are called *balanced* [21].

Unfortunately, the pairwise kernels K_H^D , and K_H^T , defined in (8) and (9), are symmetric but not balanced. Therefore, we propose to test two kernels that have been proved to have good theoretical properties [21], in that they guarantee minimal loss of information, and can be thought of as the balanced versions of K_H^D , and K_H^T . They are defined as follows

$$K_H^{DS}((a, b), (a', b')) = K_H^{SD}((a, b), (a', b')) + K_H^{ML}((a, b), (a', b')) , \quad (14)$$

$$K_H^{TL}((a, b), (a', b')) = \frac{1}{2}(k_H(a, a')k_H(b, b') + k_H(a, b')k_H(b, a')) , \quad (15)$$

where

$$K_H^{SD}((a, b), (a', b')) = \frac{1}{2}(k_H(a, a') + k_H(a, b') + k_H(b, a') + k_H(b, b')) , \quad (16)$$

$$K_H^{ML}((a, b), (a', b')) = \frac{1}{4}(k_H(a, a') - k_H(a, b') - k_H(b, a') + k_H(b, b'))^2 . \quad (17)$$

In particular, K_H^{TL} is called *tensor learning pairwise kernel* [22], whereas K_H^{DS} is called *direct sum pairwise kernel* [21].

Finally, we are left with the task of designing k_H , which is defined on the space $(\mathbb{H}_b \times \mathbb{H}_\tau) \times (\mathbb{H}_b \times \mathbb{H}_\tau)$. Since it is not required to be balanced, and both features, $\mathbf{h}_{i,t}$ and $\mathbf{m}_{i,t}$, should concur at the same time towards establishing similarity, we apply the tensor product rule to further decompose k_H into two kernels, $k_h : \mathbb{H}_b \times \mathbb{H}_b \rightarrow \mathbb{R}$ and $k_m : \mathbb{H}_\tau \times \mathbb{H}_\tau \rightarrow \mathbb{R}$, producing

$$k_H((\mathbf{h}_{i,t}, \mathbf{m}_{i,t}), (\mathbf{h}'_{i,t}, \mathbf{m}'_{i,t})) = k_h(\mathbf{h}_{i,t}, \mathbf{h}'_{i,t})k_m(\mathbf{m}_{i,t}, \mathbf{m}'_{i,t}) . \quad (18)$$

Both k_h and k_m are kernels for comparing histograms. There are several options for kernels in this domain, as it is outlined in [9], where it has been shown that an excellent

	Hug	Kick	Push	Punch	Shake		Hug	Kick	Push	Punch	Shake
Hug	12	0	0	0	0		11	0	0	0	0
Kick	0	12	0	0	0		1	8	0	1	1
Push	1	2	9	1	0		1	0	14	1	0
Punch	0	0	1	11	0		1	0	1	9	1
Shake	0	0	0	0	12		0	0	0	0	16

Fig. 2. Confusion matrices for the UT-Interaction dataset: Set 1 (left), and Set 2 (right).

compromise between performance and speed is given by the following Mercer kernel

$$k_S(\mathbf{h}_1, \mathbf{h}_2) = \sum_{k=1}^b \sqrt{h_{1,k} h_{2,k}}, \quad (19)$$

which is derived by taking into account that \mathbb{H}_b is diffeomorphic to a subset of the hypersphere \mathbb{S}^{b-1} . We refer to this as the *geodesic kernel*. Both k_h and k_m are picked to be geodesic kernels for histograms with b and τ bins, respectively.

6 Experiments

We have tested our approach on two state-of-the-art human interaction datasets: the UT-Interaction dataset [2], and the TV Human Interaction (TVHI) dataset [3]. The first one contains videos of six interaction classes: *hand shake*, *hug*, *kick*, *point*, *punch*, and *push*. We have excluded the *point* class because it is representative of a single person action. The dataset is divided into Set 1 and Set 2, each consisting of 10 videos. Set 1 videos have mostly a static background, and Set 2 videos have some background motion, with some small camera motion, which makes Set 2 slightly more challenging than Set 1. In our model we assume to have people tracking information, and since the ground-truth annotation of the dataset was not providing that, we annotated the dataset with the VATIC tool [23]. The top-row of Figure 1 shows the bounding boxes obtained with this process. Also, the second row shows the same boxes with a width that is three times of the original. Those wider boxes were used to compute the MH and the HOOF features. In particular, the motion images are computed with respect to the L channel of the Lab color space, and the HOOF features are based on the optical flow computed with the OpenCV library.

The TVHI dataset has videos from 5 different classes: *hand-shakes*, *high-fives*, *hugs*, *kisses*, and *negative* examples. The length of the videos range from 30 to 600 frames. There is a great degree of variation among the videos as they are compiled from different TV shows, which makes this dataset very challenging. As people tracking information

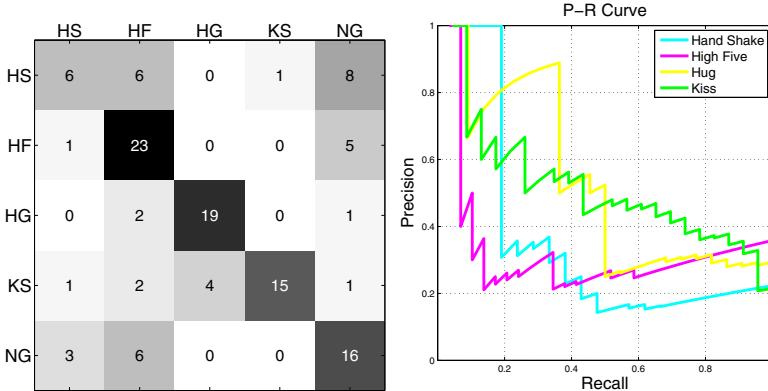


Fig. 3. Confusion matrix (left), and per-class precision-recall curves (right) for the TVHI dataset.

we were able to use the ground-truth annotations made available along with the videos, consisting of bounding boxes framing the upper bodies of all the actors in the scene. Our analysis was limited to the bounding boxes corresponding to the people interacting, and the features were extracted from boxes having a width that was double the original annotations, in order to analyze the motion in a region surrounding each person. Note that, similarly to [3], some of the original videos were not considered due to their very limited length, or due to sharp view point changes during the interaction.

We tested the kernels proposed in § 5 for classification. Different choices of K_H are evaluated, where for each of them we consider the case of interaction trajectories with or without proximity cues. Presence or absence of this information is well marked on the tables, and also on the table kernel labels, by the presence or absence of the k_d kernel (6). Since the input features $(\mathbf{h}_{i,t}, \mathbf{m}_{i,t}, \mathbf{h}_{j,t}, \mathbf{m}_{j,t})$ live in a subspace of $\mathbb{H}_{2b+2\tau}$, it is possible to test the following choices for K_H : (a) k_S , which is the geodesic kernel (19); (b) K_H^{TL} (15), where k_H is a geodesic kernel, indicated with $K_H^{TL}(k_S)$; (c) K_H^{DS} (14), where k_H is a geodesic kernel, indicated with $K_H^{DS}(k_S)$; (d) K_H^{TL} (15), where k_H is the tensor product kernel (18), indicated with $K_H^{TL}(k_h k_m)$; (e) K_H^{DS} (14), where k_H is the tensor product kernel (18), indicated with $K_H^{DS}(k_h k_m)$. Finally, for kernel K (5) we also tested a Gaussian RBF kernel with Euclidean distance.

The kernels described above were used, in conjunction with kernel (4), to train the multi-class classifier of the libSVM [24] with leave-one-out cross-validation. Table 1 shows the classification accuracy for the UT-Interaction dataset, whereas Table 2 shows the classification results for the TVHI dataset. From them we can draw a number of considerations. First, as pointed out in § 5 using an RBF kernel with Euclidean distance leads to suboptimal results. Second, we have experienced a higher degree of good performance consistency for the tensor learning pairwise kernel $K_H^{TL}(k_h k_m)$, versus the direct sum pairwise kernel $K_H^{DS}(k_h k_m)$. Third, we have verified the importance of designing kernels by taking into account the structure of the input feature space in the way that different kernels rank in terms of performance. Fourth, we have verified the importance in incorporating proximity information for discriminating between

Table 3. Classification accuracy for the UT-Interaction dataset obtained using proximity and different motion features, including only motion histograms (MH), only HOOF features, and both. Motion features are computed as indicated in Table 1.

SET 1				SET 2			
Kernel/Feature	MH	HOOF	Both	Kernel/Feature	MH	HOOF	Both
k_{Sk_d}	65.57	68.85	70.49	k_{Sk_d}	60.60	56.06	72.72
$K_H^{TL}(k_S)k_d$	68.85	73.77	88.52	$K_H^{TL}(k_S)k_d$	50.00	54.55	59.90
$K_H^{DS}(k_S)k_d$	70.49	70.49	78.68	$K_H^{DS}(k_S)k_d$	53.03	54.55	60.60
$K_H^{TL}(k_h k_m)k_d$	-	-	91.80	$K_H^{TL}(k_h k_m)k_d$	-	-	87.87
$K_H^{DS}(k_h k_m)k_d$	-	-	81.96	$K_H^{DS}(k_h k_m)k_d$	-	-	75.75

interactions. Fifth, the best classification accuracy is on-par or better than recently reported results [3,4,5,6,7], indicating that the approach is promising. Figure 2 and Figure 3 show the confusion matrices corresponding to the best classification accuracy.

For the TVHI dataset we have also performed a video retrieval experiment. In particular, we have converted the proposed kernels in pairwise distances, where the kernel K_{NLDS} is normalized to 1 when $\{\mathbf{y}_t\} = \{\mathbf{y}'_t\}$, by computing $\tilde{K}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) \doteq K_{NLDS}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) / \sqrt{K_{NLDS}(\{\mathbf{y}_t\}, \{\mathbf{y}_t\}) K_{NLDS}(\{\mathbf{y}'_t\}, \{\mathbf{y}'_t\})}$, and the distance between two interaction trajectories becomes $d(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) \doteq 2(1 - \tilde{K}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}))$. Table 2 and Figure 3 show the retrieval precision and the per-class precision-recall curves, as defined in [25]. It can be seen that even with such a simple approach, the results are comparable to the ones in [3]. We expect that by using the proposed kernels in a “learning to rank” approach [26], the retrieval precision would undergo a substantial increase.

Finally, for the various kernels Table 3 shows how classification performance is affected in three cases, namely when only the MH features are used, only the HOOF features are used, and when both are used. It can be seen that the proposed motion histogram features are capturing valuable motion history information, which is as discriminative as the one captured by the HOOF. Also, it is uncorrelated to the information captured by the HOOF, given the significant boost in classification accuracy.

Acknowledgments. This work was supported in part by grant 2010-DD-BX-0161, awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. ACM Comput. Surv. 43, 16 (2011)
2. Ryoo, M., Aggarwal, J.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV, pp. 1593–1600 (2009)
3. Patron-Perez, A., Marszalek, M., Reid, I., Zisserman, A.: Structured learning of human interactions in TV shows. IEEE TPAMI 34, 2441–2453 (2012)

4. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV, pp. 778–785 (2011)
5. Gaur, U., Zhu, Y., Song, B., Roy-Chowdhury, A.: A “string of feature graphs” model for recognition of complex activities in natural videos. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2595–2602 (2011)
6. Kong, Y., Jia, Y., Fu, Y.: Learning human interaction by interactive phrases. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 300–313. Springer, Heidelberg (2012)
7. Yu, G., Yuan, J., Liu, Z.: Propagative hough voting for human activity recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 693–706. Springer, Heidelberg (2012)
8. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR, pp. 1–8 (2008)
9. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: CVPR, pp. 1932–1939 (2009)
10. Doretto, G., Sebastian, T., Tu, P., Rittscher, J.: Appearance-based person reidentification in camera networks: problem overview and current approaches. Journal of Ambient Intelligence and Humanized Computing 2, 127–151 (2011)
11. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR, Kauai, Hawaii, USA, vol. 2, pp. 58–63 (2001)
12. Vishwanathan, S., Smola, A., Vidal, R.: Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. IJCV 73, 95–119 (2007)
13. Ljung, L.: System identification: theory for the user, 2nd edn. Prentice-Hall, Inc. (1999)
14. De Cock, K., De Moor, B.: Subspace angles and distances between ARMA models. In: MTNS (2000)
15. Chan, A., Vasconcelos, N.: Probabilistic kernels for the classification of auto-regressive visual processes. In: CVPR, vol. 1, pp. 846–851 (2005)
16. Li, W., Vasconcelos, N.: Recognizing activities by attribute dynamics. In: NIPS (2012)
17. Schölkopf, B., Smola, A.: Learning with kernels: SVM, regularization, optimization, and beyond. The MIT Press (2002)
18. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. IJCV 51, 91–109 (2003)
19. Chan, A.B., Vasconcelos, N.: Classifying video with kernel dynamic textures. In: CVPR, pp. 1–6 (2007)
20. Srivastava, A., Jermyn, I., Joshi, S.: Riemannian analysis of probability density functions with applications in vision. In: CVPR, pp. 1–8 (2007)
21. Brunner, C., Fischer, A., Luig, K., Thies, T.: Pairwise support vector machines and their application to large scale problems. JMLR 13, 2279–2292 (2012)
22. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics 21(suppl. 1), i38–i46 (2005)
23. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. IJCV, 1–21 (2012), doi:10.1007/s11263-012-0564-1
24. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
25. Buettcher, S., Clarke, C.L.A., Cormack, G.V.: Information Retrieval. The MIT Press (2010)
26. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: ICML, pp. 129–136 (2007)

A Vision-Based Algorithm for Parking Lot Utilization Evaluation Using Conditional Random Fields

Tomas Fabian

VSB-Technical University of Ostrava,
Department of Computer Science, FEECS,
17. listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic
[tomas.fabian@v sb.cz](mailto:tomas.fabian@vsb.cz)

Abstract. In this paper, we present an algorithm for estimating the occupancy of individual parking spaces. Our method is based on a computer analysis of images obtained by a camera system monitoring the activities on a parking lot. The proposed method extensively uses a priori information about the parking lot layout and the general shape of well-parked cars, which is incorporated in a simplified probabilistic car model. Discriminative features are extracted from a normalized image of every parking space, the relevance of these gradient-based features is prioritized via a selective flow, and furthermore, their spatial relationship is revealed through an undirected graphical model. We strive to avoid the training phase to reduce the time required to bring the system into a fully operational state. The reliability of the here devised approach is evaluated on the set of video sequences captured during different phases of a day and the results are compared against the ground truth data.

1 Introduction

Video-based surveillance systems had evolved into sophisticated systems during the last 70 years. During that time, CCTV systems had successfully spread through many various application areas including monitoring dangerous industrial processes, security systems in banks, streets, stores, systems supporting transport safety and traffic surveillance. In this paper, we will focus on a specific kind of traffic surveillance systems, on the so-called parking lot guidance systems and the related area of image analysis as the vision-based systems promise a number of advantages over the intrusive sensors [1]. The very first parking guidance information system was deployed in Aachen, Germany in 1971 [2]. In the present time, the problem of identifying free parking spaces in a large parking lot is a quite interesting task. In the past decade, quite a lot of work concerned in the vacant parking space detection appeared, e.g. [3–5]. There exist four main categories of parking guidance systems using different technologies including the counter-based, wired-sensor-based, wireless-sensor-based and vision-based approaches [6]. The goals of parking lot surveillance include counting parked cars, identifying the location, size or type of parked vehicles, monitoring the movement of cars and the activities of humans.

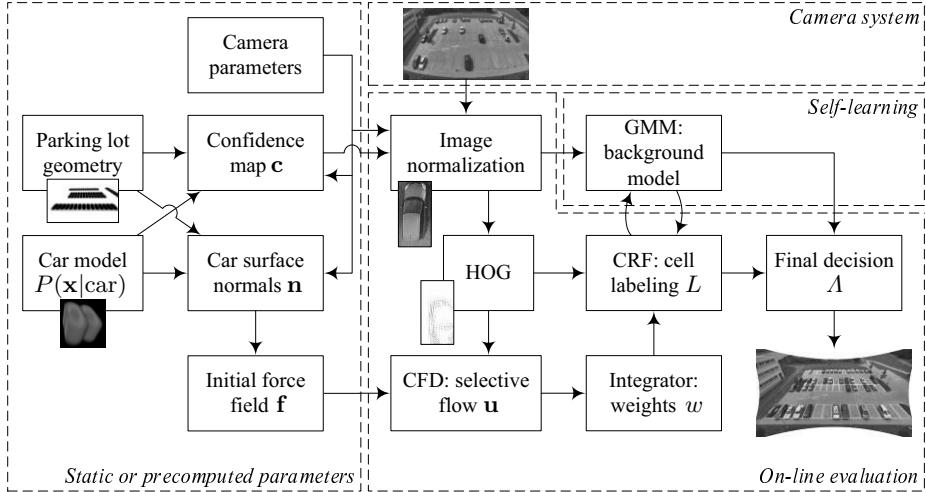


Fig. 1. The overview of our parking surveillance system. A priori known parking lot geometry, camera parameters, and probabilistic car model provide all the data necessary to calculate the confidence map for occlusion handling and expected car surface normals which form the force field initiating the advection of HOG features obtained from normalized images. The resulting flow prioritizes features that are conformal to the expected shape of a car and assign a weight to each cell. The final decision about the parking space state is devised from the labeling produced by CRF minimizing the related Gibbs energy consisting of potentials based on weight values and background GMM of an empty parking space

2 Occlusion Handling

Occlusions significantly affect the performance of object recognition and tracking algorithms. A lot of effort has been done in the area of occlusion handling in dynamic scenes, e.g. [7, 8]. Occlusions are also very common in parking lot images due to the spatial arrangement of parked cars and camera position and also some parking spaces may be heavily occluded by neighbouring parked cars. In order to cope with inter-vehicle occlusions, we propose a probabilistic 3D model of a vehicle. This model represents all feasible positions of vehicle inside the parking space. In the most simplistic way, the model can be represented by a cuboid positioned at the parking lot surface. The model itself is fully defined by its width, length, height, position of center and yaw. These parameters are treated as independent normally distributed random variables. As a result, we obtain a 3D scalar field of $128 \times 128 \times 256$ values representing the probabilities that the particular region inside the volume over a single parking space belongs to a vehicle. This can be expressed as the likelihood $P(\mathbf{x}|\text{vehicle})$, where \mathbf{x} represents some discrete volume (voxel) inside this scalar field. To put this model in the relation with the camera, we can cast a ray through the continuous scalar field $\rho : \mathbb{R}^3 \rightarrow \langle 0, 1 \rangle$ which is obtained as a trilinear interpolation of the discrete fields

of likelihoods $P(\mathbf{x}|\text{vehicle})$. With the aim of basic calculus, we can formulate the expression for the scalar field of occlusions h in terms of the line integral along a piecewise smooth curve L (line of view made up of the set of straight segments intersecting affected voxels V , see Fig. 2d) as follows

$$\begin{aligned} h(x, y) &= \bigcup_{\mathbf{x} \in V} P(\mathbf{x}|\text{vehicle}) = \int_L \rho(s) \, ds = \int_a^b \rho(\mathbf{r}(t)) \|\mathbf{r}'(t)\| \, dt \\ &= \int_a^b \rho(\mathbf{r}(t)) \|\hat{\mathbf{d}}\| \, dt = \int_a^b \rho(\mathbf{r}(t)) \, dt = \dots = \sum_{i \in I} \int_{t_i}^{t_{i+1}} \sum_{j=0}^3 a_j t^j \, dt, \quad (1) \end{aligned}$$

where the ray $\mathbf{r}(t) = O + \hat{\mathbf{d}}t$ is a bijective parameterization of the line segment originating at the point $\mathbf{r}(a)$ coincident with the camera's origin O and the end point $\mathbf{r}(b)$ that is the intersection with the parking lot plane. In addition, the integral over the interval (a, b) is decomposed into the sum of integrals over the set I of intervals (t_i, t_{i+1}) representing the parametric coordinates of intersections of the ray \mathbf{r} with the set of affected voxels V . The analytical derivation of parameters a_0, a_1, a_2 and a_3 is relatively straightforward but tedious, involving more than one hundred summands in the final expression. We left them out of this calculation to avoid unnecessary clutter. Equation (1) can be also evaluated numerically by probabilistic algorithms such as Monte-Carlo method. Returned scalar value of the function h represents the degree of our belief that the certain position (x, y) in the image of parking lot surface can be occluded exclusively by a well-parked car (see Fig. 2c). The resulting confidence field for the i -th parking space equals to $c_i(x, y) = 2h_i(x, y) - h(x, y)$, where h_i is the occlusion map where only i -th parking place is occupied and h is the occlusion map generated for the fully engaged parking lot. Figure 2a presents the unprojected confidence field for the first parking space and the projected version of the same field is in Fig. 2b.

3 Features Extraction and Prioritization

Since we know the camera position we can obtain the unwarped (or normalized) image of every parking space and the related confidence field c_i . We extract the relevant features that would allow us to discriminate between the two possible states of parking spaces. Dalal and Triggs [9] showed in their experiments that the Histogram of Oriented Gradient (HOG) is one of the most successful edge and gradient-based descriptor and significantly outperforms existing feature sets for human or vehicle detection. We use 8×16 grids of 8×8 pixel cells each containing $\beta = 9$ orientation bins corresponding to evenly spread sectors of the half angle ignoring the direction. We will refer to the HOG of i -th rectangular patch of an image using the following vector notation $\text{hog}_i = (v_0, v_1, \dots, v_{\beta-1}) \in \mathbb{N}^\beta$, where $\text{hog}(k) = v_k$ represents the number of votes for the k -th histogram channel or bin.

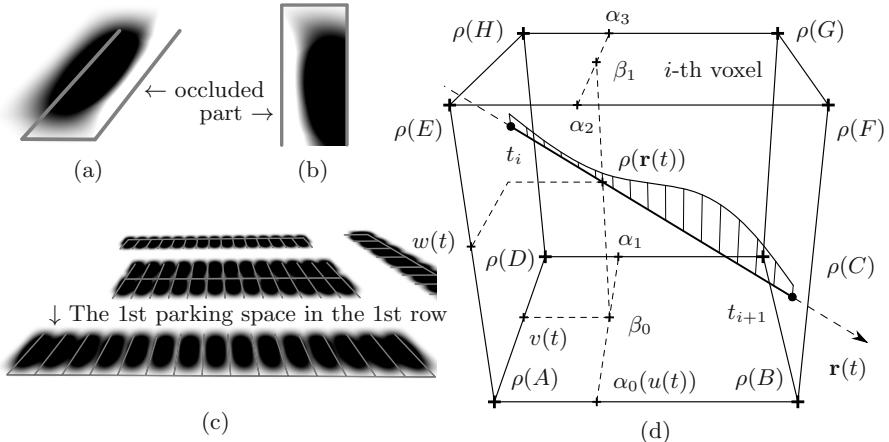


Fig. 2. (a)–(b) The scalar field c_1 represents the confidence that the pixels may belong to a vehicle parked exclusively on the first parking space. *Black level* refers to 100% confidence. (c) The scalar field h representing the parking lot surface occlusions. *Black level* refers to completely occluded parts of the parking ground. (d) The scheme for calculation of the line integral through i -th voxel

If we compare the image of an occupied parking space with an empty one, there is obvious difference in the distribution of prevailing edges. Simply put, the pertinence of the parking space to the given class may be devised from the total amount of cells, which can be regarded as the parts of parked car's edges. Moreover, based on the vector field \mathbf{n} obtained by projection of three-dimensional vector field of the car model iso-surface normals (e.g. for $h = 1$) onto the parking space image plane, we can roughly estimate the direction of such edges (i.e. the expected edge will be perpendicular to the local normal vector).

Physically-inspired from the classical fluid dynamics, we may think of every HOG-cell as an idealized flowing fan-like mass object. Cells more conformal with the edge model will experience stronger drag force resulting in a higher velocity of these cells. As a result, these cells will be easily advected by the flow field from the origin position into the detection zone. On the opposite side, cells with uniformly distributed bins will resist the flow. This will introduce the desired flexibility of our discrimination model with respect to the underlying car edge model. The motion of a Newtonian fluid with a constant density and temperature is governed by the Navier-Stokes equations (NSE) as follows

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad (2)$$

where \mathbf{u} represents the velocity vector field, p is the pressure field, ρ is the fluid density and ν is the kinematic viscosity of the fluid. The vector field \mathbf{f} is an external force field and will be discussed in the following Section 3.1. In the case

of incompressible fluids, the conservation of mass is then stated as the continuity equation $\nabla \cdot \mathbf{u} = 0$ meaning that the divergence of vector field \mathbf{u} is zero. For the sake of brevity, the implementation details of solving the NSE are left uncovered. We adopt the approach thoroughly described in [10].

The NSE were successfully applied in many fields including image analysis and in our case, we interpret the resulting pressure and velocity fields as follows. The low pressure areas correspond to the sources of strong gradients caused by eventual car edges located at the positions predicted by the field of projected normals. The high pressure regions will represent the traps for moving particles. If the particle arrive in the detection area and has a strong dominant bin in the HOG, then we can suppose that the origin of the particle is placed somewhere close to a strong edge of a parked vehicle. The trajectory \mathcal{P} in the conjunction with the actual bins configuration in particle's HOG should influence the speed of the moving particle. The new position \mathbf{r} of the cell with the total mass m in the particular time step $t+1$ is given by the formula $\mathbf{r}^{t+1} = \mathbf{F}^t \delta t^2 / m + 2\mathbf{r}^t - \mathbf{r}^{t-1}$. The steady-state drag force \mathbf{F} on the cell due to the fluid flow is derived from the standard quadratic drag equation for an object moving through a fluid and equals to

$$\mathbf{F}^t = \frac{1}{2} \rho \mathbf{u} \|\mathbf{u}\| \max_{i \in \langle 0, b \rangle} \left\{ \left(1 - |\text{rad2grad}(\text{bin2rad}(i)) \cdot \hat{\mathbf{u}}| \right) C(\text{hog}(i)) \right\}, \quad (3)$$

where the function rad2grad translates the angle in radians to unit direction vector and bin2rad converts i -th bin to radians. The constant b represents number of bins per orientation histogram. The hat over the \mathbf{u} means that it is a unit vector and has magnitude equal to 1. The drag coefficient C is associated with the number of votes $v_i := \text{hog}(i)$ in the i -th histogram channel through simple polynomial function $C(v_i) = \alpha v_i^\beta$. For the rest of our experiments, the parameters were set as follows: $\alpha = 3$ and $\beta = 5$.

3.1 Force Field Generation

We expect that the external velocity field will start transferring the features from the regions of their abundance into the detection areas (see Fig. 3b). We can start with the gradient of the iso-surface of the h function which is subsequently projected on every parking place yielding a 2D vector field of normals \mathbf{n} . The original normal vector field \mathbf{n} is very close to fulfil the stated requirements on the field \mathbf{f} which will initiate the motion of cells during the CFD steps. In order to assure that the force field fulfil the stated requirements, we define the force field to be the vector field $\mathbf{f}(x, y) = [u(x, y), v(x, y)]$ that minimizes the global energy functional

$$\mathcal{E} = \iint_{\Omega} \lambda_1 \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right) + \lambda_2 \|\mathbf{n}\|^2 \|\mathbf{f} - \mathbf{n}\|^2 + \|\nabla c\|^2 \|\mathbf{f} - \nabla c\|^2 \, dx \, dy, \quad (4)$$

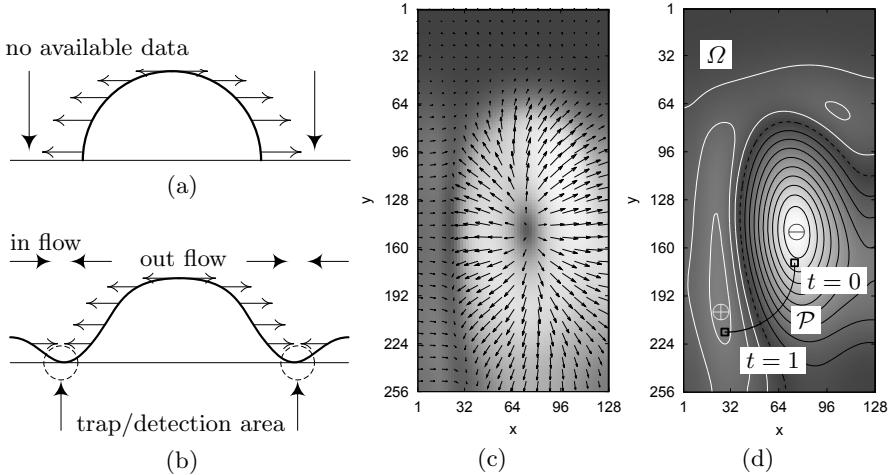


Fig. 3. (a) The original force field as obtained from the normal field \mathbf{n} and (b)–(c) the resulting force field fulfilling the constraints imposed by the energy functional \mathcal{E} . (d) The high-pressure field (white iso lines) marks the detection areas along the car boundaries

where the first term in the functional follows a standard principle, that of making the result smooth when there is no data. The second term is the data attachment term, whose minimization tends to make the force field to be similar with the normal field especially in the areas where the normal field is large. The third confidence field driven term enforces the presence of an in-flow from border areas and also partially helps to increase the pressure in the detection areas. After applying the standard methods of the variation calculus we obtain two Euler-Lagrange equations

$$\begin{aligned} \lambda_2 \|\mathbf{n}\|^2 (u - m) + \|\nabla c\|^2 (u - p_x) - \lambda_1 \Delta u &= 0, \\ \lambda_2 \|\mathbf{n}\|^2 (v - n) + \|\nabla c\|^2 (v - p_y) - \lambda_1 \Delta v &= 0, \end{aligned} \quad (5)$$

where Δ is the Laplace operator. Both Eqs. (5) can be solved iteratively by treating u and v as functions of time t according the time-marching scheme. These equations are decoupled, and therefore can be solved as separate scalar partial differential equations in u and v , provided that the partial derivatives with respect to time t on the left side of the Eqs. (6) are approximated by the first-order accurate forward difference formulas yielding

$$\begin{aligned}
u^{t+1}(x, y) &= u^t(x, y) + \delta t \left(\lambda_2 \|\mathbf{n}(x, y)\|^2 (u^t(x, y) \right. \\
&\quad \left. - m(x, y)) + \|\nabla c(x, y)\|^2 (u^t(x, y) - c_x(x, y)) - \lambda_1 \Delta u^t(x, y) \right), \\
v^{t+1}(x, y) &= v^t(x, y) + \delta t \left(\lambda_2 \|\mathbf{n}(x, y)\|^2 (v^t(x, y) \right. \\
&\quad \left. - n(x, y)) + \|\nabla c(x, y)\|^2 (v^t(x, y) - c_y(x, y)) - \lambda_1 \Delta v^t(x, y) \right).
\end{aligned} \tag{6}$$

The iteration begins by setting $u^0(x, y) = m(x, y)$ and $v^0(x, y) = n(x, y)$. To ensure the convergence of the above described iterative process, we restrict the time step δt with the Courant-Friedrichs-Lowy (CFL) condition. An example of the resulting force field is shown in the Fig. 3c.

3.2 Weight Assignment Procedure

In this section we will describe how to assign certain weight w_i to the individual cell (i.e. the rate of belonging to the car edge). As stated above, we track the position \mathbf{r} of every cell as it moves across the simulation domain Ω represented by the normalized image of a parking space. In accordance with our cells advection model, the most relevant cells travel across high-pressure areas and should gain the most votes (or weight). This can be expressed by the following path integral

$$w_i = \int_{\mathcal{P}_i} \kappa(p(s)) ds = \int_0^1 \kappa(p(\mathbf{r}(t))) \|\mathbf{r}_t(t)\| dt, \tag{7}$$

where \mathcal{P}_i is the trajectory taken by the i -th cell due to the influence of the flow field \mathbf{u} (see Fig. 3d). The factor $\|\mathbf{r}_t(t)\|$ represents the speed of traversal of the trajectory as the parameter t runs between two endpoints $t = 0$ and $t = 1$. The real-valued function κ converts the pressure into the weight value. We suggest to define this function as $\kappa(x) = \sqrt{\max(x, 0)}$. This definition reflects our exclusive interest in the areas with positive pressure and also reduces the influence of the pressure magnitude on the resulting weight.

4 CRF Toolkit for 3-Class Cell Labeling

In our approach, parking space status inference can be considered as a labeling problem that involves assigning HOG cells (or sites) \mathcal{S} a set of three labels $\mathcal{L} = \{0 = \text{no edge}, 1 = \text{unknown}, 2 = \text{car edge}\}$ with a subsequent decision Λ on the parking space state from the set $\Sigma = \{0 = \text{vacant}, 1 = \text{occupied}\}$. In other words, we seek for an optimal mapping $L : \mathcal{S} \rightarrow \mathcal{L}$ and a function $\Lambda : L \rightarrow \Sigma$. The mapping L is represented as a random field with the nodes aligned to the cells generated over the normalized image of a parking space. The random field $L = \{y_i : i \in \mathcal{S}\}$, where each random variable y_i takes on a value from the set of labels \mathcal{L} , is realized as a Conditional Random Field (CRF). The CRF is an undirected graphical model originally developed for labeling sequential data [11].

Later, Kumar and Herbert [12] introduced the generalized discriminative framework for 2D images which allows the modeling of different types of interactions in labels and data.

Here, we want to find a configuration of L which maximizes the a posteriori (MAP) estimate of the underlying field given the observed data. This problem can be reformulated as a minimization of an energy function

$$E(\mathbf{y}; \mathbf{x}, \Theta) = \sum_{i \in \mathcal{S}} \psi_i^U(y_i; \mathbf{w}, \mathbf{c}, \Theta) + \sum_{(i,j) \in \mathcal{N}} \psi_{i,j}^P(y_i, y_j; \mathbf{h}), \quad (8)$$

where the feature vector \mathbf{x} contains acquired weights \mathbf{w} , representative colors of individual cells \mathbf{c} , and the HOG \mathbf{h} . Minimizing the energy function is known to be NP-hard problem. There exist the approximate solutions such as graph cut, generalized belief propagation (GBP), and the tree re-weighted message passing (TRW). All our experiments are performed with the loopy belief propagation (LBP). Description of the inference method is beyond the scope of this paper and here we present the main ideas of the related potential function design.

Unary Potential. We could simply assume that the parked car will introduce a lot of edges all around the normalized image of a given parking space. Alas, this assumption may be violated by a strong ground pattern. Therefore, to assign the site labeling, we rely on the weights w which should be more resistant to this phenomenon than the original image gradient magnitude. To verify the edge hypotheses, we also take into account the color \mathbf{c}_i representing each cell. This yields the unary potential

$$\psi_i^U(y_i; \mathbf{w}, \mathbf{c}, \Theta) = \underbrace{-\log P_s(y_i|w_i, \Theta)}_{\text{CFD shape prior}} - \lambda_U \underbrace{\log \mathcal{L}_{y_i}(P_c(k_{\text{best}}|\mathbf{c}_i, \Theta))}_{\text{GMM color model}}, \quad (9)$$

that consists of two compounds. The first one represents the shape prior

$$P_s(y_i|w_i, \Theta) = \begin{cases} 1 - P(y_i = \text{ce}|w_i + \tau_{\text{ce}} - \tau_{\text{ne}}, \Theta) & , y_i = \text{ne} \\ 1 - (P_s(y_i = \text{ne}|w_i, \Theta) + P(y_i = \text{ce}|w_i, \Theta)) & , y_i = \text{u} \\ P(y_i = \text{ce}|w_i, \Theta) & , y_i = \text{ce} \end{cases}. \quad (10)$$

where the posterior probability of labeling the i -th site (or cell) equals to a one-dimensional sigmoidal function as follows

$$P(y_i = \text{car edge}|w_i, \Theta) = \frac{1}{1 + \exp((\tau_{\text{ce}} - w_i)/s)}. \quad (11)$$

The probability P_s relates weight values with the probabilities of particular labels (see Fig. 4 for further reference). The second compound P_c is the likelihood that the given cell color \mathbf{c}_i match the most probable mixture component k_{best} of the actual GMM background model containing a mixture of $K = 2$ Gaussian

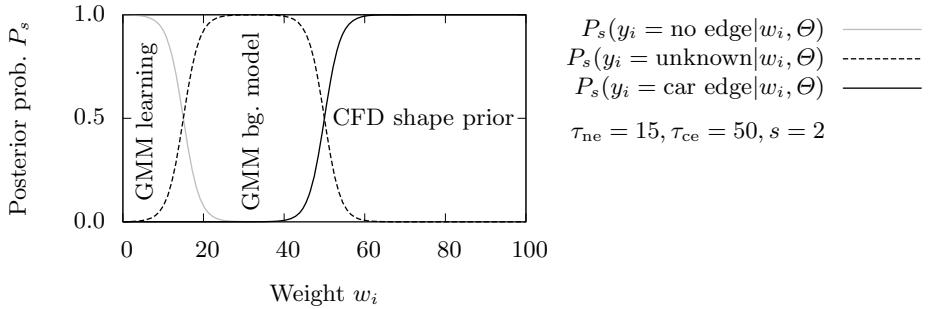


Fig. 4. Posteriors used for defining the shape priors in the unary potential. The Π -shaped function (dashed line) represents the range of unreliable weight values originating from the CFD shape prior

densities (for shadowed and unshadowed regions). Besides the GMM, the set of global parameters Θ contains three experimentally evaluated constants: s is the steepness factor, and the threshold values τ_{ne} , τ_{ce} are the inflection point of the sigmoid function for the no edge label and car edge label. Translation functions $\mathcal{L}_{y_i}(x)$ return x , 0, and $1 - x$, respectively.

Pairwise Potential. In our approach, to define the pairwise relationship between two neighbouring cells, we need some criterion that will compare histograms of two given cells. Our dissimilarity measure is defined as follows

$$\phi(i, j) = \frac{4}{\beta^2} \sum_{k=0}^{\beta-1} \sum_{l=0}^{\beta-1} \frac{\text{hog}_i(k) \text{hog}_j(l)}{(\text{hog}_i(k) + \text{hog}_j(l))^2} \frac{\rho(k, l)}{\rho_{\max}}, \quad (12)$$

where the metric ρ returns the distance between two bins k and l of a histogram and is given by the formula

$$\rho(k, l) = \beta \left| u - \lfloor u \rfloor - \left\lfloor u - \lfloor u \rfloor + \frac{1}{2} \right\rfloor \right|, \quad (13)$$

where $u = |k - l| / \beta$. The constant $\rho_{\max} = \lfloor \beta/2 \rfloor$ in Eq. (12) represents the maximum possible distance among all bins. Finally, the pairwise potential ψ^P is given by the multiplication of the dissimilarity measure and the delta function

$$\psi_{i,j}^P(y_i, y_j; \mathbf{h}) = \lambda_P \phi(i, j) \delta_{y_i, y_j}. \quad (14)$$

We expect that such definition will favour the same labeling in the case of similar distributions of votes in the histograms of the cells with large weights.

We use the loopy belief propagation (max-sum algorithm) to approximate the MAP inference. The state of the parking space is simply devised from the sum of the individual car edge labels of the final labeling (see Fig. 5).

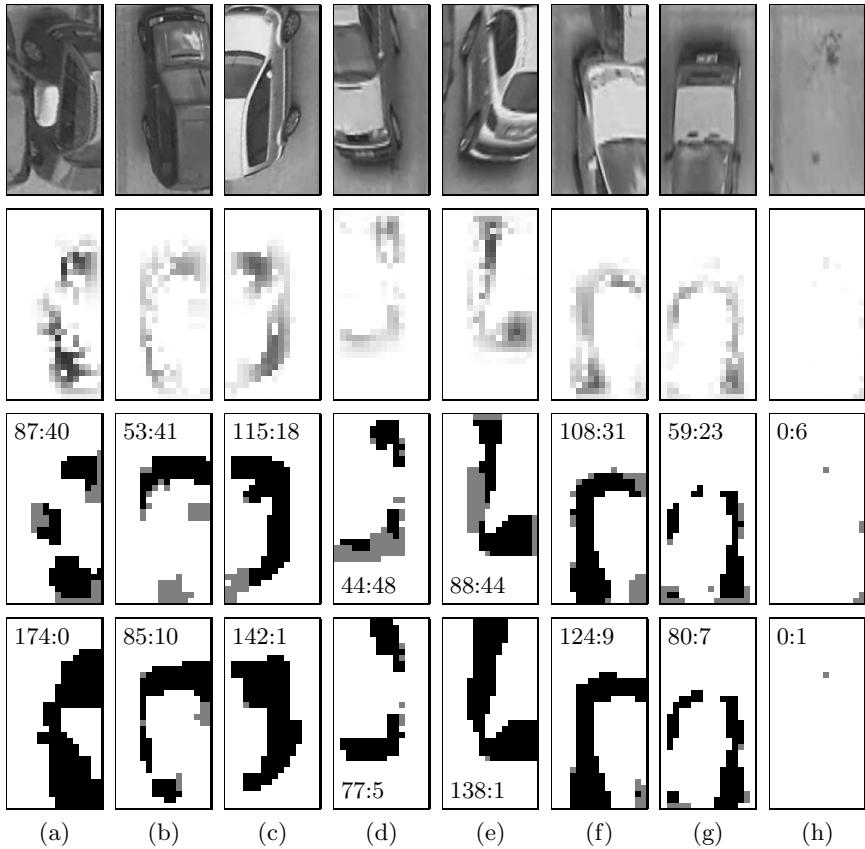


Fig. 5. (first row) Examples of normalized images, (second row) corresponding weight maps of selected edges, and (third row) labeling obtained via the CRF without the background term. (h) Empty parking spaces containing only a few cells labeled as unknown (*gray*, the second number) provide the reliable sources of color patches for generating and updating the background GMM. (fourth row) When enabled, the background color model strengthens the difference between vacant and occupied parking spaces in terms of number of cells labeled as car edges (*black*, the first number). The parameters are: $\lambda_U = 0.45$, $\lambda_P = 50$

5 Evaluation

The evaluation is based on the series of four video sequences captured during the most relevant parts of a day when a lot of cars are arriving or leaving monitored parking lot and it is summarized in the Table 1. We observed 56 parking spaces in 4 rows with the network HD camera. Individual test images contain more than 14000 occupied parking spaces and more than 8000 vacant parking spaces in total. Performance evaluation includes confusion matrix, F1 score, and MCC.

Table 1. Results of our algorithm compared against the ground truth data

Original image	Row	TP	FN	FP	TN	Acc.	Recall	Prec.	Spec.	F1	MCC
	1	584	15	19	1181	0.981	0.975	0.968	0.984	0.972	0.958
	2	486	7	1	1065	0.995	0.986	0.998	0.992	0.992	0.988
	3	606	2	2	949	0.997	0.997	0.997	0.998	0.997	0.995
	4	1235	11	14	540	0.986	0.991	0.989	0.975	0.990	0.967
	All	2911	35	36	3735	0.989	0.988	0.988	0.990	0.988	0.979
	1	1667	0	13	120	0.993	1.000	0.992	0.902	0.996	0.946
	2	1416	0	0	144	1.000	1.000	1.000	1.000	1.000	1.000
	3	1363	0	12	184	0.992	1.000	0.991	0.939	0.996	0.965
	4	1695	0	3	102	0.998	1.000	0.998	0.971	0.999	0.985
	All	6141	0	28	550	0.996	1.000	0.995	0.952	0.998	0.973
	1	1248	0	57	495	0.968	1.000	0.956	0.897	0.978	0.926
	2	579	0	1	979	0.999	1.000	0.998	0.999	0.999	0.999
	3	975	0	3	582	0.998	1.000	0.997	0.995	0.998	0.996
	4	927	1	84	781	0.953	0.999	0.917	0.903	0.956	0.909
	All	3729	1	145	2837	0.978	1.000	0.963	0.951	0.981	0.957
	1	353	0	33	364	0.956	1.000	0.915	0.917	0.955	0.916
	2	423	1	0	226	0.998	0.998	1.000	1.000	0.999	0.997
	3	392	0	9	247	0.986	1.000	0.978	0.965	0.989	0.971
	4	627	4	1	118	0.993	0.994	0.998	0.992	0.996	0.975
	All	1795	5	43	955	0.983	0.997	0.977	0.957	0.987	0.963

6 Conclusion

We presented a new algorithm for the vision-based evaluation of parking lot utilization based on the analysis of the spatial arrangement of HOG features in the normalized image of a single parking space. The hypothesis about their expected arrangement in the case of occupied parking space is build upon the probabilistic car model which comprises the available data about the layout of the parking lot and the camera system. The utilization of a priori known information allows us to remove the problematic learning phase, and as a result, the method does not require preparation of any training data set. The consistency of obtained features with an expected shape of potentially parked car is evaluated by the means of features prioritization through the adopted CFD technique that adds the flexibility to adapt the parked car appearance model to various possible configurations (e.g. car size, position, and yaw). The contextual constraints represented by the CRF ensure the spatial consistency of the final labeling as

well as the integration of the self-learned parking lot background color model. The experiments show that the algorithm performs well over the wide range of lighting conditions and the achieved F1 score was no lower than 98.1% in case of all rows. In our future work, we also plan to address an effective utilization of GPUs to reduce the overall latency of our parking lot surveillance system.

Acknowledgement. This work was supported by the grant SP2013/185 of VSB-TU of Ostrava, Faculty of Electrical Engineering and Computer Science.

References

1. Lee, S., Yoon, D., Ghosh, A.: Intelligent parking lot application using wireless sensor networks. In: Proc. Int. Symposium on Collaborative Technologies and Systems (CTS), Irvine, California, USA, pp. 48–57 (2008)
2. Lin, S.F., Chen, Y.Y., Liu, S.C.: A vision-based parking lot management system. In: Proc. Int. Conf. on Systems, Man and Cybernetics (SMC), Taipei, Taiwan, pp. 2897–2902 (2006)
3. Wu, Q., Huang, C., Yu Wang, S., Chen Chiu, W., Chen, T.: Robust parking space detection considering inter-space correlation. In: Proc. Int. Conf. on Multimedia and Expo (ICME), Beijing, China, pp. 659–662 (2007)
4. Huang, C.C., Wang, S.J.: A hierarchical bayesian generation framework for vacant parking space detection. Transactions on Circuits and Systems for Video Technology 20, 1770–1785 (2010)
5. Ichihashi, H., Katada, T., Fujiyoshi, M., Notsu, A., Honda, K.: Improvement in the performance of camera based vehicle detector for parking lot. In: Proc. Int. Conf. on Fuzzy Systems (FUZZ), Barcelona, Spain, pp. 1–7 (2010)
6. Bong, D., Ting, K., Lai, K.: Integrated approach in the design of car park occupancy information system. IAENG Int. Jour. of Comp. Sci. 35, 7–14 (2009)
7. Pan, J., Hu, B.: Robust occlusion handling in object tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Minneapolis, Minnesota, USA, pp. 1–8 (2007)
8. Yang, T., Pan, Q., Li, J., Li, S.Z.: Real-time multiple objects tracking with occlusion handling in dynamic scenes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), San Diego, California, USA, pp. 970–975 (2005)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR), San Diego, California, USA, pp. 886–893 (2005)
10. Stam, J.: Stable fluids. In: Proc. 26th Annual Conf. on Computer Graphics (SIGGRAPH), Los Angeles, California, USA, pp. 121–128 (1999)
11. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th Int. Conf. on Machine Learning (ICML), San Francisco, California, USA, pp. 282–289 (2001)
12. Kumar, S., Hebert, M.: Discriminative fields for modeling spatial dependencies in natural images. In: NIPS. MIT Press (2003)

Automatic Pain Intensity Estimation with Heteroscedastic Conditional Ordinal Random Fields

Ognjen Rudovic¹, Vladimir Pavlovic², and Maja Pantic^{1,3}

¹ Comp. Dept., Imperial College London, UK

² Dept. of Computer Science, Rutgers University, USA

³ EEMCS, University of Twente, The Netherlands

Abstract. Automatic pain intensity estimation from facial images is challenging mainly because of high variability in subject-specific pain expressiveness. This heterogeneity in the subjects causes their facial appearance to vary significantly when experiencing the same pain level. The standard classification methods (e.g., SVMs) do not provide a principled way of accounting for this heterogeneity. To this end, we propose the *heteroscedastic* Conditional Ordinal Random Field (CORF) model for automatic estimation of pain intensity. This model generalizes the CORF framework for modeling sequences of ordinal variables, by adapting it for heteroscedasticity. This is attained by allowing the variance in the ordinal probit model in the CORF to change depending on the input features, resulting in the model able to adapt to the pain expressiveness level specific to each subject. Our experimental results on the UNBC Shoulder Pain Database show that modeling heterogeneity in the subjects with the framework of CORFs improves the pain intensity estimation attained by the standard CORF model, and the other commonly used classification models.

1 Introduction

Automatic analysis of pain has received increased attention over the last few years mostly because of its applications in health care. For example, in intensive care units in hospitals, it has recently been shown that enormous improvements in patient outcomes can be gained from the medical staff periodically monitoring patient pain levels. However, due to the burden of work/stress that the staff are already under, this type of monitoring has been difficult to sustain, so an automatic system would be an ideal solution [1]. Recent research has evidenced the usefulness of facial cues for automatic pain analysis (e.g., see [2]), however, it has mainly focused on detection of presence/absence of pain.

In this paper, we address the problem of estimating the level of patients' shoulder pain from video recordings of their facial expressions, provided by the recently released UNBC-MacMaster Shoulder Pain Expression Archive Database [2]. The recorded patients suffer from chronic shoulder pain, intensity of which is quantified into discrete ordinal levels, ranging from no pain to the maximal level of pain, measured using the Prkachin and Solomon Pain Intensity (PSPI) metric [3]. As the patients perform a range of arm motion tests in front of the camera, the aim being to estimate their pain level in every frame of the video. This poses a number of challenges for the modeling task. First,

different pain levels are characterized by subtle changes of facial appearance within subjects, and large changes of facial appearance between subjects. This is because the latter depends on what constitutes the maximal level of the change in facial appearance of each subject. Consequently, the between-subject variation can easily overshadow the pain-intensity-related variation. Second, subjects' facial expressions are typically toned down due to a long-term exposure to chronic pain. Therefore, it is important to account for temporal dynamics of pain intensity changes.

To the best of our knowledge, only a few works ([1,4,5]) have addressed the problem of automatic pain intensity estimation so far. Lucey et al. [1] proposed a system for a three-level pain intensity estimation at the sequence level. The authors used the shape- and appearance-based features obtained using an Active Appearance Model (AAM). These features were then used to train separate Support Vector Machine (SVM) classifiers for each pain intensity level. To deal with spurious noisy signals, a moving-average smoothing filter was applied to the SVM output probability scores. Kaltwang et al. [4] proposed a feature-fusion approach for continuous pain intensity estimation based on the Relevance Vector Regression (RVR) model. As the input, the authors used the shape features, and the appearance features, obtained by computing the Discrete Cosine Transform (DCT) and Local Binary Patterns (LBPs) from the normalized facial appearance. As the targets for the regression model, the authors used the discrete pain intensity levels defined on a 16 point scale. Finally, Hammal and Cohn [5] performed the estimation of 4 pain intensity levels. The authors applied Log-Normal filters to the normalized facial appearance, which resulted in high-dimensional facial features. These features were then used to separately train SVMs for each pain intensity on a frame-by-frame basis. Note that the works mentioned above focus mainly on the feature extraction step. The classification/regression of the target pain intensity is performed consequently by applying the standard learning techniques for nominal data, therefore ignoring the fact that pain intensity is defined on the ordinal scale. Finally, none of these methods explore temporal dynamics of the pain intensity.

In this paper, we propose a model for pain intensity estimation that is based on the Conditional Ordinal Random Field (CORF) [6,7] model, specifically designed for estimation of sequences of ordinal variables. Although the CORF model can address the limitations of the existing methods mentioned above, its underlying assumption is that the noise on the ordinal targets (in our case, the pain intensity) is homogeneous, i.e., constant. To account for heterogeneity in the subjects, we need to relax this assumption. This is attained by allowing its variance to change depending on the input features, resulting in the *heteroscedastic* CORF model. In contrast to the existing methods for pain intensity estimation, the proposed model is able to adapt to varying pain expressiveness levels of different subjects. The benefit of this is reflected in the results of the experiments conducted on the ShoulderPain dataset [2].

The remainder of the paper is organized as follows. Sec.2 reviews standard ordinal regression models. In Sec.3 we introduce the proposed heteroscedastic CORF model. Sec.4 shows the results of the experimental evaluation, and Sec.5 concludes the paper.

2 Ordinal Regression Models

Different models for data with ordinal targets have been proposed (e.g., see [8] for an overview). In this paper we restrict the consideration to the popular probit threshold model proposed by McCullagh(1980) [9]. In this model, it is assumed that there is a latent continuous variable Y^* that underlies the observed ordinal response Y . For example, in the context of the target task, Y represents intensity of pain described as ‘none’, ‘moderate’ or ‘severe’. These outcomes may literally be considered as resulting from pain severity, the unobserved continuous latent response Y^* . Since we are interested in the intensity of pain, we need to model the relationship between the unobserved variable Y^* (i.e., the latent process causing pain) and the observed response Y (i.e., the intensity of pain). This relationship can be expressed using the following probit threshold models.

2.1 Homoscedastic Threshold Model

Let $Y^* = f(x) + \sigma Z$ be a 1-D continuous latent variable, where x is a vector of covariates (i.e., image features), where $f : X \rightarrow \mathbb{R}$ and Z is a noise variable with the standard normal distribution $\mathcal{N}(0, 1)$. The probability distribution function of Y^* is then given by $\Pr(Y^* \leq z) = \Phi\left(\frac{z-f(x)}{\sigma}\right)$, where $\Phi(\cdot)$ is the CDF of the standard normal distribution. Under the threshold concept, the observed ordinal response Y is obtained as $Y := \{y \in \{1, \dots, R\} | b_{y-1} < Y^* \leq b_y\}$, where $b_0 = -\infty \leq \dots \leq b_R = \infty$ are increasing thresholds or cut-off points. The conditional probability of Y is then given by:

$$\Pr(Y = y|x) = \Phi\left(\frac{b_y - f(x)}{\sigma}\right) - \Phi\left(\frac{b_{y-1} - f(x)}{\sigma}\right). \quad (1)$$

2.2 Heteroscedastic Threshold Model

The homoscedastic threshold model has some limitations. In real-world data, the uncertainty of the labels may depend on the input x . That is, on some x the label y will almost certainly appear, and on other x the label Y may have nearly uniform distribution [8]. This can be leveraged by allowing the scale σ to depend on inputs x , i.e., $\sigma \equiv \sigma(x)$, where $\sigma : X \rightarrow \mathbb{R}_+$, with \mathbb{R}_+ denoting the set of positive real numbers. Using the notation from Sec.2.1, the continuous latent variable is defined as $Y^* = f(x) + \sigma(x)Z$. Then, the conditional distribution function of Y with heteroscedastic noise is

$$\Pr(Y = y|x) = \Phi\left(\frac{b_y - f(x)}{\sigma(x)}\right) - \Phi\left(\frac{b_{y-1} - f(x)}{\sigma(x)}\right), \quad (2)$$

where the uncertainty of labels is adjusted by the intensity of $\sigma(x)$.

3 Heteroscedastic Conditional Ordinal Random Fields

In this section, we present the proposed model for automatic pain intensity estimation. The model is based on the CORF model for temporal data with ordinal targets.

We extend this model by accounting for heterogeneity of subjects, which is incorporated in the model by using the modeling approach of heteroscedastic ordinal regression from the previous section.

3.1 The Model

Consider the standard Conditional Random Field (CRF) [10] model. It represents the conditional distribution $P(\mathbf{y}|\mathbf{x})$ as the Gibbs form clamped on the observation \mathbf{x} :

$$P(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}; \boldsymbol{\theta})} e^{s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}, \quad (3)$$

where $Z(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{y} \in \mathcal{Y}} e^{s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}$ is the normalizing partition function (\mathcal{Y} is a set of all possible output configurations), and $\boldsymbol{\theta}$ are the model parameters¹ of the *score function*.

The choice of the output graph $G = (V, E)$ and the cliques critically affects the representational capacity and the inference complexity of the model. For simplicity, a linear-chain model with *node* cliques ($r \in V$) and *edge* cliques ($e = (r, s) \in E$) is often assumed. By letting $\{\mathbf{v}, \mathbf{u}\}$ be the parameters of the node features, $\boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, y_r)$, and the edge features, $\boldsymbol{\Psi}_e^{(E)}(\mathbf{x}, y_r, y_s)$, respectively, the score function $s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$ can be expressed as the sum:

$$s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \sum_{r \in V} \mathbf{v}^T \boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, y_r) + \sum_{e=(r,s) \in E} \mathbf{u}^T \boldsymbol{\Psi}_e^{(E)}(\mathbf{x}, y_r, y_s). \quad (4)$$

The score function in (4) has a great modeling flexibility, allowing the node and edge features to be chosen depending on the target task.

Node Features. In the CORF framework, the node features are defined using the homoscedastic ordinal regression model in (1). In our model, we use the heteroscedastic ordinal regression model, defined in (2), to set the node features as:

$$\mathbf{v}^T \boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, y_r) \rightarrow \sum_{c=1}^R I(y_r = c) \cdot \left[\Phi\left(\frac{b_{y_r} - f(\mathbf{x}_r)}{\sigma(\mathbf{x}_r)}\right) - \Phi\left(\frac{b_{y_r-1} - f(\mathbf{x}_r)}{\sigma(\mathbf{x}_r)}\right) \right]. \quad (5)$$

By applying the Representer Theorem to the regularized negative log-likelihood in (9), we obtain the optimal functional form for the location model $f(\cdot)$ as

$$f(\mathbf{x}_*) = \sum_{i=1}^S \alpha_i k_f(\mathbf{x}_i, \mathbf{x}_*), \quad (6)$$

where $k_f(\cdot, \cdot)$ is a Mercer kernel, and S is the number of kernel bases. Similarly, the scale model $\sigma(\cdot)$ is obtained as

$$\sigma(\mathbf{x}) = \exp(\beta_0 + \sum_{i=1}^M \beta_i k_\sigma(\mathbf{x}_i, \mathbf{x}_*)), \quad (7)$$

¹ For simplicity, we often drop the dependency on $\boldsymbol{\theta}$ in notations.

where we also include an intercept β_0 , so when the data do not exhibit heterogeneity (or they do, but to a lesser extent), we recover the homoscedastic ordinal model. Also, to guarantee the non-negativity of σ , we use the exponential form of the kernel function.

The most important aspect of using the varying scale $\sigma(\mathbf{x})$ is that the inputs x can now directly influence the locations of the thresholds b in the ordinal model, which are constant in the homoscedastic CORF model. In this way, the proposed model with heteroscedastic (ordinal) node features can automatically adapt its thresholds to account for individual differences in pain tolerance and/or the level of individual pain expressiveness.

Edge Features. The edge features are defined as in the standard CRF model, i.e., using the absolute difference between the features of the temporally neighbouring frames, resulting in

$$\Psi_e^{(E)}(\mathbf{x}, y_r, y_s) = \left[I(y_r = k \wedge y_s = l) \right]_{R \times R} \otimes |\mathbf{x}_r - \mathbf{x}_s|, \quad (8)$$

where $I(\cdot)$ is the indicator function that returns 1 (0) if the argument is true (false) and \otimes denotes the Kronecker product. The role of the edge features is to enforce smooth predictions of the pain intensities across time.

With the node and edge features as defined above, we arrive at the following optimization problem:

$$\arg \min_{\theta} \sum_{i=1}^N -\ln P(\mathbf{y}_i | f(\mathbf{x}_i), \sigma(\mathbf{x}_i), \theta) + \Omega(\theta), \quad (9)$$

where N is the number of the training image sequences, $\Omega(\theta)$ is the (kernel-inducing) regularizer, and $\theta = \{\mathbf{b}, \alpha, \beta, \mathbf{u}\}$ are the model parameters.

3.2 Regularizers

As the objective function in (9) is nonlinear and nonconvex, it is critical to regularize it to improve the model's performance. We apply L_2 regularizer to the kernel weights and parameters \mathbf{u} in order to avoid diverging solutions. To encourage the latent coordinates $f(\mathbf{x})$ to be close in the latent space, we employ the widely used Laplacian regularizer for kernels:

$$\Omega(\|f\|_K) = \sum_{i,j} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} = 2\alpha^T K L K \alpha, \quad (10)$$

where K is computed using the kernel function k_f , and $L = D - W$ is the graph Laplacian, with $D_{ii} = \sum_j W_{ij}$. The similarity W is derived using the target labels y as

$$W_{ij} = 1 - \frac{|y_i - y_j|}{R - 1}, \quad y_i, y_j = 1, \dots, R. \quad (11)$$

Notice that when the absolute difference between two pain intensities increases, the extent of distance enlargement in (11) increases accordingly. This regularization approach has been shown effective in other facial-expression-related modeling tasks (eg. see [7]).

3.3 Learning and Inference

To minimize the objective in (9), we use the quasi-Newton limited-memory BFGS method. We briefly describe the learning strategy. Initially, we set the scale models σ to 1 to form a homoscedastic model. This is accomplished by optimizing the parameters of the location model f , the ordinal thresholds b and the transition parameters u . In the next step, we fix the parameters of the homoscedastic model and optimize w.r.t. the parameters of the scale model. In the final run, we optimize all the parameters simultaneously. The regularization parameters are found using a cross-validation procedure on the training set. Once the parameters of the model are estimated, the inference of test sequences is carried out using Viterbi decoding.

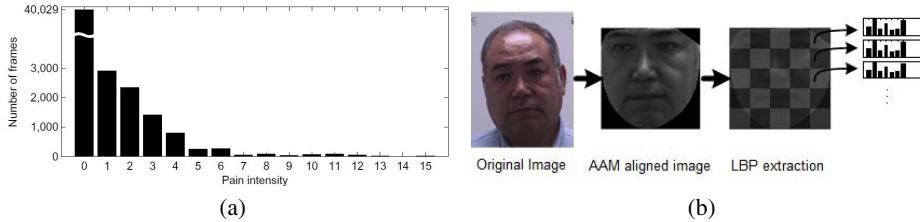


Fig. 1. a) Distribution of the pain intensity levels in The ShoulderPain dataset [2], b) Feature extraction process

4 Experiments

We conducted experiments on The ShoulderPain dataset [2] containing video recordings of patients suffering from shoulder pain while performing range-of-motion tests of their arms (see Sec. 1 for details). 200 sequences of 25 subjects were recorded (48,398 frames in total). For each frame, discrete pain intensities (0-15) according to Prkachin and Solomon [3] are provided by the database creators (see Fig. 1(a)). All the image sequences with the pain intensity > 0 were pre-segmented, so that the number of frames with the intensity 0, the most frequent in the dataset, was balanced with the second most frequent intensity. The resulting intensity distribution was still highly imbalanced, so we discretized it into 6 pain levels as: 0 (none), 1 (mild), 2 (discomforting), 3 (distressing), 4-5 (intense), and 6-15 (excruciating). The ratio of the highest and the lowest pain level was 3:1. This data balancing was performed in order to avoid the tested methods overfitting the majority classes. To evaluate the methods, we selected 147 image sequences from 22 subjects, 10 of which were used as the training set, and the rest as the test set.

To obtain the input features, we first aligned the image frames using a piece-wise affine warp based on the 66 points of the AAM provided by the database creators (see [2,4] for details). The aligned images were then divided into 6x6 even patches to preserve local texture information. From each image patch we extracted Local Binary Patterns (LBP) [11] with radius 2, resulting in 59 histogram bins per patch. This process is outlined in Fig. 1(b). We used LBPs as the input features since they have been shown to perform well for the facial affect data (e.g., see [7,4]).

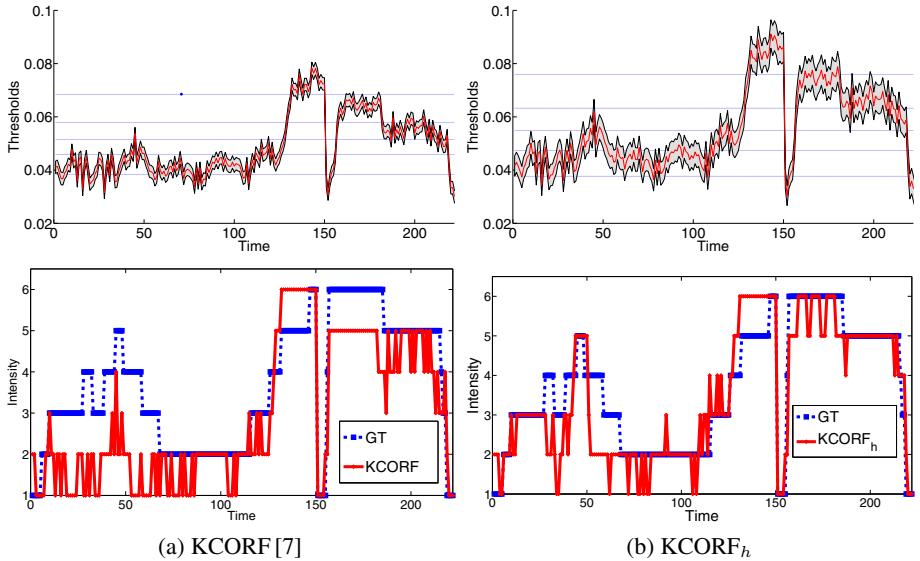


Fig. 2. Comparison of the: (a) homoscedastic and the proposed (b) heteroscedastic KCORF models with the same dynamic features. The upper row shows the values of the latent variable Y^* across time, where the horizontal lines are the learned thresholds. The estimated variance is also shown on Y^* . The *Time* represents the frame number, where we concatenated two sequences of two test subjects (1-150 / subject 1, 151-222 / subject 2). Note the change in variance in the heteroscedastic model as the subjects change. The bottom row shows the intensity prediction by the two methods.

We compare the proposed heteroscedastic (kernel) CORF (KCORF_h) model with its homoscedastic counterpart, KCORF [12], recently proposed for AU temporal segmentation. We used 150 kernel bases for the location and scale models. The bases selection was performed by sampling 25 kernel bases from each pain intensity at random. It was found that this is a good trade-off between the performance and computational complexity of the models. Using the small number of kernel bases also helped to reduce the overfitting. For both the kernel methods, we used the Histogram Intersection (CHI) kernel [13], since it is a non-parametric kernel, and, therefore, it does not involve learning of additional parameters. The balancing trade-off between the regularization and the log-likelihood terms was estimated by grid search under cross validation on the training data.

As a baseline model, we used one-vs-all SVM [14], since most of the prior work on pain intensity estimation is based on this classifier. We also performed comparisons with the state-of-the-art static ordinal regression models, Support Vector Ordinal Regression with implicit constraints (SVOR) [15] and Gaussian Process Ordinal Regression [16]. For the kernel methods, we used the same kernel function as explained above. Finally, we performed the comparison with the base models for sequential data: Gaussian Hidden Markov Models (GHMM)[17] and linear-chain Conditional Random fields (CRFs) [18], since these models are commonly used for modeling sequential data. For the GHMM, each pain intensity level was treated as the model's state parametrized

using a single Gaussian. We also included comparisons with the Laplacian-regularized Conditional Ordinal Random Field (CORF) [12] model, recently proposed for emotion intensity estimation. Because learning in the linear models (GHMM/CRF/CORF) is intractable due to the high dimensionality of the input features, we applied different dimensionality reduction techniques. The reported results are the best obtained, and they were achieved using the 6D features derived with the Kernel Locality preserving projections [19]. The performance of the tested models is reported using: (i) average F-1 measure computed from predictions for each pain intensity, (ii) the mean absolute loss computed between actual and predicted pain intensities, and (iii) Intra-Class Correlation (ICC(3,1) [20]). The ICC is commonly used in behavioral sciences to quantify agreement between different coders, and it is a measure of correlation or conformity of data with multiple targets. The higher the ICC the better.

Fig.2 shows the latent variable learned in the homoscedastic KCORF and the proposed heteroscedastic $KCORF_h$ model. Note that the variance in the heteroscedastic model varies across time. This is especially true when switching between the subjects. The change in the variance helps to adjust the locations of the intensity thresholds in the heteroscedastic ordinal model depending on the test subject. Therefore, depending on the pain expressiveness level of each subject, the model changes its parameters accordingly. Based on the prediction results shown in Fig.2, it is evident that this helps to improve estimation of the pain intensity levels, especially of the higher levels. For example, around the frame number 50, the heteroscedastic model correctly detects level 5, in contrast to the homoscedastic model. Also, the heteroscedastic model gives smoother predictions compared to the homoscedastic model. Since both models use the same dynamic features, we attribute this to the heteroscedastic component in the proposed model.

Table 1 shows the performance of different classification methods applied to the target task. First, note that all methods attain low the F-1 measure. This is expected because the large variation in facial appearance of different subjects poses a significant challenge for any classifier. We checked the training results of the tested methods and found that all methods attained significantly higher F1 values. This overfitting of the models is ascribed to the fact that subject-specific variation in the used features dominates over the pain-level-specific variation. We next examine how far off are the predictions from the labels. This is reflected in the absolute loss by the tested models. Note that the standard classification methods (SVM/GHMM/CRF) exhibit the highest loss, followed by the static ordinal regression models (SVOR/GPOR). The better results are attained by the dynamic ordinal models, i.e., KCORF and $KCORF_h$, with the latter performing the best. This evidences that both the ordinal and temporal modeling contribute to improving the pain intensity estimation. Furthermore, accounting for heterogeneity in subjects

Table 1. The performance of different methods applied to the task of automatic pain intensity estimation. The features for the linear models (GHMM/CRF/CORF) were pre-processed using KLPP[19].

Methods	SVM	SVOR	GPOR	GHMM	CRF	CORF	KCORF	$KCORF_h$
F-1 [%]	31.1	33.9	34.1	24.8	34.7	35.5	36.8	40.2
Abs. Loss	1.25	1.10	1.07	1.30	1.22	0.92	0.88	0.80
ICC [%]	46.5	57.1	57.8	39.4	49.0	63.2	66.5	70.3

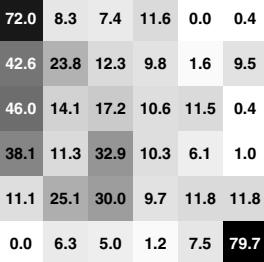
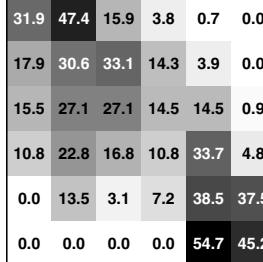
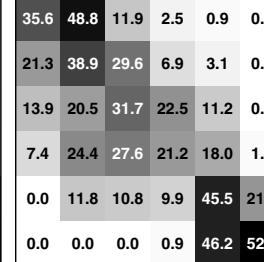
		
(a) SVM	(b) KCORF	(c) KCORF _h

Fig. 3. Confusion matrices obtained using different models. For a baseline, we include the results attained by the SVM-based method.

additionally helps to improve the estimation. The same conclusions can be drawn from the ICC scores for the tested models. However, it is important to mention that the ICC used here is insensitive to bias in the predictions, in contrast to the absolute loss. Nevertheless, the obtained scores reveal that the ordinal models exhibit better conformity between the predictions and the labels, with the proposed model achieving the highest score. To further analyze the performance of the models, we plot in Fig.3 confusion matrices for the SVM, KCORF, and proposed KCORF_h. Note that both the ordinal models confuse mostly the neighboring intensity levels, which explains their high ICC scores and low absolute loss. On the other hand, the misclassification by the SVM does not conform to any pattern. We attribute this to the fact that SVM treats the output variables as nominal. From Fig.3(a), it is also evident that the SVM fails to differentiate well between intermediate intensity levels, as opposed to the ordinal models. Finally, compared to the homoscedastic KCORF model, the KCORF_h reduces the misclassification with the classes being further from the diagonal, which, again, evidence the importance of modeling the heterogeneity in subjects.

5 Conclusion

In this paper, we proposed the heteroscedastic CORF model for automatic pain intensity estimation. The proposed model relaxes the homoscedasticity assumption in the CORF model, designed for modeling sequential ordinal data. Our experimental results indicate that, when LBPs are used as the image descriptors, the subjects in the dataset used do exhibit a certain level of heterogeneity. Based on the three performance measures used in our experiments, it is evident that accounting for this heterogeneity results in better pain intensity estimation attained by the proposed model compared to that attained by the homoscedastic ordinal model, and the other classification models.

Acknowledgments. This work has been funded in part by the EPSRC grant EP/H016988/1: Pain rehabilitation: E/Motion-based automated coaching. The work is further funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB), and by the National Science Foundation under Grant No. IIS 0916812.

References

- [1] Lucey, P., Cohn, J., Prkachin, K., Solomon, P., Chew, S., Matthews, I.: Image and Vision Computing (42), 197–205
- [2] Lucey, P., Cohn, J., Prkachin, K., Solomon, P., Matthews, I.: Painful data: The UNBC-McMaster shoulder pain expression archive database. In: FG, pp. 57–64. IEEE (2011)
- [3] Prkachin, K., Solomon, P.: The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. Pain 139, 267–274 (2008)
- [4] Kaltwang, S., Rudovic, O., Pantic, M.: Continuous pain intensity estimation from facial expressions. In: Bebis, G., et al. (eds.) ISVC 2012, Part II. LNCS, vol. 7432, pp. 368–377. Springer, Heidelberg (2012)
- [5] Hammal, Z., Cohn, J.F.: Automatic detection of pain intensity. In: Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI 2012, pp. 47–52. ACM (2012)
- [6] Kim, M., Pavlovic, V.: Structured output ordinal regression for dynamic facial emotion intensity prediction. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 649–662. Springer, Heidelberg (2010)
- [7] Rudovic, O., Pavlovic, V., Pantic, M.: Kernel conditional ordinal random fields for temporal segmentation of facial action units. In: Fusillo, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012 Ws/Demos, Part II. LNCS, vol. 7584, pp. 260–269. Springer, Heidelberg (2012)
- [8] Kanamori, T.: Statistical models and learning algorithms for ordinal regression problems. Information Fusion 14, 199–207 (2013)
- [9] McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society. Series B (42), 109–142
- [10] Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In: ICML, pp. 282–289 (2001)
- [11] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24, 971–987 (2002)
- [12] Rudovic, O., Pavlovic, V., Pantic, M.: Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In: CVPR (2012) (in press)
- [13] Barla, A., Odone, F., Verri, A.: Histogram intersection kernel for image classification. In: ICIP 2003, vol. 3,2, pp. III-513–III-516 (2003)
- [14] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
- [15] Chu, W., Keerthi, S.S.: New approaches to support vector ordinal regression. In: ICML, pp. 145–152 (2005)
- [16] Chu, W., Ghahramani, Z.: Gaussian processes for ordinal regression. JMLR 6, 1019–1041 (2005)
- [17] Murphy, K.P.: The bayes net toolbox for matlab. Computing Science and Statistics 33, 2001 (2001)
- [18] Lafferty, J.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, pp. 282–289. Morgan Kaufmann (2001)
- [19] He, X., Niyogi, P.: Locality Preserving Projections. In: NIPS (2004)
- [20] Shrout, P., Fleiss, J.: Intraclass correlations: uses in assessing rater reliability. Psychology Bulletin (1979)

Robot Trajectory Planning Using OLP and Structured Light 3D Machine Vision

M. Rodrigues¹, M. Kormann¹, C. Schuhler², and P. Tomek³

¹ Sheffield Hallam University, Sheffield, UK

² TWI – The Welding Institute, Cambridge, UK

³ MFKK Invention and Research Services Center Ltd, Hungary

Abstract. This paper proposes a new methodology for robotic offline programming (OLP) addressing the issue of automatic program generation directly from 3D CAD models and verification through online 3D reconstruction. Limitations of current OLP include manufacturing tolerances between CAD and workpieces and inaccuracies in workpiece placement and modelled work cell. These issues are addressed and demonstrated through surface scanning, registration, and global and local error estimation. The method allows the robot to adjust the welding path designed from the CAD model to the actual workpiece. Alternatively, for non-repetitive tasks and where a CAD model is not available, it is possible to interactively define the path online over the scanned surface.

1 Introduction

Welding represents one of the single largest applications of robots in manufacturing engineering, as approximately a quarter of all industrial robots are being used in connection to welding tasks [1]. The development of flexible automation systems that can be set up quickly and switched over to another product line are essential to increase productivity and profitability while maintaining product quality within prescribed tolerances. The challenge is that small and medium sized enterprises (SMEs) normally do not have the resources to invest in expensive technologies and extensive human training [2]. In particular, robot programming is a demanding specialised activity that, for non-repetitive tasks, can take several hundred times longer than the actual robot execution time [3].

There are two methods of robot programming, namely online and offline (OLP) programming. Online is normally carried out by skilled operators guiding the robot through a sequence of locations in space [4]. Although conceptually simple, for complex geometries it becomes difficult, very tedious and time consuming. Attempts have been made to improve online programming by the addition of sensors and additional calibration [5, 6]. However, the process has to be repeated again for a workpiece with a slightly different design. Despite these issues, online programming is the programming of choice for most SMEs.

OLP methods utilise 3D CAD data to generate and test robot programs and are widely used in automation systems with large product volumes [3]. Once the workpiece and robot cell are modelled, the operator can simulate the program

and test for collisions. The programs are then downloaded to the robot for execution. The main advantage is that OLP does not require the actual robot so it does not adversely affect utilisation time. Some limitations and open issues of OLP [7–9] can be summarised as follows:

- manufacturing tolerances between real and ideal CAD workpieces,
- inaccurate placement of the workpiece within the robot cell,
- inaccuracies between physical and modelled work cell,
- thermal effects during welding,
- lack of a methodology to deal with complex features.

The MARWIN project [10] develops a cognitive welding robot interface where welding tasks and parameters are intuitively selected by the end-user directly from a library of CAD models. No knowledge of robot programming is required, as robot trajectories are automatically calculated from the CAD models and validated through fast 3D scanning of the welding scene. The role of the user is limited to high level specification of the welding task and to the confirmation and/or changing of welding parameters and sequences as suggested by the control program. MARWIN uses a 3D structured light scanner where the light source and camera are in a parallel arrangement; its development and mathematical formulation have been described in [2, 11].

The focus of this paper is on the problem of automatic program generation in OLP as this is a perceived gap and a requirement, as no system exist in the market which implements the complete offline programming chain, although many separate *ad hoc* solutions exist [3]. The approach is to incorporate a fast area 3D scanner and propose a methodology for OLP that addresses most of the issues above, and can include pre- and post-verification of welding quality. Furthermore, the proposed method is also suitable for non-repetitive workpieces (for which CAD models may or may not be available) through a combination of sensor-guided online and offline programming.

2 Methodology

The method described here deals with interactive definition of control points defining a robotic welding path including the tooltip orientation and its translation to the actual workpiece – which may be slightly different from its ideal CAD model. This adaptive translation is the method's main novelty as it can deal with uncertainties between CAD descriptions and real world workpieces in the robotic cell. Most of the limitations highlighted in the previous section are thus addressed in the following steps.

1. **CAD model generation.** Here it is assumed that a CAD model is available and can be loaded into the 3D modelling environment.
2. **Control points and tag creation.** This involves the definition of robot position tags from 3D CAD data with specific tool centre point. This paper proposes an interactive 3D method in which the user selects the path control

points directly on the model surface. For each selected control point, the solution will automatically generate the approach and retreat locations and the tooltip orientation.

3. **3D surface scanning.** A fast structured light 3D reconstruction method is used to scan workpiece surfaces that include the welding path. There is no need to scan the entire workpiece.
4. **3D registration with CAD models and error checking.** Automatic registration of CAD and scanned surface is performed based on point visibility constraints. The global and local root mean square errors (RMSE) between CAD model and their nearest points on the scanned surface give an indication to the user whether welding should proceed or not.
5. **Translation of control points from CAD to scanned surface.** If the global and local RMSEs are within set thresholds, each control point defined on the CAD model is translated to the scanned surface. This will guarantee that the welding path will be adapted to the actual workpiece thus, minimising path uncertainty.
6. **Trajectory planning.** The inverse kinematics of industrial robots usually yields multiple solutions in Cartesian space. Here, the control points together with derived approach, retreat and orientation information from step 2 are used to generate a unique solution. This step can easily be achieved by standard OLP software from the robot manufacturers. In the MARWIN project, this is achieved using the ABB RobotStudio which can deal with issues such as reachability, transitions, collision avoidance, and so on.

Steps 1 and 6 are outside the scope of this paper; in what follows we describe and demonstrate steps 2–5.

3 Control Points and Tag Creation

To demonstrate the concepts, an interactive modelling application named *3D Striper MARWIN* has been developed, whose interface is shown in Fig. 1 (left)

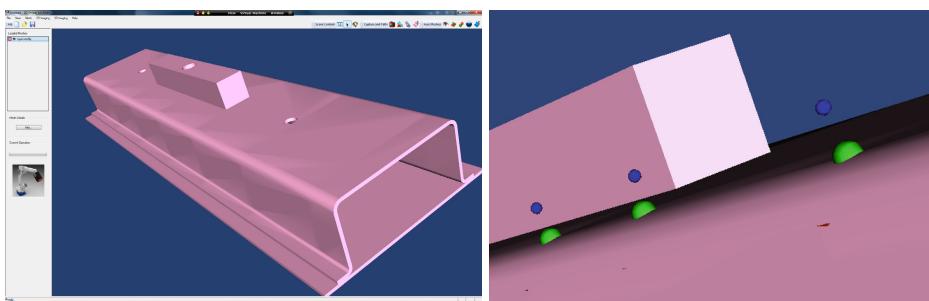


Fig. 1. Left: the 3D Striper MARWIN interface with a loaded CAD model. Right, welding control points selected through mouse clicks

with a loaded CAD model. In order to define a sequence of points for a welding path including approach points and tooltip orientation, an aligned bounding box with the (x, y, z) axes is estimated. This is simply defined by 6 bounds $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$. When the user marks a point on the surface of the model, a ray-tracing algorithm is used to find the intersection with the bounding box and the intersection with the mesh. This is a problem of line-plane intersection where it is assumed that the line has a starting point S and direction \mathbf{c} . The intersection line is given by

$$L(t) = S + \mathbf{c}t \quad (1)$$

The solution only involves finding the intersection point with the generic plane [12]. The generic plane is the xy -plane or $z = 0$. The line $S + \mathbf{c}t$ intersects the generic plane when $S_z + \mathbf{c}_z t_i = 0$ where t_i is t “intersection”:

$$t_i = -S_z / \mathbf{c}_z \quad (2)$$

From equation (2), the line hits the plane at point \mathbf{p}_i :

$$\mathbf{p}_i = S - \mathbf{c}(S_z / \mathbf{c}_z) \quad (3)$$

Thus, for every ray, i.e. for every point marked on the surface of the model by the user through a mouse click in Fig. 2 (right) two points are obtained \mathbf{p}_1 (blue sphere) and \mathbf{p}_2 (green). These points define the welding sequence where \mathbf{p}_1 is the intersection with the bounding box and \mathbf{p}_2 is the intersection with the mesh. Note that it is unlikely that the intersection on the mesh will rest on a vertex. Most likely, it will intersect on a polygon’s face somewhere between vertices. A good approximation then is to find the three vertices on the mesh that are the nearest to the intersection line. Such vertices define a plane and it then becomes straightforward to determine the exact intersection point through Equation (3).

The tooltip alignment is determined by Euler’s theorem [12]. Defining vectors \mathbf{u} and \mathbf{q} as

$$\mathbf{u} = \mathbf{p}_2^k - \mathbf{p}_1^k, \quad \mathbf{q} = \mathbf{p}_1^{k+1} - \mathbf{p}_1^k \quad (4)$$

where k is the index of each pair of points $(\mathbf{p}_1, \mathbf{p}_2)$. The desired alignment is that the tooltip x -axis is aligned with vector \mathbf{u} and the y -axis is aligned with \mathbf{q} . To perform the alignment, the required rotation is decomposed into a sequence of known steps:

1. Perform two rotations around the y and z axes by angles θ and ϕ so that the x -axis becomes aligned with \mathbf{u} .
2. Perform a z -roll of angle β around the newly rotated x -axis such that the y -axis becomes aligned with \mathbf{q} .

The above transformation requires the multiplication of 3 matrices:

$$\mathbf{R}_{\mathbf{u}}(\beta) = \mathbf{R}_y(-\theta)\mathbf{R}_z(\phi)\mathbf{R}_x(\beta) \quad (5)$$

Figure 2 shows the result of such alignment where the rotated x -axis is shown in yellow, the y -axis in red, and the z -axis in green. The approach (and retreat)

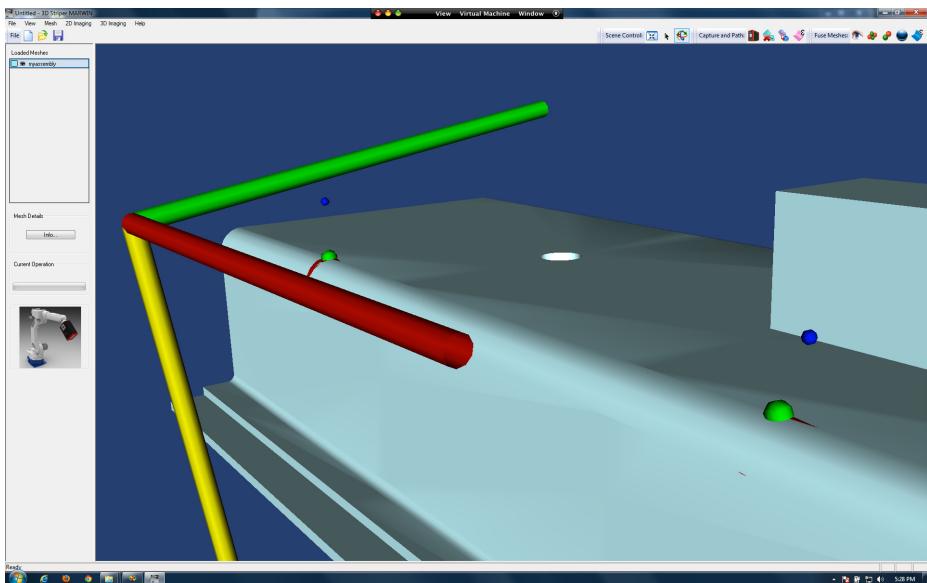


Fig. 2. Euler's theorem is used to determine the rotation matrix for tooltip alignment

position is \mathbf{p}_1 and the welding position is \mathbf{p}_2 which are achieved by a rotation around \mathbf{u} defined by equation (5) followed by a translation \mathbf{t} from \mathbf{p} (any position in the work space) to \mathbf{p}_1 and \mathbf{p}_2 :

$$\mathbf{p}_i = \mathbf{R}\mathbf{p} + \mathbf{t}, \quad \text{where } i = 1, 2. \quad (6)$$

The approach path from a generic position \mathbf{p} to \mathbf{p}_1 can be seen as a midpoint approach while from \mathbf{p}_1 to \mathbf{p}_2 is the final positioning of the tooltip.

4 3D Surface Scanning

The principle of operation of structure light scanning is to project patterns of light onto the target surface whose image is recorded by a camera [13]. The shape of the captured pattern is combined with the spatial relationship between the light source and the camera, to determine the 3D position of the surface along the pattern. The main advantages of the method are speed and accuracy; a surface can be scanned from a single 2D image and processed into 3D in 40ms [14, 15].

The expressions to compute the coordinates (x, y, z) of a surface point from a pixel location (v, h) on stripe n (mapping to a point on the surface of the scanned object) is defined as in [2, 11]:

$$x = D_p - \frac{D_p D_s}{vPD_p + W_n}, \quad y = \frac{hPD_p D_s}{vPD_p + W_n}, \quad z = \frac{W_n D_s}{vPD_p + W_n} \quad (7)$$

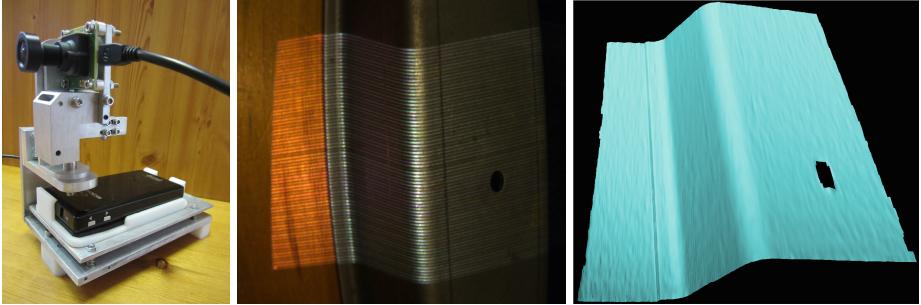


Fig. 3. Left: GMPR scanner; middle: 2D image; right: 3D reconstructed surface

where D_s is the constant vertical distance between the camera and the projector (Fig. 3, left), D_p is the constant distance between the projector and the system origin (or calibration plane), W is the constant width between successive light planes (or stripes) in the calibration plane, P is the pixel size in the sensor plane of the camera. The mathematical formulation of such arrangement is simpler than of those of standard scanners which results in less computing cycles, thus making the parallel design appropriate for 3D real-time processing.

Fig. 3 depicts the GMPR scanner developed by the MARWIN project which is capable of processing a single 2D image (Fig. 3 middle) into a surface patch (right). The scanner is attached to the robot tooltip such that each patch is incrementally registered until the desired workpiece surface is fully scanned. All scanned surfaces are thus, described in robot coordinate system.

5 3D Registration with CAD Models and Error Checking

In order to ensure correct calculation of robot trajectories based on the scanned surface, it is necessary to register the CAD model of the welding assembly to its scanned 3D model. This is to verify whether or not the scanned scene matches its CAD description and, if so, translate the control points from the CAD to the scanned surface. It is stressed that only translated points to the scanned surface will be used for trajectory calculation. The ICP (Iterative Closest Point) estimation algorithm [16] is used with the additional constraint of point visibility. The closest points in the ICP are found by calculating the Euclidean distances between a point \mathbf{p} in the first frame (the CAD model) and a point \mathbf{q} in the second frame (the scanned surface S) given by

$$d(\mathbf{p}, S_k) = \min_{j \in \{1, \dots, N_k\}} d(\mathbf{p}, \mathbf{q}_{k,j}) \quad (8)$$

Equation (8) means that every point in the CAD model needs to be checked against every point in the scanned surface. Once the closest points are estimated, the two sets of points \mathbf{p}_i and \mathbf{q}_i are paired to each other. The registration goal

is to estimate the parameters (\mathbf{R}, \mathbf{t}) rotation matrix and translation vector by minimising the following objective function:

$$F(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^m \sum_{j=1}^{N_i} p_{i,j} d^2(\mathbf{R}\mathbf{p}_{i,j} + \mathbf{t}, S_k) + \sum_{k=1}^n \sum_{l=1}^{N_k} q_{k,l} d^2(\mathbf{R}^T \mathbf{p}'_{k,l} - \mathbf{R}^T \mathbf{t}, S_i) \quad (9)$$

From the objective function in (9) the distance minimisation between the two sets of points is performed in a least squares sense:

$$f(\mathbf{R}, \mathbf{t}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{R}\mathbf{p}_i + \mathbf{t}, \mathbf{q}_i\|^2 \quad (10)$$

When the transformation model (\mathbf{R}, \mathbf{t}) has been estimated, transform every point in the CAD model. This iteration is repeated until convergence to a minimum set threshold or when a predefined number of iterations is reached.

The proposed visibility constraints are necessary for partial registration, as the ICP is guaranteed to fail if one tries to register both sets of data without adequate constraints. The method is that the user will pre-orient the CAD model such that its visible surface is in a similar orientation as the scanned surface, which is always looking down the workpiece from the robot's coordinate x -axis. Once this step is completed, the method proposed here is the removal of hidden

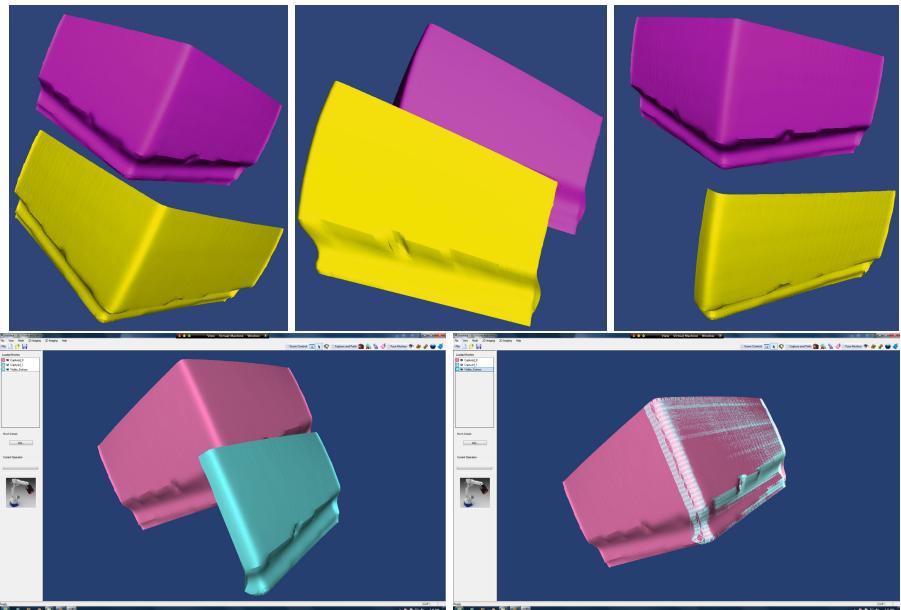


Fig. 4. Visibility constraints and registration. Top row (left and middle): model is re-oriented to display desired surface; (right): hidden surfaces are removed from a selected model. Bottom row (left): initial position; (right): after registration.

surfaces based on the depth buffer as depicted in Fig. 4 (top row). Each vertex $P = (P_x, P_y, P_z)$ of a face is available in the viewport as a scaled and shifted version of

$$(x, y, z) = \left(\frac{P_x}{-P_z}, \frac{P_y}{-P_z}, \frac{aP_z + b}{-P_z} \right) \quad (11)$$

The constants a and b are chosen such that the third component in (11) equals zero if P lies in the near plane and unity if P lies in the far plane. Successful registration is then performed with the visible and scanned surfaces, as shown in Fig. 4 (bottom row) through Equations (9) and (10).

6 Translation of Control Points from CAD to Scanned Surface

Upon registration convergence, each point \mathbf{p}_2 that was originally defined on the surface of the CAD model needs to be translated to the surface of the scanned model as depicted in Fig. 5. This is achieved by finding the intersection of the vector \mathbf{u} defined by Equation (4) with the scanned mesh using Equations (2) and (3).

In order to decide whether or not to proceed to the generation of the welding path, the root mean square error (RMSE) is calculated both globally and locally. The global RMSE considers all visible points and their nearest points on the scanned mesh. Defining \mathbf{p} a set of points in the CAD model and $\hat{\mathbf{p}}$ the nearest points in the scanned surface, the global RMSE is evaluated as:

$$\text{RMSE}(\hat{\mathbf{p}}) = \frac{1}{N} \sqrt{\sum_{k=1}^N (\hat{\mathbf{p}}_k - \mathbf{p}_k)^2} \quad (12)$$

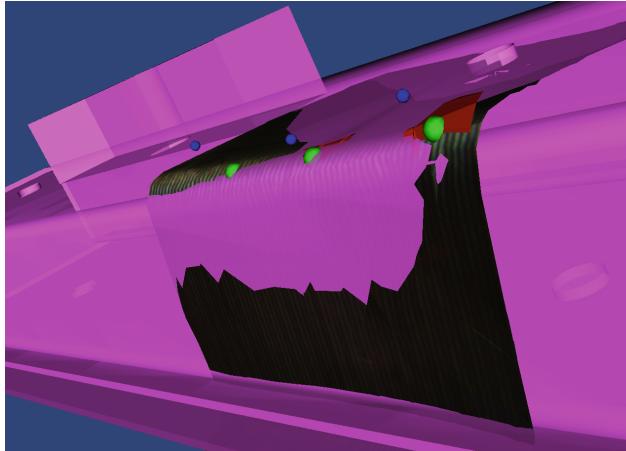


Fig. 5. Control points originally defined on the CAD model (magenta) are translated along the vector \mathbf{u} to the surface of the scanned model

where k is the number of points in the visible surface. The local RMSE only considers the control points lying on the path. Equation (12) is also used for local RMSE but in this case k is the index of the control points, \mathbf{p} is the set of control points in the CAD model and $\hat{\mathbf{p}}$ the translated points along vector \mathbf{u} .

If both global and local RMSE are smaller than a set minimum threshold, then the set of control points and approach points are saved to an XML file. This information will be used by the ABB RobotStudio to generate the robot trajectory as described in Step 6 of the methodology in Section 2. Otherwise, an error condition is flagged to the user who may need to adjust the workpiece(s), proceed to a new scanning, or re-estimate visibility.

7 Conclusion

This paper has proposed and demonstrated a new methodology for robot offline programming in welding tasks using a combination of 3D scanning and 3D model manipulation through an interactive interface. The method is based on defining control points on a CAD model that represent the welding path and then verifying whether or not the CAD model matches the actual workpiece through online scanning and registration. Registration is optimised by defining visibility constraints. Two measures of discrepancy between CAD and scanned surface are performed based on global error and local path error. If the scanned surface matches its CAD model within given error thresholds, then specification of the welding task can proceed.

The ability to automatically calculate robot trajectories and welding sequences directly from CAD models and then verifying these online through 3D reconstruction satisfies a principal aim of the MARWIN project which is to provide a user-centred, cognitive system for robotic welding tasks. Furthermore, the proposed methods address open issues in OLP concerning discrepancies between CAD and manufactured workpieces, inaccurate placement of workpieces in the robot cell, and inaccuracies between physical and modelled work cell.

While the main focus of the paper has been to address OLP methodology, the techniques developed here are also suitable for online programming of non-repetitive workpieces where a CAD model may not be available. In this case, the user would select the control points directly on the scanned surface and proceed to generate robot trajectories. The next stage of the MARWIN project is to integrate the developed methods and software routines onto an OTC robot control system whose results will be reported in the near future.

We acknowledge financial support from the EC under Grant Agreement no. 286284 Research for the Benefit of SMEs, MARWIN Project from 2011–2013.

References

- [1] European Commission Report: Smart, Sustainable Manufacturing: EU Research Sparks Revolution in Machine Tools, Robots and Automation. Research and Innovation Report, Brussels (2003)

- [2] Rodrigues, M., Kormann, M., Schuhler, C., Tomek, P.: An Intelligent Real Time 3D Vision System for Robotic Welding Tasks. In: IEEE 9th Int. Symposium on Mechatronics and its Applications (ISMA 2013), Jordan, Amman, April 9-11, pp. 1–6 (2013)
- [3] Pan, Z., Polden, J., Larkin, N., van Duin, S., Norrish, J.: Recent progress on programming methods for industrial robots. *Robotics and Computer Integrated Manufacturing* 28(2), 87–94 (2012)
- [4] Junior, M.H., Wei, L., Yong, L.S.: An industrial application of control of dynamic behaviour of robots – a walk-through programmed welding robot. In: Proc. 2000 IEEE Int. Conf. on Robotics and Automation, San Francisco, CA (April 2000)
- [5] Schraft, R.D., Meyer, C.: The need for an intuitive teaching method for small and medium enterprises. In: ISR-Robotik, Munich, Germany, May 15-18 (2006)
- [6] Pan, Z., Zhang, H.: Robotic programming for manufacturing industry. In: Proc. Int. Conf. on Mechanical Eng. and Mechanics, Wuxi, China, November 5-7 (2007)
- [7] Dai, W., Kampker, M.: User Oriented Integration of Sensor Operations in a Offline Programming System for Welding Robots. In: Proc. IEEE Conf. on Robotics and Automation, San Francisco, CA (April 2000)
- [8] Bi, A.M., Lang, S.Y.T.: A Framework for CAD- and Sensor-Based Robotic Coating Automation. *IEEE Transactions on Industrial Informatics* 3(1), 84–91 (2007)
- [9] Neto, P., Mendes, N., Pires, J.N.: CAD-Based Robot Programming: the role of Fuzzy-PI Force Control in Unstructured Environments. In: 6th IEEE Conf. on Automation Science and Engineering, Toronto, Canada, August 21-24 (2010)
- [10] MARWIN: Decision Making and Augmented Reality Support for Automatic Welding Installations, Project funded by the EC through Grant Agreement 286284. Research for the Benefit of SMEs (2013), <http://www.marwin-welding.eu/>
- [11] Rodrigues, M., Kormann, M., Schuhler, C., Tomek, P.: Structured Light Techniques for 3D Surface Reconstruction in Robotic Tasks. In: Burdak, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnerek, A. (eds.) CORES 2013. AISC, vol. 226, pp. 805–814. Springer, Heidelberg (2013)
- [12] Hill Jr., F.S.: *Computer Graphics Using OpenGL*, 2nd edn., 922 p. Prentice-Hall Inc. (2001)
- [13] Robinson, A., Alboul, L., Rodrigues, M.: Methods for Indexing Stripes in Uncoded Structured Light Scanning Systems. *Journal of WSCG* 12(1-3), ISSN 1213–6972
- [14] Rodrigues, M., Robinson, A.: Novel methods for real-time 3D facial recognition. In: Sarrafzadeh, M., Petratos, P. (eds.) Strategic Advantage of Computing Information Systems in Enterprise Management, Athens, Greece, pp. 169–180. ATINER (2010)
- [15] Rodrigues, M., Robinson, A.: Real-time 3D Face Recognition using Line Projection and Mesh Sampling. In: EG 3DOR 2011 - Eurographics 2011 Workshop on 3D Object Retrieval, Llandudno, UK, April 10, pp. 9–16 (2011)
- [16] Besl, P., McKay, N.: A method for Registration of 3-D Shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 14(2), 239–256 (1992)

Improving Accessibility of Virtual Worlds by Automatic Object Labeling

Ilias Apostolopoulos, Eelke Folmer, and George Bebis

Computer Science and Engineering

University of Nevada, Reno

{ilapost,bebis}@cse.unr.edu, eelke.folmer@gmail.com

Abstract. User generated virtual worlds, such as Second Life, typically lack accurate metadata for their virtual world objects. This is a significant problem for blind users who rely on textual descriptions in order to access virtual worlds using synthetic speech. In this paper, we consider the problem of automatic object labeling to improve accessibility of virtual worlds for users with disabilities. Taking advantage of the primitive-based representation of virtual world objects in Second Life, we present an approach that leverages histogram-based geometric object representations, machine learning and crowdsourcing to accurately label virtual world objects at a large scale. We report excellent classification results using seven challenging object classes.

1 Introduction

Though virtual worlds, such as Second Life, have somewhat waned in popularity over the past years, they still attract a respectable number of users and are still very profitable [2]. A number of accessible interfaces [8, 1, 3] have been developed that allow users with visual impairments to navigate their avatar and explore Second Life using audio, for example, by extracting textual descriptions that can be read with a screen reader [8]. Such interfaces require textual descriptions of objects to be present in order to create descriptions of the avatar’s environment.

Second Life’s content is entirely user generated. When users create new objects, they typically leave the object’s name to its default value. A study of Second Life content in 2009 revealed that nearly 32% of Second Life’s objects are called “object” [16]. This accessibility problem is similar to web images lacking the alt attribute tag. Over the past years, a number of techniques have been developed for labeling virtual world objects in order to make virtual worlds more accessible. This is a challenging problem since the same object can be created in completely different ways (i.e., using different number and type of primitives) by different users. These techniques typically employ some form of crowdsourcing (i.e., humans provide object descriptions).

Although object recognition is a task where average human performance outperforms machine learning, labeling millions of objects is a tedious and time consuming task when done manually. Unlike objects in real images where it may be difficult to segment due to background clutter, extracting virtual world

objects does not require explicit segmentation as they can be captured in isolation from the background. Virtual world objects are represented using geometric primitives which enables extracting geometric object representations efficiently and fast. In this paper, we present an approach that uses a small amount of crowdsourcing efforts to train a classifier that enables large-scale, real-time labeling of virtual world objects.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 describes the proposed approach. Our experimental evaluation and results are provided in Section 4. Section 5 discusses our results and provides directions of future work.

2 Related Work

Various techniques have been proposed for 3D object classification and retrieval. Many of them analyze 3D objects by considering a monolithic 3D model [13, 14, 15, 5, 10, 17]. These techniques rely on extracting shape features that are invariant to geometric transformations.

In [5], second-order 3D and spherical-kernel moment invariants are extracted from the density function in order to represent objects with a set of rotation-scale-translation invariant features. These features are used to find the closest match over a database of objects.

In [13], the creation of a "shape function" was proposed. First, randomly selected points are sampled on the surface of an object. These points are then used to compute geometric object properties such as the Euclidean distance of two random points, the square root of the area of the triangle formed by three random points, etc. These properties, which are called shape functions, are used to extract shape distributions, and compare them against each other to find out which one can better represent different classes of objects.

Using machine learning to automatically label 3D objects was proposed in [17] using features such as volume-surface ratio, moment invariants, and Fourier transform coefficients. A different approach considered the topological and skeletal object structure by utilizing multi-resolution Reeb graphs [10].

In [11], 3D objects are divided into smaller parts followed by classification. These parts are grouped into part classes. To classify an object, points are sampled on the query object and are compared to different parts, yielding a probability that this object belongs to a class.

In a related approach, instead of dividing the object into parts, the object is rendered in a volume using voxels [6]. Only the voxels inside the object are taken into consideration. Features are extracted from those voxels and stored in histograms. Support Vector Machines (SVM) are then used for training and classification.

A more recent work utilizes the bag of features technique [7]. Similarly to previous technique, the object is rendered in a volume using voxels. Points are sampled in the object and local patches are extracted based on the points. These patches are then represented as histograms which are used for classification.

3 Approach

We present an automated approach which consists of four main steps (see Fig. 1):

1. Collect objects using a spider.
2. Filter objects to remove noise and duplicates.
3. Extract useful features.
4. Train a classifier to distinguish among different object classes.

Object Collection. An automated agent/spider has been developed which can control an avatar in the virtual world, and can teleport automatically from region to region. In each region, the agent collects objects in that region and stores them in an XML file using their unique identifier as their name. By passing keywords as parameters, the agent collects specific objects we are interested in; for example, it can collect all objects that have a specific name as a substring in their name.

Filter Objects. The agent collects objects based on their name. For example, passing the keyword “table” results in a collection of objects whose name includes the string “table”. This means that the agent may collect objects that are both tables (e.g., “my table”) but also not tables (e.g. “table lamp”, “table cloth”). Moreover, it can collect mislabelled objects. To verify the correctness of the collected objects, we need to filter them by visually inspecting them.

Because this may involve large amounts of human effort, we use a crowdsourcing marketplace where human workers are paid to verify the correctness of a label given a picture of that object. Crowdsourcing can be done in a large scale at a low cost; a higher level of accuracy may be expected when involving a higher number of individuals for verifying the labels. Our agent can take the XML definition of an object as input and render it in the virtual world. We render objects in a white box to allow for a good contrast between the object and the background. For objects that are white, a different background color is used. The objects are scaled to fit and fill the box so as to make sure the picture contains the whole object. The agent takes a snapshot from each object at a certain distance. Then, the snapshot is saved as an image into a folder that is named after the object’s class.

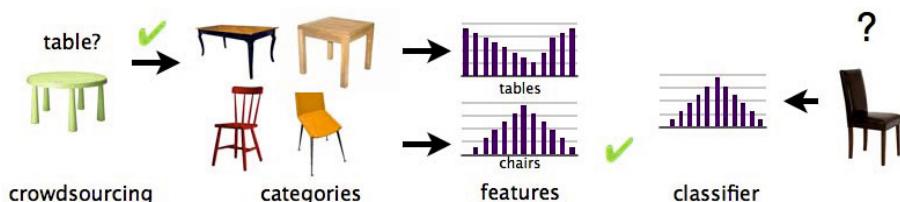


Fig. 1. A four step process is used for classifying objects: (1) objects are collected based on their class name; (2) crowdsourcing verifies the correctness of each object class name; (3) useful features are extracted to represent each object and (4) a classifier is trained to classify objects into known object classes

These pictures are used to create human intelligence tasks (HITs) at a crowd-sourcing marketplace, where workers are paid to confirm whether the label of an object matches its picture. The results of multiple HITs are compared to determine the accuracy of a given label for an object. Although this reduces the possibility of wrong labels, it does not completely eliminate objects having wrong labels. Therefore, when training a classifier, some training examples might have been labelled incorrectly.

Determine Useful Features. The previous step results in a set of objects with accurate labels that can be used to train a classifier. In Second Life, 3D objects are formed by 3D primitives or prims (e.g., boxes, spheres etc.) instead of meshes. Various features can be extracted from each primitive. Here, we have experimented with the type of prims, their size, orientation, and eccentricity.

Prim type describes the type of the primitive (e.g., box, sphere, cylinder, etc.). Second Life contains eight main prim types. These prim types are box, sphere, cylinder, prism, torus, tube, ring, and unknown. It should be mentioned that besides the main prim types, there is a special prim type called "sculpted" prim whose shape is determined by an array of x, y, z coordinates stored as RGB values in an image file (i.e. texture or map). Since the properties of each sculpted prim are mainly defined by embedded textures rather than parameters which are common in all other types, the encapsulated information may be misleading for a learning agent. Therefore, objects with sculpted prims were not included in our data set.

Size refers to the volume of the bounding box of a primitive. For scale invariance, we represent it as the ratio of the volume of the primitive to the total volume of the object (i.e., sum of prim volumes). Orientation is defined as the relative angle between the primitive's orientation and the object's average orientation; this is also rotation invariant. We estimate the average orientation of an object by the orientation of its largest dimension. Finally, eccentricity is defined in scale invariant way as the ratio of the smallest to the largest dimension of the primitive.

Besides these features, several other features were explored. Some prims in Second Life, for example, have taper and/or twist. Taper thins out either the top or the bottom side of prim while twist twists either the top or the bottom side of the object. However, after running several experiments using these features, our results indicated that they were not helpful in distinguishing between object classes. In addition, color was taken into consideration but was discarded quickly as in virtual worlds, objects typically have a wide range of colors.

To take advantage of the prim-based representation of objects in Second Life, we have adopted a bag-of-features approach [4] which has demonstrated good success in computer vision. In the bag-of-features approach, objects are represented as order-less collections of local features using histograms. The main idea is quantifying the number of occurrences of local features in an object. Since the number of local features might be very large, a visual vocabulary is typically built by clustering the local features. Features belonging to the same cluster are represented by the center of that cluster which is referred to as "visual" word.

Therefore, bag-of-features represent objects as histograms of visual words drawn from a visual vocabulary which is built from local features.

When using type as a feature, each prim type is considered as a visual word. In this case, each object is represented as a histogram which encodes the relative frequency of occurrence of each prim type in an object. When using any of other of the three features (i.e., size, orientation, eccentricity), the visual words are defined by dividing the range of values that each feature can assume into a number of bins and associating the center of each bin with a visual word. In this case, each object is represented as a histogram which encodes the relative frequency of occurrence of various feature values in the object.

It should be mentioned that building the visual vocabulary for each feature is fast and easy. In general, the bag-of-features approach is sensitive to background clutter since it cannot distinguish between objects and background. This is not an issue in Second Life since objects in virtual worlds are defined independently of other objects.

Classifier Training. The histograms described in the previous section are used for training the object classifier. From our experiments, we have found that using each feature separately does not provide enough discrimination. However, more powerful object representations can be built by considering combinations of features. For this reason, we have investigated combinations of features by building joint histograms where each dimension corresponds to a different feature. Joint histograms provide more discriminatory information, however, they are memory and time consuming. Moreover, they tend to be quite sparse which implies that some kind of dimensionality reduction is necessary to eliminate redundancy and improve classification time and accuracy.

To address the issues above, we have adopted Linear Discriminant Analysis (LDA) [12]. LDA is a powerful dimensionality reduction technique which projects the data into an appropriate space where inter-class variability is maximized while intra-class variability is minimized. The reduced dimensionality data is used to train a Support Vector Machine (SVMs) classifier [9]. SVMs are supervised classifiers that have been shown to be an attractive and more systematic approach to learning linear or nonlinear decision boundaries. Given a set of points and assuming two classes, SVM finds the hyper-plane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyper-plane. This is equivalent to performing structural risk minimization. Here, we use a multi-class SVM which is a generalization of the two-class SVM classifier.

4 Experiments

We have validated our technique experimentally using objects extracted from Second Life.

Collect Objects. An agent for Second Life was developed in Windows using the LibOpenMetaVerse library. For our experiments, we had the agent collect

Table 1. The first line indicates the number of items collected from Second Life. Consecutive lines represent the number of items that were filtered out and the last line shows the final number of remaining items.

	Buildings	Cars	Chairs	Lamps	Plants	Tables	Trees
Items Collected	2880	821	3144	2170	2658	1207	3317
1-prim items	-1639	-374	-14	-666	-1608	-147	-2179
2-prim items	-185	-46	-22	-247	-269	-117	-223
Duplicate items	-284	-104	-58	-548	-520	-47	-593
Items with sculptures	-4	-44	-27	-57	-62	-11	-139
AMT	-329	-165	-2619	-115	-59	-153	-50
Final # of items	439	88	404	537	140	732	133

objects for the following seven object classes: “building”, “car”, “chair”, “lamp”, “plant”, “table”, “tree”. Table 1 shows the number of objects collected for each class. These classes were chosen because a the plethora of examples available for each one of them in Second Life. It should be noted that some object classes are structurally very similar (e.g., see Figure 2), making it challenging to define appropriate features for accurate classification.

Filter Objects. Before training, the objects collected are filtered through various stages. First, objects with one or two primitives are removed because they do not contain enough information for classification purposes. Then, we filter objects using Amazon Mechanical Turk (AMT) which is a popular marketplace for HITs. Pictures were created for each object and used to create HITs in AMT. Workers were asked to confirm a given label for 10 images by checking a checkbox under each image. Seven to nine out of ten images used in each HIT were taken out of our collected items and the remaining one to three were taken from another set of images of virtual people (avatars) and places. Noise was introduced to be able to detect and filter out the entries submitted by unreliable workers, who may just label all or no objects. Figure 3 shows an example HIT.

The remaining objects are further filtered in order to remove potential duplicates.

Classification. To investigate the importance of various feature combinations, we have experimented with joint feature histograms where each dimension

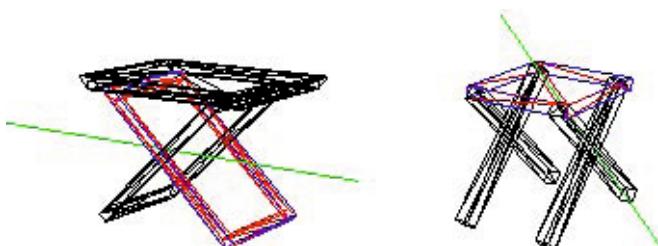


Fig. 2. An example of a table and a chair that look very similar



Fig. 3. A sample HIT with 7 images of tables and 3 noisy images

Table 2. Examples of joint histogram sizes before and after excluding empty bins

	number of bins before	number of bins after
2D	4500	2257
3D	90000	5528
4D	720000	7537

corresponds to a different feature. The entries of joint histograms are determined by computing the number of occurrences of combinations of feature values in an object.

When considering three or four features together, the corresponding histograms might have very large size which becomes problematic when applying LDA. Since most entries in joint histograms are empty, we only keep track of non empty bins. By taking the union of non-empty bins across all objects in our dataset, we determine which bins are most important (i.e., contain non-zero entries). These bins are then concatenated into a one dimensional histogram of fixed length which is used to represent objects in our dataset more efficiently. This process reduces the size of histograms significantly as shown in Table 2, without omitting any useful information.

The concatenated 1D histograms are then provided to LDA which determines the most discriminatory features. The results of LDA are used to train the SVM classifier To test the accuracy of the classifier, we used a five-fold cross-validation procedure. In five-fold cross-validation, the data is randomly divided into five mutually exclusive partitions of equal size. Then, one of this partitions is chosen for testing the classifier while the other four are used for training the classifier. This process is repeated five times, each time using a different partition for testing and the other four partitions for training. We report the average five-fold cross validation error.

Parameters. An important parameter when computing the histograms is the number of histogram bins. To see how this parameters affects classification performance, we performed experiments by varying the number of bins from ten to hundred with a step of five (except for type where the number of bins was equal to the number of prim types). Figure 4 shows our results in the case of size, orientation, and eccentricity. As it can be observed, best results were

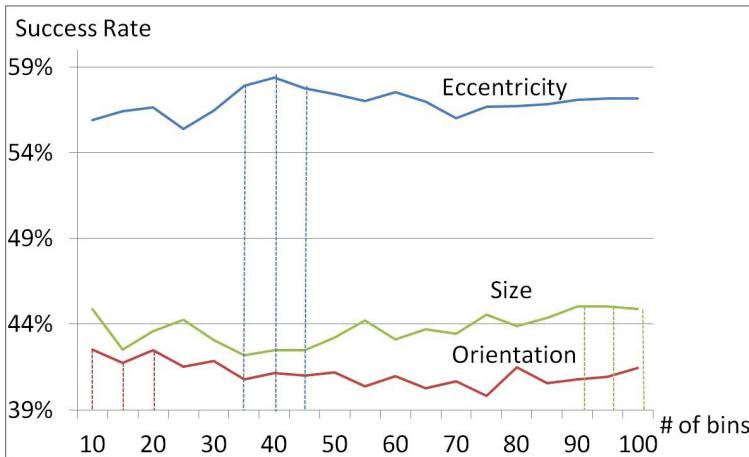


Fig. 4. Success rate of different features by varying the number of histogram bins

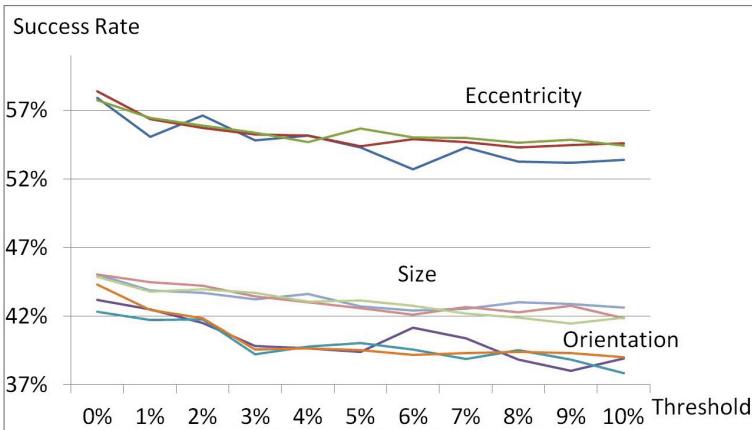
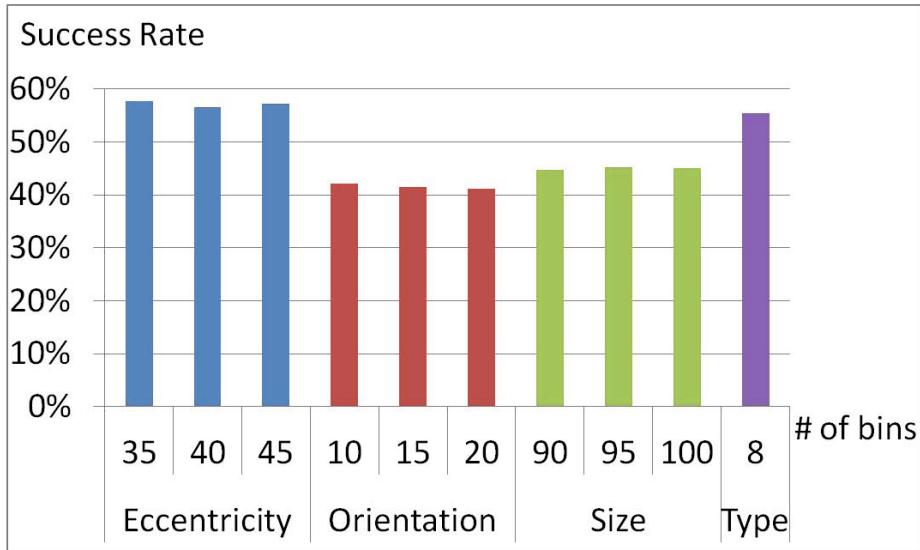


Fig. 5. Graphs of various features' success rate based on the amount of threshold

obtained using 90-100 bins for size, 10-20 bins for orientation, and 35-45 bins for eccentricity.

We also considered removing primitives having very small size as they might introduce mostly noise. Specifically, prims that were smaller than a percentage of the whole object were removed. Various levels of thresholding were considered starting from 0% and going as high as 10% in increments of 1%. For each feature, we experimented with all three optimum number of bins found in the previous experiment. Our experimental results, shown in Figure 5, indicate that thresholding does not improve classification accuracy.

**Fig. 6.** Results for 1D histograms

Results. In this section, we report our results using 1D histograms (i.e., using each feature separately), 2D histograms (i.e., using pairs of features) and 3D histograms (i.e., using triplets of features). For each feature, we have considered the optimum number of bins found in the previous subsection.

Fig. 6 shows our results using 1D histograms; as expected, performance is rather poor indicating that more information is needed for discriminating among the object classes. Best results (i.e., 57.67%) were obtained using eccentricity.

Next, we investigated 2D histograms; Tables 3,4,5,6, show different combinations with performance being for most combinations between 55% to 65%. While

Table 3. Results for 2D histograms Type-Size, Type-Orientation, and Type-Eccentricity

Type-Size			Type-Orientation			Type-Eccentricity		
90	95	100	10	15	20	35	40	45
64.15%	65.01%	64.89%	54.77%	54.65%	54.40%	61.37%	64.73%	64.48%

Table 4. Results for 2D histogram Size-Orientation

	10	15	20
90	56.21%	62.27%	65.92%
95	57.27%	63.05%	67.39%
100	58.38%	64.07%	68.05%

Table 5. Results for 2D histogram Size-Eccentricity

	35	40	45
90	94.35%	95.58%	97.05%
95	95.62%	96.27%	98.07%
100	96.31%	97.17%	97.95%

Table 6. Results for 2D histogram Orientation-Eccentricity

	35	40	45
10	59.12%	62.02%	62.97%
15	62.39%	63.50%	65.87%
20	66.00%	68.13%	69.32%

these results are better than the results obtained using 1D histograms, they are still not satisfactory. However, the combination of size with eccentricity yields a much higher performance between 94% and 98%. This implies that combining size and eccentricity provides high discriminatory information for the 7 object classes.

Our results using 3D histograms are shown in Tables 7,8,9 ,10. As it can be observed, the combination of size, orientation, and eccentricity yields the best results, reaching a classification accuracy as high as 99.96%. These results are consistent with the results obtained using 2D histograms; the addition of an extra feature to the size-eccentricity combination has improved performance.

Table 7. Results for 3D histogram Type-Size-Orientation

	10	15	20
90	83.24%	89.27%	90.17%
95	83.04%	88.28%	90.76%
100	84.76%	89.88%	91.48%

Table 8. Results for 3D histogram Type-Size-Eccentricity

	35	40	45
90	98.77%	99.34%	99.55%
95	99.22%	99.39%	99.63%
100	99.30%	99.59%	99.63%

Table 9. Results for 3D histogram Type-Orientation-Eccentricity

	35	40	45
10	78.86%	80.87%	82.18%
15	83.78%	85.21%	86.36%
20	85.91%	87.75%	88.73%

Table 10. Results for 3D histogram Size-Orientation-Eccentricity

	10-35	10-40	10-45	15-35	15-40	15-45	20-35	20-40	20-45
90	99.75%	99.88%	99.92%	99.84%	99.88%	99.92%	99.84%	99.88%	99.92%
95	99.71%	99.88%	99.92%	99.88%	99.88%	99.92%	99.88%	99.88%	99.92%
100	99.88%	99.92%	99.96%	99.88%	99.92%	99.96%	99.92%	99.92%	99.96%

While it is possible to consider 4D histogram (i.e., combine all four features), our results using 3D histograms are already very satisfactory. Therefore, the extra computational time needed to compute 4D histograms would not be justified by a possible small increase in performance.

Once training has been completed, a classifier using size-orientation-eccentricity takes on average 2.4 seconds to classify a 3D object.

5 Conclusion

We have presented a new approach for automatically classifying virtual 3D objects in Second Life. Our experimental results show that it is possible to obtain very high classification accuracy using 3D histograms based on size, orientation, and eccentricity. Our technique can be integrated in existing accessible virtual world clients [8, 1, 3]. When a user encounters an object lacking a name, our technique may allow for recognizing it in real time.

It is worth mentioning that our training set might contain wrong labels and that there is a tradeoff between cost and accuracy when filtering objects using crowdsourcing. More accurate object labels can be obtained by choosing workers with higher acceptance rates, injecting more noise, having more strict agreement rules, decreasing the number of images in a HIT and increasing payment. However, each of these strategies will increase the associated costs. Especially when having to filter large numbers of objects, cost may become a serious consideration. Future work will research whether it is still possible to reach acceptable recognition rates using fewer examples.

For future work, we plan to build a hierarchical classification system in order to assign objects to more abstract categories (e.g., furniture, vehicles). It would be interesting to use crowdsourcing again in order to have workers verify whether objects with a given label (*cat*) belong to a certain category (*animal*). This can help us to establish rules for a taxonomy *animal* \leftarrow *cat* that the classifier could use. This approach is similar to Yuan [16], with the difference that it could be performed at a much larger scale by taking advantage of crowdsourcing marketplaces, such as AMT.

Acknowledgements. This work is supported by NSF Grant IIS-0917362.

References

- [1] Ibm human ability and accessibility center, virtual worlds user interface for the blind, http://www-03.ibm.com/able/accessibility_research_projects/virtual_worlds_accessible_UI.html (access April 5, 2009)

- [2] Linden lab's second life 'extremely profitable,' company looking to expand, <http://massively.joystiq.com/2012/03/15/linden-labs-second-life-extremely-profitable-company-looking/> (access August 18, 2012)
- [3] Virtual guide dog project, <http://virtualguidedog.org> (access March 4, 2009)
- [4] Csurka, G., Dance, C., Willamowski, J., Fan, L., Bray, C.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision (2004)
- [5] Cybenko, G., Bhasin, A., Cohen, K.D.: Pattern recognition of 3D cad objects: Towards an electronic yellow pages of mechanical parts. Int. J. Smart Eng. Syst. Design 1(1), 1–13 (1997)
- [6] Fehr, J., Burkhardt, H.: Harmonic shape histograms for 3D shape classification and retrieval. In: IAPR Conference on Machine Vision Applications, pp. 384–387 (2007)
- [7] Fehr, J., Streicher, A., Burkhardt, H.: A *bag of features* approach for 3D shape retrieval. In: Bebis, G., et al. (eds.) ISVC 2009, Part I. LNCS, vol. 5875, pp. 34–43. Springer, Heidelberg (2009)
- [8] Folmer, E., Yuan, B., Carr, D., Saps, M.: Textsl: a command-based virtual world interface for the visually impaired. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2009), Pittsburgh, Pennsylvania, USA, pp. 59–66 (2009)
- [9] Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13(4), 18–28 (1998)
- [10] Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 203–212. ACM (2001)
- [11] Huber, D., Kapuria, A., Donamukkala, R., Hebert, M.: Parts-based 3D object classification. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, pp. 82–89. IEEE Computer Society, Washington, DC (2004)
- [12] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mullers, K.R.: Fisher discriminant analysis with kernels. In: Proceedings of the 1999 IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing IX, pp. 41–48. IEEE (1999)
- [13] Chazelle, B., Osada, R., Funkhouser, T., Dobkin, D.: Matching 3D models with shape distributions. In: Shape Modeling and Applications, pp. 154–166 (2001)
- [14] Shilane, P., Funkhouser, T.: Selecting distinctive 3D shape descriptors for similarity retrieval. In: IEEE International Conference on Shape Modeling and Applications, SMI 2006, p. 18. IEEE (2006)
- [15] Vranić, D.V.: 3D model retrieval. University of Leipzig, Germany. PhD thesis (2004)
- [16] Yuan, B., Saps, M., Folmer, E.: Seek-n-tag: a game for labeling and classifying virtual world objects. In: Proceedings of Graphics Interface, GI 2010, Ottawa, Ontario, Canada, pp. 201–208 (2010)
- [17] Zhang, C., Chen, T.: Indexing and retrieval of 3D models aided by active learning. In: Proceedings of the Ninth ACM International Conference on Multimedia, pp. 615–616. ACM (2001)

Hierarchical Image Geo-location on a World-Wide Scale

Alexandru N. Vasile¹ and Octavia Camps²

¹ Massachusetts Institute of Technology - Lincoln Laboratory, Lexington, MA, USA

² Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

alexv@ll.mit.edu, camps@ece.neu.edu

Abstract. There are increasingly large amounts of imagery and video collected from a variety of sensor modalities. Considering that each individual image may contain considerable amounts of information, the ability to interpret, understand and extract scene information is highly beneficial for many communities such as online social networking sites, intelligence agencies and companies dealing with large-scale data mining. In order to enable automated scene understanding, there is a need for an organizing principle to store, visualize and exploit the data. Three-dimensional geometry provides such an organizing principle as imagery and video have inherent 3D structure and can be associated with geographic coordinates. Imagery with geo-spatial information can be used to develop a common 3D world model representation that integrates data across a wide variety of sensor modalities. In this paper, we leverage multiple large geo-spatial databases to create a 3D world model and develop a hierarchical image geo-location framework using a coarse-to-fine approach to geo-locate a query image. Starting at the coarsest level, we developed a novel method to geo-locate images to regions of the world through a process of terrain classification. Next, we developed novel medium-scale and fine-scale localization steps to rule out most of the coarsely geo-located regions and determine candidate geo-locations with geo-positioning accuracy at a city level. Our method was demonstrated on a 6.5 million image database and shown to improve on current state of the art in the areas of both terrain classification and image geo-location.

1 Introduction

In the last decade, there has been an explosion in the amount of digital imagery and video. Vast numbers of photos, shot by inexpensive digital cameras, can now be accessed via the web, using online databases such as Flickr. But, even though these huge imagery data sources are now accessible, they are usually unstructured and not well organized. For example, searching on Flickr requires the user to type in a text search query and to manually search through lots of unorganized thumbnails, which can be a frustrating experience. Some organizing principle is needed to enable efficient navigation, understanding and exploitation of these large imagery archives.

Fortunately, three-dimensional geometry provides such an organizing principle for imagery collected at different times, places and perspectives. For example, suppose we have a set of photos collected in a city. Those pictures represent 2D projections of

3D world-space onto a variety of image planes. If the target's geometry is captured in a 3D map, it can be used to mathematically relate the different ground photos to each other. Moreover, the 3D map connects together data collected by completely different sensors at different times. So we can relate a photo of a city shot by a ground camera with a corresponding aerial Radar point cloud or satellite image. This can add a lot of context to a scene, and improve scene understanding. But, we can only get this improvement in scene understanding provided all these data products are geo-located with the 3D map. Figure 1 captures this common 3D world model representation.

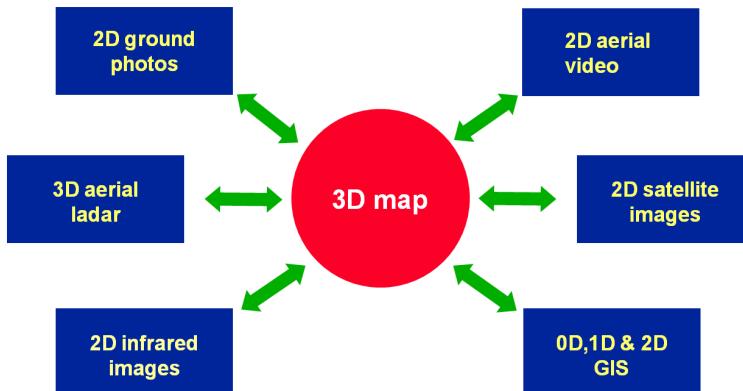


Fig. 1. 3D world-model representation for organizing data from multiple sensing modalities. The 3D map provides a geometrical framework for organizing imagery collected at different times, places and perspectives, enabling improved image context and scene understanding

In order to initialize this 3D world model representation, we need to start out with enough imagery that not only has geo-spatial metadata but also samples well the entire world (i.e. millions of images). Fortunately, there is a wealth of imagery available online that has geo-spatial metadata. For example, Flickr has enabled geo-tagging of images since August 2006; within the first day, 1.2 million images were geo-tagged. However, most images on the web usually do not have geo-spatial metadata available, leaving the task up to a human operator to manually annotate the data, which can often be tedious or impractical. In the absence of meta-data, we need to rely on scene content to deduce geo-spatial information. Depending on the scene content, such as how many features we might be able to extract and how salient those features are, we can geo-locate an image to various levels of geo-spatial accuracy, such as to a particular continent, region, city, street or even actual camera pose.

The problem of image geo-location has been addressed by several authors in the past. Most of the geo-location research falls into two categories: 1. localization by landmark recognition using local image features and 2. localization by similar image retrieval using global features that capture whole image content. Geo-location by landmark recognition [1][2][3][4] tends to focus on limited image datasets (100s of thousands) that are already highly localized around a set of landmarks or comprised of images from at most one city. Most of these methods apply feature matching and structure-from-motion techniques to estimate camera location and pose. For the

problem of geo-location on a world-wide scale using million of images, direct localization by landmark recognition is not computationally tractable. Another drawback of the above methods is that they require extremely dense sampling of the world, with at least one or two instances that have the exact scene content as the query image; in currently available image databases with world-wide coverage, most locations do not have such dense image sampling, leading to poor localization performance.

The second research category, of image geo-location by similar image retrieval, holds more promise for geo-location on a world-wide scale. Seminal work by Hayes et al. [5][6] demonstrated the feasibility and potential for image localization on a world wide scale. The method applied a single-stage unsupervised algorithm on a multi-million image world-wide database to directly geo-locate a query image to a set of likely locations in the world. One of the drawbacks of the method was the use of a single stage classifier, resulting in the need for both a high-dimensional feature space to separate highly complex classes and the need to use an unsupervised classification method for computational efficiency. Applying an unsupervised classification method in high dimensions is not ideal as such methods are known to suffer as feature dimensionality increase due to their inability to discard irrelevant feature dimensions for a given task [6].

An improvement to the methods in [5],[6] is to use a hierarchical image geo-location approach. From an algorithmic perspective, developing a hierarchical geo-location framework has several advantages. Rather than resorting to the use of a high-dimensional feature space to separate highly complex geographic classes in one step, by implementing multiple hierarchical stages we can solve multiple simpler classification problems, each in a lower dimensional feature space to avoid the curse of dimensionality [7]. Furthermore, a hierarchical approach has the potential for improved geo-location accuracy by allowing for both the use of simple, unsupervised classifiers for the initial stages followed by ever more complex classifiers for the later stages. Establishing such a hierarchical framework also makes sense from a computational point of view. Because several models relating to specific locations can exist, comparing all models over the vast space of all possible images may not be computationally feasible. Paring down the search space using coarse geo-location models with rough spatial descriptors on large databases, followed by increasingly complex descriptors applied on reduced-size databases makes the geo-location problem much more computationally tractable.

From a classification standpoint, the problem of geo-locating an image with no geo-spatial metadata to a city-sized geographic class is very challenging. There are many thousands such city-sized geographic classes in the world that we need to separate. Besides the sheer number of classes, the boundary between geographic classes (e.g. is this Bangkok or Paris?) is extremely complex because it must divide a spectrum of scene types (indoors, outdoors, close-up, perspective, street, highway, tall and short buildings) that might be present in both locations.

In order to overcome this complex classification problem, in this paper we propose a novel hierarchical image classification approach that geo-locates a query image of an urban scene to a particular city location in the world. As shown in Figure 2, we start out with a query image and a 3D world model representation composed of several large databases, namely the 6.5 million image geo-spatial image database from

[5][6], a world-wide land-coverage and terrain type database and a terrain-labeled image database. At the coarse scale, we consider a query image as a whole by extracting rough scene content to assign the image to a land class type, such as *urban*, *forest*, *coast*, *country* or *mountain*. Once a terrain label is obtained, such as ‘*urban*’ for instance, we can reduce the image and geo-spatial search space by filtering the larger database for images with geo-tags in close proximity to urban areas. This has the effect of reducing the geo-spatial and database search space anywhere from 70 to 90%. For the medium-scale geo-location method, additional image content is extracted through the use of multiple low-level features per image. Those features are matched against a pre-computed feature database using a novel supervised classifier to reduce the geographical search space on the order of 99%, with multi-candidate locations of geo-location accuracy at the region to country level. Results from the medium-scale classifier are sent to a fine-scale classifier to obtain geo-location accuracy at city level. The key contributions of this paper are:

1. The development of a new geo-tagged and terrain-labeled large-scale image database to represent the 3D world model and its application to a novel coarse geo-location method, with terrain classification results that are an improvement of 6% over previously reported results.
2. A medium and fine-scale geo-location method that improves upon previous image retrieval techniques to geo-locate a query image to city-level accuracy.
3. A hierarchical geo-location framework that is demonstrated to have improved geo-location accuracy. The algorithm was tested on a geo-tagged 6.5 million image database and demonstrated to have a relative improvement of 20% in geo-location accuracy compared to previous methods.

The paper is organized as follows: Section 2 discusses the coarse-scale geo-location method and results; Section 3 describes the medium and fine scale geo-location algorithms and results, while Section 4 concludes with a discussion.

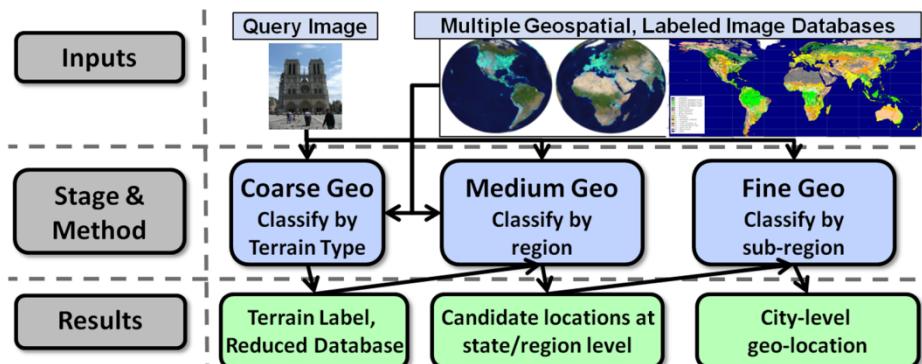


Fig. 2. Proposed hierarchical geo-location approach. The method starts out with a query image and a 3D world model representation composed of several geo-spatial databases with millions of data samples. The coarse-scale geo-location method applies a computationally efficient terrain classifier to the query image in order to reduce the search space. On the resulting reduced-size database, the medium and fine scale geo-location methods apply more complex classifiers in order to obtain improved geo-location accuracy, with localization at the city scale

2 Coarse-Scale Geo-location

We present the coarse geo-location approach where we geo-locate a query image to a particular region of the world by classifying the terrain type in that image based on image content. To achieve the goal of image geo-location by terrain classification, we first create a 3D world model representation composed of a large training database of geo-tagged, terrain labeled images. This database is created by merging knowledge from three publically available databases, namely a geospatial terrain type and land coverage database, a 6.5 million image database that is only geo-tagged and a database of terrain-labeled images. We develop a coarse geo-location method that uses the generated 3D world model to test a hold-out set of 5000 images and demonstrate an improvement over current state of the art in terrain classification, with over 91% terrain classification accuracy. The resulting terrain label for the image is used to reduce geographical search space and segment the original large database by filtering for images with geo-tags in close proximity to the resulting terrain-label. The reduction in search space allows the usage of more complex medium-scale and fine-scale geo-location classifiers that are both accurate and computationally tractable.

In the following section, we briefly talk about related work on image retrieval and how it applies to the problem of image geo-location and terrain classification.

2.1 Background and Related Work

There has been a significant interest in applying machine learning methods to scene classification and image retrieval. Most of these methods use feature vectors such as SIFT [8], texton dictionaries [9], color histograms, Gist [10] or a combination of these [6], with feature vector dimensions on the order of 100-1000. When the training database is on the order of tens of thousands of examples, the most common and reliable method to train is using Kernel Support Vector Machines (SVMs). However, the problem of image geo-location on a world-wide scale requires much larger image training databases, on the order of millions of images, to have enough sampling of the various locations around the world.

For computational scalability, retrieval methods that use millions of training samples typically use the K-nearest neighbor (KNN) training algorithm in a feature space defined by a few image feature types and then use those nearest neighbors for various tasks, such as object recognition [11][12][13][14], image completion [6] and object classification. Nearest neighbor techniques are attractive in that they are trivially parallelizable, require no training, have good classification performance and perform well from a computational perspective with query complexity that scales linearly with the size of the data set. Nearest neighbor methods rely on a feature vector of low to medium size in dimensions (100-200), either using SIFT [8], texton vocabularies [9], Gist [10] or a combination of these [6]. Given a new image, the same feature vector is computed and the nearest neighbors in feature space are found from the training database. Using those neighbors, a majority rule is implemented to determine the label for the new query image. These KNN methods tend to work well in low to medium sized dimensional spaces, but tend to suffer as feature dimensionality increases [6][7].

The reason for this is that nearest neighbor methods lack one of the fundamental advantages of supervised learning methods, which is the ability to discard irrelevant feature dimensions for a given task [6].

In the context of image geo-location, the KNN algorithm has been used in seminal work by [6] as part of a single stage algorithm that extracted a single feature vector per image from a 6.5 million image database. To separate the highly complex city-sized classes present in image geo-location, the feature vectors were high dimensional, of size close to 3000 dimensions. Given a query image and its associated feature vector, KNN was applied in this high-dimensional space to retrieve the k-nearest images, thereby directly obtaining the k most likely candidate geo-locations for the query image [6]. As noted beforehand, KNN tends to suffer as feature dimensionality increases, so working in a 3000 dimensional feature space is not ideal. Furthermore, [6] only used data from a geo-tagged image database and did not use additional knowledge to penalize unlikely matches. In the next section, we present a novel coarse-scale geo-location method that addresses some of these shortcomings.

2.2 Coarse-Scale Geo-location Method

The coarse geo-location method builds upon research on image terrain classification from [15] as well as image geo-location research from [6]. Our method uses the same image database from [6], but improves on the geo-location approach by not only avoiding the problem of KNN in high dimensions but also using additional data sources to enrich our 3D world model representation in order to penalize unlikely matches. Starting with the 6.5 million image geo-tagged database from [6], we develop a method to probabilistically annotate terrain labels to each of the images by combining knowledge from two additional databases: a world-wide land-coverage and terrain-type geospatial database and 2689 image terrain-labeled truth database from [15]. By adding these new data sources, we are now able to penalize unlikely matches that might otherwise happen with an image database that only has geo-tags. For instance, the correct recognition of a coastal image as being a coastal scene would make the image a highly unlikely match to an image with a geo-tag from an inland area.

We create this probabilistically labeled geo-tagged/terrain data set in a two-stage process. In the first stage, we use the geo-tags along with our world-wide land coverage geo-labeled database to weakly label our images as belonging to a subset of 5 terrain classes. In essence, we are using the geo-spatial metadata embedded with the image to determine a terrain label prior. We classify the world using 5 terrain types, namely: *coast*, *country*, *forest*, *mountain*, and *urban*. We primarily use the USGS GLCC Database [16] to assign a subset of labels to each 1x1km land tile. From this data base, we determine layers for urban, forest, country and coast. We apply a 1km dilation operation for urban, forest and country label regions, and a 3km dilation operation for coastal regions that are derived from sea-land contour lines. Since the USGS GLCC Database does not contain labels for our mountain regions class, we extract this information from UNEP, Mountains and Tree cover in Mountain Regions 2002 Database [17]. Figure 3 shows some examples of the data from the databases used to create the 3D world model.

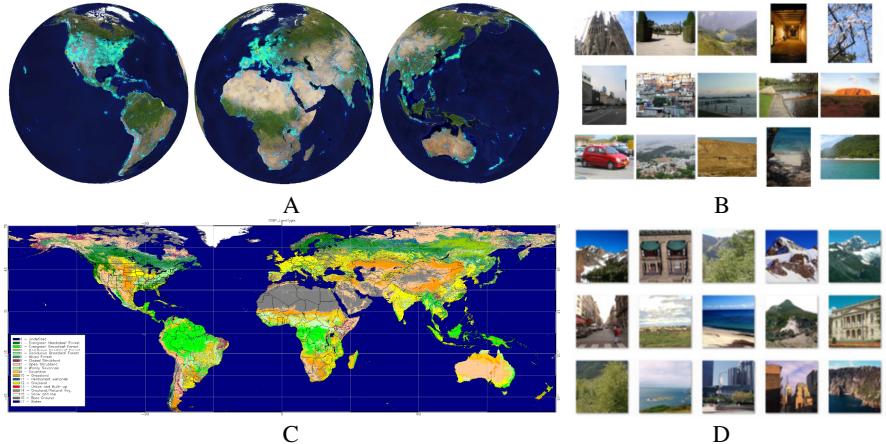


Fig. 3. Databases used to create the geo-tagged, terrain labeled 3D world model representation. A) Distribution of the 6.5 million geo-tagged image database across the world, color-coded by density using the ‘jet’ color-map (cyan low density, yellow, red represent higher density). B) Example of images in the database. Images are from indoors, outdoors, close-up and perspective and do not only contain landmarks, but also more ordinary pictures of streets, cars, etc. C) USGS GLCC terrain land coverage database used to extract terrain labels priors conditioned on geo-spatial position. D) Terrain-labeled truth database from [15] with 2689 images

The newly created geo-spatial terrain coverage database will now be used to create probabilistic terrain class priors on the 6.5 million geo-tagged image database, where the probability of a terrain label for an image given that image’s geo-tagged latitude-longitude information can be derived as follows:

$$P(\text{label} | \text{lat}(i), \text{lon}(i)) = \frac{\text{label} \in \{\text{GeoLabel}(\text{lat}(i), \text{lon}(i))\}}{\sum_{\text{class}=1}^C \text{class} \in \{\text{GeoLabel}(\text{lat}(i), \text{lon}(i))\}}. \quad (1)$$

where $C=5$ is the number of classes, i is the i^{th} image in the 6.5 million image database and $\text{GeoLabel}()$ is our terrain labeled spatial database indexed in lat-lon coordinates. This completes the first stage in which we probabilistically label each image given its geo-tag.

In the second stage of creating the 3D model representation, we extract feature vectors for images in both the 6.5 million image geo-tagged database as well as for images in the terrain-labeled truth database from [15]. We utilize the Gist feature descriptor [15], which has been shown to work well for terrain classification and scene categorization [5][6]. Using the extracted Gist feature vectors, we initialize a KNN classifier on the 2689 image terrain-labeled database from [15]. For each of the 6.5 million probabilistically labeled images, we run the pre-computed Gist feature through the KNN classifier. Instead of determining a single label based on the typical KNN majority-rule, we instead take the K neighbors and their associated truth labels to find the probability of a terrain label given the image’s Gist feature vector, F_i :

$$P(\text{label} | F_i) = \frac{\sum_{j=1}^K \text{label}(T_j) == \text{label}}{K}. \quad (2)$$

where K is the number of neighbors used in KNN, j is the nearest j^{th} Gist vector from the truth database to feature F_i in L2 distance of Gist feature space and i is the i^{th} image in the 6.5 million image database. We now use the geo-spatial probabilistic labeling as a prior to express the probability of a label for an image given its geo-tag and Gist feature F_i as:

$$P(\text{label} | F_i, \text{lat}(i), \text{lon}(i)) = P(\text{label} | F_i) * P(\text{label} | \text{lat}(i), \text{lon}(i)). \quad (3)$$

This probabilistically labeled geo-tagged/terrain data set composed of feature vectors, associated terrain labels and geo-tags serves as the improved representation of the 3D world model. Next, we train an additional classifier on the world-model database and test on a hold-out set of images, for which we have both geo-tags and terrain labels. Similar to [6], we chose a KNN classifier to make the classification problem computationally tractable. For each test image, we compute a Gist feature and use the KNN classifier with K' nearest neighbors (note that this is a different parameter than K used in creation of 3D world model). Unlike [6] who used KNN to label the query image by neighbor majority rule, we choose the label for the image by computing the label likelihood over the neighborhood Gist features as follows:

$$\text{Predicted Label} = \underset{\text{label}}{\operatorname{argmax}} \sum_{j=1}^{K'} P(\text{label}(j) | F_j, \text{lat}(j), \text{lon}(j)). \quad (4)$$

where K' is the number neighbors used in KNN, j is the nearest j^{th} Gist vector from the 6.5 million image database to the Gist feature derived from the hold-out image.

2.3 Coarse-Scale Geo-location Results

The coarse geo-location method was tested on a hold-out set of 5000 test images, 1000 images per terrain class. We extracted a Gist feature for each image, using Gabor filters at 4 angles and 4 scales (16 channels). Color information is captured using red-green and blue-yellow center surround, each with 6 scale combinations, leading to 12 sub-channels [15]. Intensity is captured using dark-bright center surround with 6 channel combinations, leading to 6 sub-channels, for a total of 34 sub-channels. Each channel is encoded by a 16 bin histogram, leading to a feature vector of length 544. PCA was applied to the extracted Gist feature to reduce the feature dimensionality from 544 to 80 dimensions in order to avoid sparsity concerns while still keeping 94% of the feature vector variance. By cross-validation, we chose $K=9$ for creation of the geo-tagged/terrain labeled database, and $K'=200$ for the k-nearest neighbors used to predict the label of a test image. We compare the results of our method, shown in Table 2, to the baseline method as detailed in [15], shown in Table 1. The baseline had an average accuracy of 85.54%, while our method had an accuracy of 91.26%, a significant improvement of 5.72% over the baseline. In particular, our method was able to classify coastal areas with an 11.3% improvement over the baseline and country areas with a 6.4% improvement. Based on the percentage of world land coverage types in our database, the search space reduction varies anywhere from 70% for “country” scenes to 90% for “urban” and “coastal” scenes.

In summary, we have developed an improved method to coarsely geo-locate images on a world-wide scale by classification of terrain types. The resulting terrain label for a query image can now be used to reduce the geographical search space, as well as choose a specific medium geo-location classifier trained to further distinguish spatial locality within that particular terrain class.

Table 1. Confusion Matrix for Baseline Method

	Coast	Country	Forest	Mountain	Urban
Coast	81.1	9.2	2.9	4.6	2.2
Country	10.6	82.7	3.6	1.3	1.8
Forest	0.8	3.5	89.1	4.9	1.7
Mountain	1.4	4.5	4.2	87.6	2.3
Urban	2.2	4.2	1.1	5.3	87.2

Table 2. Confusion Matrix for Coarse-scale Geo-location Method

	Coast	Country	Forest	Mountain	Urban
Coast	92.4	2.5	1.9	1.7	1.5
Country	6.3	89.1	2.2	1.1	1.3
Forest	0.7	2.4	92.1	3.7	1.1
Mountain	0.5	1.1	2.7	92.6	3.1
Urban	1.5	2.2	1.6	4.6	90.1

3 Medium-Scale and Fine-Scale Geo-location

We describe the medium and fine scale geo-location method, where a query image has already been labeled as belonging to a certain terrain type, leading to a significant reduction in geospatial search space. The next step is to geo-locate within this reduced search space. For the medium-scale geo-location problem, we will focus on urban scenes to answer the following question: given this urban image, which city was the image taken from?

3.1 Background and Related Work

Unlike coarse geo-location, where the problem consisted of classification between 5 terrain classes, the medium-scale classification problem is much more complex: we are now attempting to differentiate between thousands of city classes. Furthermore, the boundary between the classes (e.g. is this location Bangkok or Paris?) is extremely complex because it must divide a spectrum of scene types that might be present in multiple of these geographic locations [6].

Because of the very large number of classes and the complex class boundaries between them, there is a need to work with features in higher dimensions to get accurate class separation. As mentioned in section 2.1, the KNN method used for coarse geo-location tends to work well in low to medium-sized dimensional spaces, but suffers as feature dimensionality increases [6]. For the medium-scale geo-location, we still have a large image database (on order of millions) that prevents us from directly using a supervised classifier, but we need to solve a complex classification problem. Thus, there is potential gain in classification performance as long as we have a computationally tractable supervised training method that focuses on the relevant features for the given task or query. One promising approach for high dimensional features is to combine the supervised learning power of SVM with computationally efficiency of

KNN. The medium and fine-scale geo-location methods proposed in this paper are inspired by SVM-KNN [18][6] and prior KNN enhancements [19][20]. The method is a hybrid of non-parametric, KNN techniques and parametric, supervised SVM learning techniques. The philosophy behind this method is that learning becomes easier if we focus on examining the local space around a query instead of the entire space in the original problem domain.

Let's consider our image geo-location problem where we are attempting to differentiate between multiple cities (e.g. is this location Bangkok or Paris?). When looking at the combined training data for both cities, there might not be a simple parametric boundary between these geographic classes in feature space. However, if we were to look within a space of similar scenes to the query image (e.g. images of streets), then it may become much easier and more feasible to divide the classes. This is exactly what we intend to do with the KNN-SVM algorithm. Given a query image, we will use KNN to roughly find a local space of similar scenes and then use an online SVM classifier trained just on the nearest neighbors to find a possibly non-linear parametric boundary and classify the query image. The proposed KNN-SVM algorithm will not only be computationally tractable, but also has the potential to have significantly improved classification performance over a KNN-only method.

3.2 Medium and Fine-Scale Geo-location Methods and Results

Our KNN-SVM classifier builds upon the baseline method described in [6]. Given a query image, we first extract a single feature vector for each image using several popular feature detectors from literature. We extract Tiny Images, as detailed by Torralba et al. in [21], to create 16x16 color images as one of our features. In addition, we use color histograms of size 4x14x14 bins in CIE L*a*b* space for a total of 784 dimensions. Texton features are also used due to their ability to distinguish well between different building textures in cities. Similar to [6] we use a 512 entry universal texton dictionary [22] by clustering data to a set of bank filters with 8 orientations, 2 scales and 2 elongations. Finally, we apply the same Gist feature descriptor as detailed in section 2, of size 544 dimensions. Given a query image and our large urban-only database, we developed the following KNN-SVM method:

1. Starting with urban-only database of N images, automatically label “regions” using mean-shift clustering with 200km bandwidth.
2. Use baseline KNN-SVM from [18] with $K_1 = 2000$ to find a “region” label
3. Run KNN-SVM with data only from region, with $K_2 = \sqrt{N(\text{label} == \text{"region")}}$
4. Cluster on the globe the K_2 nearest neighbors by mean-shift, using window size=50 km. Consider each cluster as a city for SVM.
5. Compute the pair-wise distances between all K_2 nearest neighbors using image features with L1 distance.
6. Convert the pair-wise distance into a positive semi-definite kernel matrix using the Radial Basis Function (RBF) kernel and train C 1-vs-all non-linear SVMs.
7. For each classifier C, compute distance of the query point to the decision boundary. The class with maximum positive distance is declared the “winner”.
8. Estimate GPS of query as average of all members of the winning class.

We tested the new algorithm using a 500 image hold out set, composed of geo-tagged images from 5 cities with 100 images per city. The images include the cities of 1. Lubbock, Texas, 2. Boston, MA, 3. Paris, France, 4. Vienna, Austria and 5. Dubrovnik, Croatia. For each query image, we test the image against the whole database to determine the geo-location performance for finding the particular city amongst data from the entire world. Successful geo-location for a query image is defined as finding a location within 200km of the actual GPS location as specified in the geo-tagged metadata of the query image. Table 3 captures geo-location performance.

The results indicate that we can geo-locate a query image to a particular city with an accuracy of 12% to 18%, with an average of 15%. Previous results from [6] on the same database resulted in an accuracy of 12.5%. Our method has an absolute improvement of 2.5%, leading to a 20% relative improvement over previous results.

Table 3. Medium and Fine-scale Geo-Location Confusion Matrix at the city-level

	Lubbock, TX	Boston, MA	Paris, France	Vienna, Austria	Dubrovnik, Croatia	Other
Lubbock, TX	16	2	0	0	1	81
Boston, MA	2	18	0	0	0	80
Paris, France	0	0	12	2	2	84
Vienna, Austria	0	0	1	13	0	86
Dubrovnik, Croatia	0	0	2	1	16	81

4 Discussion and Conclusions

Image geo-location on a world-wide scale is a very challenging problem. Besides being an interesting problem in itself, it can be tremendously useful for many other vision tasks, such as image retrieval, object detection and recognition. For instance, the distribution of likely geo-locations of a particular image provides additional context, such as terrain type, population density, and salient cultural markers. This additional metadata can be used as priors for object detection and recognition to tailor a particular object detection algorithm at recognizing objects that might be found in that particular region of the world.

In this paper, we developed a novel image geo-location framework using a hierarchical approach. We applied the method to geo-locate images using a 6.5 million image database. Starting at the coarsest level, we determined regions in the world by terrain classification and obtained a 91.3% accuracy rate, a 6% improvement over previous state of the art. In addition, we developed a novel medium and fine-scale classifier to geo-locate images at the city scale, resulting in a relative improvement of 20% over previous state of the art on this particularly challenging data set. Future work will focus on further improving the coarse geo-location by upgrading the KNN classifier to a KNN-SVM classifier similar to the one used for the medium and fine scale geo-location. In addition, we plan to further test the medium and fine scale geo-location classifiers on an expanded test data set with both natural and urban images.

References

1. Zhang, W., Kosecka, J.: Image Based Localization in Urban Environments. In: 3DPVT 2006 (2006)
2. Zamir, A.R., Shah, M.: Accurate Image Localization Based on Google Maps Street View. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 255–268. Springer, Heidelberg (2010)
3. Pollefeys, M., Nister, D., Frahm, J., Akbarzadeh, A., Mordo-hai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P.: Detailed Real-Time Urban 3D Reconstruction from Video. IJCV 78(2-3), 143–167 (2008)
4. Snavely, N.: Scene Reconstruction and Visualization from Internet Photo Collections. Doctoral thesis, University of Washington (2008)
5. Hays, J., Efros, A.A.: IM2GPS: estimating geographic information from a single image. In: CVPR 2008 (2008)
6. Hays, J.: Large Scale Scene Matching for Graphics and Vision. CMU PhD Thesis (2009)
7. Bellman, R.E.: Dynamic programming. Princeton University Press, Rand Corporation (1957) ISBN 978-0-691-07951-6
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
9. De Bonet, J.S., Viola, P.: Structure driven image database retrieval. In: Advances in Neural Information Processing, vol. 10, pp. 866–872 (1997)
10. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. In: Visual Perception. Progress in Brain Research, vol. 155 (2006)
11. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
12. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR, vol. 2, pp. 2161–2168 (2006)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2008)
14. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: label transfer via dense scene alignment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2009)
15. Oliva, A., Torralba, A.: Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision 42(3) (2001)
16. Global Land Cover Characterization Database,
<http://edc2.usgs.gov/glcc/glcc.php>
17. UNEP, Mountains and Tree cover in Mountain Regions (2002),
http://www.unep-wcmc.org/mountains-and-tree-cover-in-mountain-regions-2002_724.html
18. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: CVPR 2006 (2006)
19. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: CVPR 2008 (2008)
20. Domeniconi, C., Gunopulos, D.: Adaptive nearest neighbor classification using support vector machines. In: NIPS (2001)
21. Torralba, A., Fergus, R., Freeman, W.T.: Tiny images. MIT-CSAIL-TR-2007-024 (2007)
22. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. ICCV (July 2001)

An Image Based Approach for Content Analysis in Document Collections

Reinhold Huber-Mörk and Alexander Schindler

Intelligent Vision Systems, Safety & Security Department
AIT Austrian Institute of Technology GmbH
Vienna, Austria
`reinhold.huber-moerk@ait.ac.at`

Abstract. We consider the task of content based analysis and categorization in large-scale historical book scanning projects. Mixed content, deprecated language, noise and unexpected distortions suggest an image based approach. The use of keypoint extractors combined with the bag of features approach is applied to scanned text documents. In order to incorporate spatial information into the bag of features approach we consider three methods of spatial verification. An approach based on comparison of statistical properties of local keypoint properties such as size orientation and scale showed comparable quality in content comparison while being computationally much more efficient. Cluster analysis delivers groups of pages characterized by common properties, especially duplicated page content is detected with high reliability.

1 Introduction

Approaches for fast access to digital-born or digitized modern documents are successfully applied on the web and in modern document workflows employing information retrieval techniques. We investigate content analysis, especially detection of duplicated content, in large scale scanning projects of historical books [16] and newspapers [3]. The large amount of visual data raises issues of automatic indexing, quality assurance and information extraction based on image processing. A common approach to document image analysis is to index and compare pages based on textual information extracted through Optical Character Recognition (OCR). This method is quite limited with respect to accuracy and flexibility, especially when taking into account historical documents that are typically characterized by mixed content, deprecated language, annotations, noise and unexpected distortions. In fact, non-textual content is sometimes predominating and contains significant information (e.g. stamps, handwritten remarks etc.). Considering historical books, which may consist entirely of ancient typesettings and drawings, OCR tends to fail [6,19].

In general, several approaches for the identification of individual objects in large image collections have been proposed in the literature. Near-duplicate detection of keyframes using one-to-one matching of local descriptors was described for video data [27]. A bag of features (BoF) [5] derived from local descriptors

was described as an efficient approach to near-duplicate video keyframe retrieval [24]. For detection of near-duplicates in images and sub-images local descriptors were applied [14]. Doermann et. al [6] discussed the problem of duplicate detection in large document image archives and pointed out the advantage of an image recognition based approach. The approach taken at this time was to base the analysis on the shape of characters. Beusekom et. al [22] address the analysis of different versions of scanned historical documents. Baluja and Covell [1] describe an approach to differentiate between text and image content, especially line drawings, in scanned document pages.

Image based approaches can be used for detection of image content and are commonly based on local image feature descriptors. One of the most prominent local keypoint detection and description methods is the Scale Invariant Feature Transform (SIFT) [17]. The BoF derived from local descriptors such as SIFT was described as an efficient approach to content based retrieval and detection from image data. The BoF approach is inspired by the bag of words approach based on term frequency weighting and comparison in text retrieval [18]. SIFT features are combined with AdaBoost learning to obtain relevant information in large-scale book-scanning systems, e.g. preview page [1]. Discrimination of main body text and decorative elements in historical manuscripts using SIFT and a support vector machine (SVM) classifier is described by Garz et. al [9].

Detection of duplicated pages in an automatic document scanning workflow was investigated for historical documents where a BoF based on SIFT descriptors approach was chosen and results were compared to manually obtained data [12]. This paper demonstrates image content grouping by similarity using a BoF approach combined with spatial verification schemes. Groups of similar pages, e.g. layout or graphical properties etc., could be identified as well as duplicated content.

The remainder of this paper is organized as follows. Section 2 describes methods for visual content comparison and introduces our approach. Results are presented in Section 3 and Section 4 summarizes the paper.

2 Image Comparison

To detect and describe interest regions in document images we used the SIFT keypoint extraction and description approach. Subpixel image location, scale and orientation are associated with each SIFT keypoint. The associated SIFT descriptor consists of a 4×4 location grid containing 8 gradient orientation bins in each grid cell. The descriptor vectors of length 128 will be used to learn a visual dictionary, i.e. the BoF. Spatial verification becomes important as the BoF does not represent spatial relationships between the visual words present in an image. We will compare three methods for spatial verification: (1) estimation of a homography and comparison in the image domain, (2) comparison of the co-occurrence statistics of the visual words for two images and (3) global detector statistics comparison.

The inverse document frequencies (idf) is often used in combination with term frequencies (tf) of visual words in the BoF approach. Jégou et. al pointed

out weaknesses of the idf scheme related to burstiness of visual features, i.e. multiple occurrence of a specific visual words in one image or specific visual words occurring among many images [13]. For text we observed burstiness of visual words occurring at similar spacings between characters at fine scales and for pieces of characters at very fine scales. Combination with a higher level description, e.g. using visual word co-occurrence matrices [26] or co-ocsets [4], by spatial subdivision such as spatial pyramid matching [25], or verification based on matching in the image space [15] try to overcome limitations of a purely tf based approach.

Learning of the visual dictionary is performed using a clustering method applied to all SIFT descriptors of all images, which could become computationally very demanding. As a single scanned book page already contains a large number of local descriptors we applied preclustering of descriptors to each image. In contrast to a similar procedure, where all descriptors for all images of the same category are clustered independently and subsequently appended to the BoF [10], we construct a list of clustered descriptors for each page and cluster this list in a second step in order to obtain a dictionary for the whole book. We used k-means and euclidean distance for preclustering and final clustering of the BoF. Individual terms i occur on each page with varying frequency t_i . The visual histogram of term frequencies t_i for an individual book page is derived from the BoF representation by counting the indices of the closest descriptors. The term frequencies t_i are represented in its normalized form, i.e. $\sum_{i=1 \dots |V|} t_i = 1$, where V is the set of visual words contained in the visual vocabulary for an individual book. Matching of two visual term frequency histograms t^a and t^b is based on histogram intersection $T_{ab} \in [0, 1]$ given by

$$T_{ab} = \sum_{i=1}^{|V|} \min(t_i^a, t_i^b). \quad (1)$$

To group image with respect to similarity we first calculate the similarity measure T_{ab} for each page a to all other pages b in the collection B (usually referring top a single book). Taking the maximum of $T_{ab}, \forall b \in B, b \neq a$ delivers a view of collection consistency, i.e. if all T_{ab} are similar the document content is quite homogeneous and if T_{ab} shows different modes the content and page structure is supposed to be mixed. Figure 1(a) shows a plot for T_{ab} for an example book-scan consisting of 256 pages. The main body of the book receives a maximum T_{ab} of around 0.6 which basically is related the self-similarity of the text pages. Bursts exceeding this value are typically duplicated pages. Single maximum peaks correspond to very similar pages of low noise, e.g. empty pages. Peaks of low similarity measure are pages not similar to any other content, e.g. these are often the cover pages.

Spatial verification is applied to each page a using a shortlist delivered by ranking the similarity T_{ab} . Details of combination of term frequency matching with three different approaches for spatial verification is described in the following.

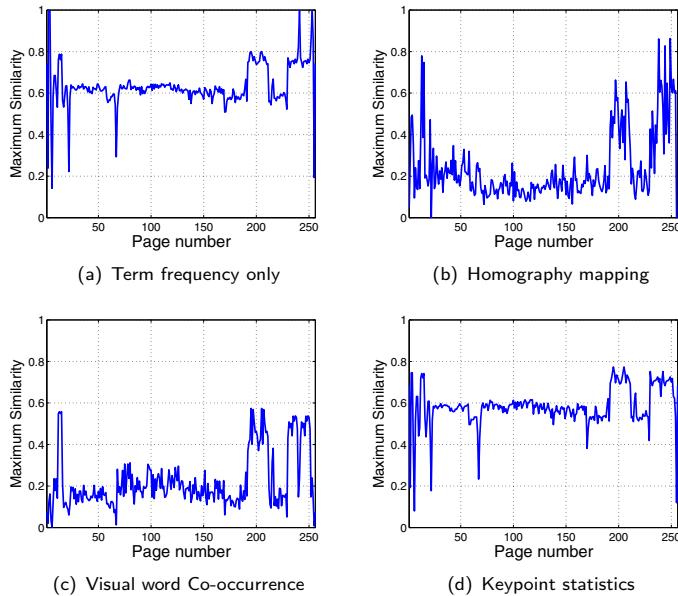


Fig. 1. Maximum similarity for each image of a book collection

2.1 Homography Estimation and Image Similarity

An affine transformation was found sufficient to overlay images obtained by current book-scan devices as the main problem of bent pages typically occurring with flatbed scanners is not observed. Matching of SIFT features uses the RANSAC procedure for robust estimation [8]. In order to limit the complexity of the matching procedure spatial subsampling of keypoints by identifying the most salient keypoints with respect to a spatial grid was employed [11]. The similarity of two overlaid images is expressed by the mean structural similarity index (SSIM) [23], where a mean $\text{SSIM} \rightarrow 1$ indicates identical content and an $\text{SSIM} \rightarrow 0$ means unrelated content. Figure 1(b) shows the similarity plot for the considered example.

2.2 Co-occurrence of Visual Words

We follow the basic ideas concerning descriptive visual words as described by Zhang et. al [26]. For each image co-occurrence matrices $c(v_s, v_t), \dots, v_s, v_t \in V$ counting the concurrent appearance of visual words v_s and v_t in a spatial neighborhood are constructed. Each keypoint is assigned to a visual word and delivers a contribution to the co-occurrence statistics by counting the concurrent presence of the visual words assigned to all keypoints contained in its spatial neighborhood. The spatial neighborhood is a circular region around a keypoint location depending on the corresponding keypoint size delivered by SIFT.

The size d of the influence region for a keypoint was chosen as $d = s \cdot p_d$, where s is the estimated size of the SIFT keypoint and p_d the scaling parameter. A selection of $p_d = 6$ is derived from the spatial support of the SIFT detector. Visual word co-occurrence matrices c^a and c^b are normalized $\sum_{s \in |V|} \sum_{t \in |V|} c(v_s, v_t) = 1$ and matched using 2D-histogram intersection $C_{ab} \in [0, 1]$

$$C_{ab} = \sum_{s=1}^{|V|} \sum_{t=1}^{|V|} \min(c^a(v_s, v_t), c^b(v_s, v_t)). \quad (2)$$

The most crucial part of this approach is to identify spatially adjacent neighboring keypoints for each keypoint. The k-d tree representation allows efficient representation and queries of spatially local neighborhoods [2]. Nevertheless, the computational demands of this method were found on the order of magnitude when compared to homography based image comparison. Figure 1(c) shows the similarity plot for the considered example.

2.3 Global Keypoint Property Statistics

We suggest a method based on global statistics of keypoint properties. SIFT delivers location, size and orientation for each keypoint.

We use a measure of inhomogeneity to characterize the spatial distribution of keypoints [21]. The image is subdivided into a sequence $s^2, s = 1, 2, 4, \dots$ of rectangular regions of equal size and the number of keypoints m_i falling into region $i, i = 1, \dots, s^2$ is obtained

$$h = \sum_{j=1}^{\log_2 s} w^{1-j} h(2^j), \quad h(s) = \frac{1}{2n} \sum_{i=1}^{n^2} |m_i - \frac{n}{s^2}|, \quad (3)$$

where $w = 4.79129$ was derived in [21]. Images having spatially uniformly distributed keypoints obtain values of $h \rightarrow 0$ and whereas for spatially concentrated keypoints we get $h \rightarrow 1$.

We exploit the orientation estimation delivered by SIFT using a measure for circular uniformity U . The U measure was introduced by Rao [20] in his test for circular uniformity

$$U = \frac{1}{2} \sum_{i=1}^{n-1} (|(\alpha_{i+1} - \alpha_i) - \lambda| + |(360 - \alpha_n + \alpha_1) - \lambda|). \quad (4)$$

where $\lambda = 360/n$ and the angle directions $0 \leq \alpha_i < 360$ are sorted in ascending order $\alpha_{i+1} > \alpha_i, \forall i = 1, \dots, n-1$. We normalize the uniformity measure to the range $[0, 1]$ by $u = U/360$. Images with keypoints directions pointing uniformly into all directions obtain $u = 0$ and coherently oriented keypoints we get $u \rightarrow 1$.

We have chosen the variance of the keypoint size S as a descriptor for the distribution of the size estimations over all detected keypoints. A normalized version s is obtained from

$$s = S / (\sigma_0 2^{O_{\max} + (S_{\max} - 1)/S_{\max}}), \quad (5)$$

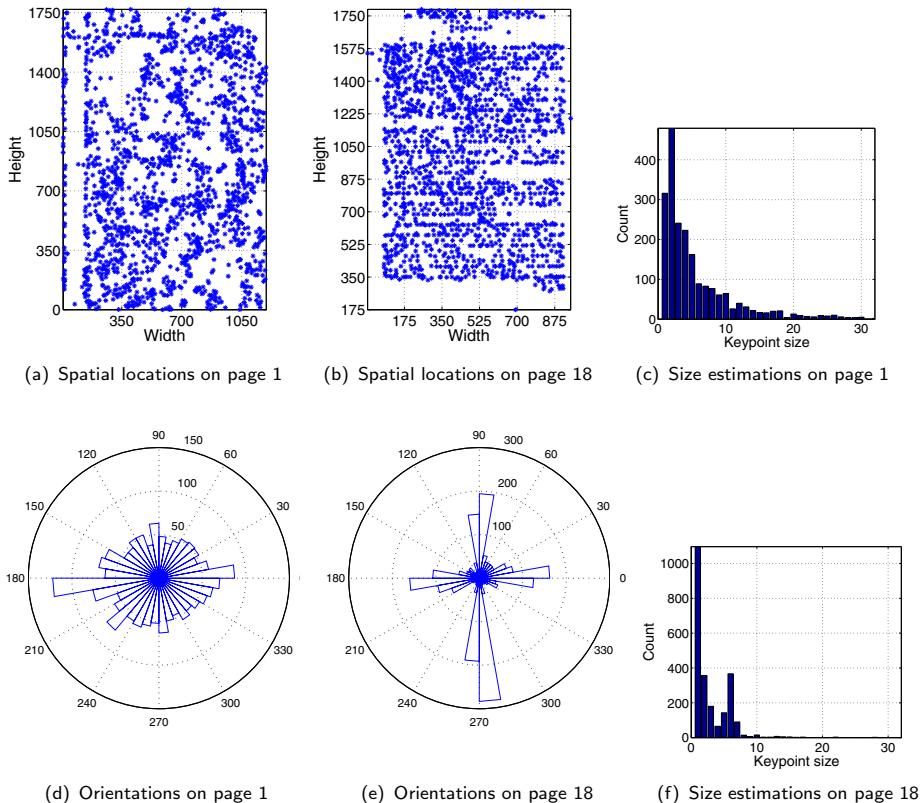


Fig. 2. Statistical characterization of keypoints derived from spatial, circular and scale properties for two different book pages

where σ_0 is the initial Gaussian smoothing parameter. The maximum scale index is denoted by s_{\max} and is typically set to 4. The maximum octave index is given by o_{\max} and depends on the image resolution.

Figure 2 shows spatial radial and size distributions for keypoint properties. The plots on the left correspond to the image shown in Figure 3(a) and $h_1 = 0.13335$, $U_1 = 226.2887$ and $S_1 = 3.8964$ were calculated. The plots on the right correspond to the image shown in Figure 3(c) and $h_{18} = 0.2116$, $U_{18} = 274.6681$ and $S_{18} = 2.2229$ were calculated, i.e. text content tends to be spatially homogeneous, circular less uniform and has lower scale variation.

2.4 Combination of Term Frequency with Spatial Verification

Term frequency based comparison was used to deliver a shortlist L , we used a size of $|L| = 3$ in our experiments. Based on this shortlist we applied spatial verification and combined term frequency matching and spatial matching in a

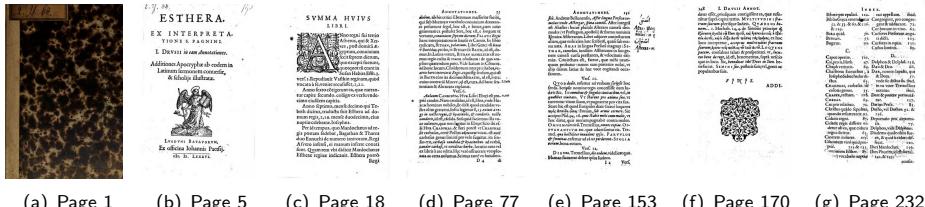


Fig. 3. Scanned sample pages of a historical book

conjunctive fashion, i.e. the final similarity measure is derived from $T_{ab} \cdot V_{ab}$, $b \in L$ where V_{ab} is either $V_{ab} = SSIM$ in the case of homography based verification, $V_{ab} = C_{ab}$ in the case of co-occurrence based verification and the following heuristics $V_{ab} = 1 - \sqrt{(|h_a - h_b| + |u_a - u_b| + |s_a - s_b|)/3}$ is used for the approach based on keypoint property statistics.

3 Results

We consider collections of historical books obtained by an automated book-scanning device, see Figure 3 for sample pages extracted from an exemplary book. A BoF for each scanned book is constructed and visual term histograms for each page are extracted. A combination with spatial verification is performed as described in Sec. 2.4. It was observed, that page content and structure also depends on the absolute page count. Furthermore, when considering the case of duplicated pages it was observed that duplicated pages occur blockwise with respect to the scan sequence. This occurs due to the operations of the automatic book scan device, which turns back a number of pages in situations of a possible error.

We performed an unsupervised clustering of the space spanned by maximum similarity for each page and the normalized page count. Identified clusters were characterized by similar page content or layout. We used the DBSCAN algorithm [7] to discover clusters in the page similarity/index space. Figure 4(a) shows the result of DBSCAN, where seven clusters have been identified. The isolated star-shaped points are outliers and represent unique content. The main body of the book is covered by clusters 4 and 5, example pages are shown in Figure 4(d) and Figure 4(e). Duplicated pages are detected from clusters 1, 2 and 3, e.g. Figure 4(b) shows two scanned pages of same content with small differences in skew and noise. Duplicate detection delivered correct results when visually verified. The regions of duplications are automatically obtained by thresholding of the similarity measure [12].

Run time measurements were performed on a Xeon 3.6 GHz computer using a MATLAB/C implementation. To analyze a scanned book with 256 pages scanned at 72 DPI (1800×1200 pixel) it took 31 seconds for tf matching only, combined matching based on homography estimation and image comparison took

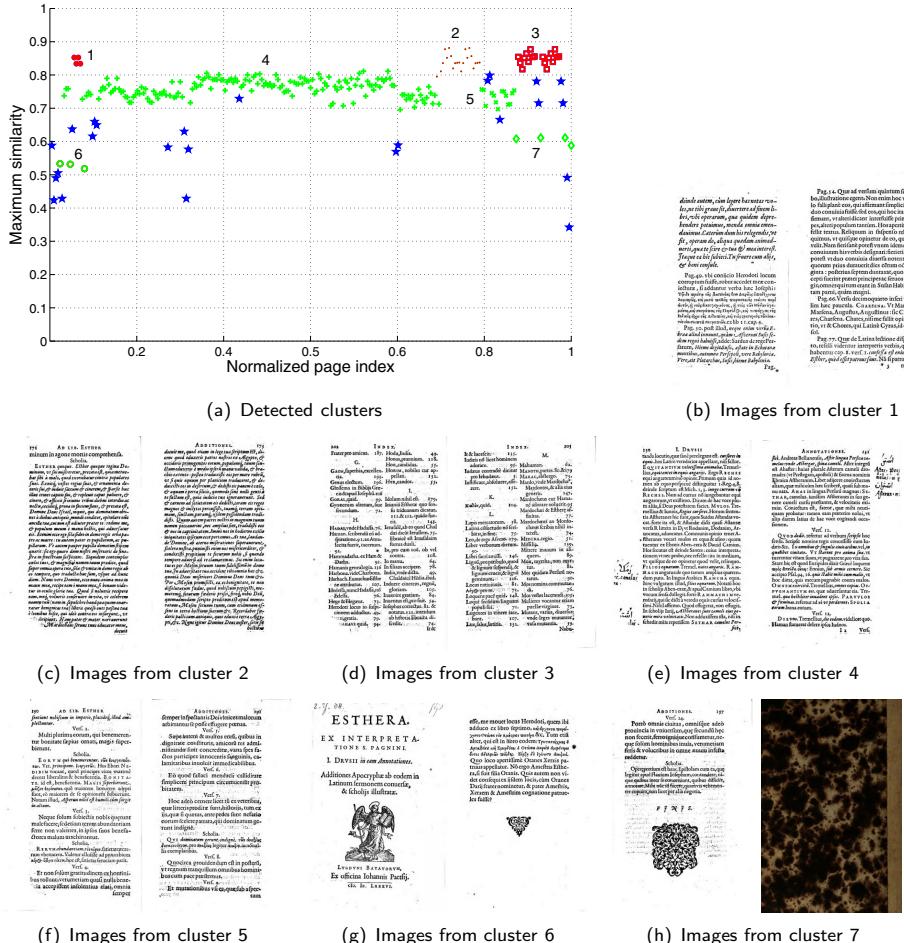


Fig. 4. Clustering of page similarity/index plane and sample images from different clusters

449 seconds, combination with co-occurrence verification took 451 seconds and combined matching using keypoint property statistics consumed 128 seconds.

4 Conclusion

We have presented an approach for visual page content clustering applicable to duplicate detection in book-scan systems. A BoF approach combined with spatial verification based on keypoint statistics was found suited for the analysis of scanned historical book collections. Background knowledge on the sequential nature of the scanning process and incorporation of spatial knowledge using global

keypoint statistics improves results significantly. Clusters of duplicated content are automatically detected and subject manual quality assurance in a library workflow. The system is currently evaluated for the task of content characterization and duplicate detection at the Austrian National Library. Future research includes content classification with respect to image quality categories.

Acknowledgment. The authors would like to thank Sven Schlarb from the Austrian National Library (ONB) for providing data and expertise on library workflows.

This work is part of the SCALable Preservation Environments (SCAPE) project which aims at developing scalable services for planning and execution of institutional digital preservation strategies. The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement no. 270137).

References

1. Baluja, S., Covell, M.: Finding images and line drawings in document-scanning systems. In: Proc. Intl. Conf. on Doc. Anal. and Retrieval, ICDAR 2009 (2009)
2. Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Commun. ACM* 18(9), 509–517 (1975)
3. Chaudhury, K., Jain, A., Thirthala, S., Sahasranaman, V., Saxena, S., Mahalingam, S.: Google newspaper search - image processing and analysis pipeline. In: Proc. Intl. Conf. on Doc. Analysis and Recognition, ICDAR 2009 (2009)
4. Chum, O., Matas, J.: Unsupervised discovery of co-occurrence in sparse high dimensional data. In: Proc. Comp. Vis. and Pat. Rec., CVPR 2010 (2010)
5. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV 2004 (2004)
6. Doermann, D., Li, H., Kia, O.: The detection of duplicates in document image databases. *Image and Vision Computing* 16(12-13), 907–920 (1998)
7. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. Conf. on Knowledge Discovery and Data Mining, KDD 1996 (1996)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
9. Garz, A., Sablatnig, R., Diem, M.: Layout analysis for historic manuscripts using SIFT features. In: Proc. Intl. Conf. on Doc. Anal. and Rec., ICDAR 2011 (2011)
10. Hazelhoff, L., Creusen, I., van de Wouw, D., de With, P.H.N.: Large-scale classification of traffic signs under real-world conditions. In: Proc. SPIE Electronic Imaging: Algorithms and Systems VI (2012)
11. Huber-Mörk, R., Schindler, A.: Quality assurance for document image collections in digital preservation. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P., Zemčík, P. (eds.) ACIVS 2012. LNCS, vol. 7517, pp. 108–119. Springer, Heidelberg (2012)
12. Huber-Mörk, R., Schindler, A., Schlarb, S.: Duplicate detection for quality assurance of document image collections. In: Proc. Conf. on Digital Preservation, iPRES 2012 (2012)

13. Jégou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: Proc. Computer Vision and Pattern Recognition, CVPR 2009 (2009)
14. Ke, Y., Sukthankar, R., Huston, L.: An efficient parts-based near-duplicate and sub-image retrieval system. In: Proc. Intl. Conf. on Multimedia, MULTIMEDIA 2004 (2004)
15. Knopp, J., Sivic, J., Pajdla, T.: Avoiding confusing features in place recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 748–761. Springer, Heidelberg (2010)
16. Langley, A., Bloomberg, D.S.: Google books: making the public domain universally accessible. In: Proc. of SPIE, Doc. Rec. and Retrieval XIV (2007)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. of Comput. Vision 60(2), 91–110 (2004)
18. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, 7th edn. Cambridge University Press (2008)
19. Ramachandrula, S., Joshi, G.D., Noushath, S., Parikh, P., Gupta, V.: PaperDiff: A script independent automatic method for finding the text differences between two document images. In: Proc. Intl. Workshop on Docu. Anal. Syst. (2008)
20. Rao, J.S.: Bahadur efficiencies of some tests for uniformity on the circle. Ann. Math. Statist. 43(2), 468–479 (1972)
21. Schilcher, U., Gyarmati, M., Bettstetter, C., Chung, Y.W., Kim, Y.H.: Measuring inhomogeneity in spatial distributions. In: Proc. Vehicular Technology Conference, VTC 2008 (2008)
22. van Beusekom, J., Shafait, F., Breuel, T.M.: Image-matching for revision detection in printed historical documents. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 507–516. Springer, Heidelberg (2007)
23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Proc. 13(4), 600–612 (2004)
24. Wu, X., Zhao, W.-L., Ngo, C.-W.: Near-duplicate keyframe retrieval with visual keywords and semantic context. In: Proc. Conf. on Image and Video Retrieval, CIVR 2007 (2007)
25. Xu, D., Cham, T.J., Yan, S., Duan, L., Chang, S.-F.: Near duplicate identification with spatially aligned pyramid matching. IEEE Trans. Circuits Syst. Video Techn. 20(8), 1068–1079 (2010)
26. Zhang, S., Tian, Q., Hua, G., Huang, Q., Li, S.: Descriptive visual words and visual phrases for image applications. In: Proc. Intl. Conf. on Multimedia, MULTIMEDIA 2009 (2009)
27. Zhao, W.-L., Ngo, C.-W., Tan, H.-K., Wu, X.: Near-duplicate keyframe identification with interest point matching and pattern learning. IEEE Trans. Pat. Anal. Mach. Intell. 9(5), 1037–1048 (2007)

Simultaneous Bidirectional Geometric Model Synchronization between CAD and VR Applications

Dimo Chotrov and Stoyan Maleshkov

Technical University Sofia, Virtual Reality Lab, Sofia, Bulgaria
`{dchotrov,maleshkov}@tu-sofia.bg`

Abstract. Most of the research in the field of CAD – VR integration targets the use of VR as a post-processing tool and relatively fewer attempts are aiming at returning feedback from VR back to CAD. In this paper we propose a method allowing the modification of CAD models in a virtual environment with (bidirectional) synchronization of the model description between the CAD and VR applications as the changes are made. A sample implementation of the method is also described and some examples of specific test cases used to confirm the applicability of the method are given.

1 Introduction

Virtual Reality (VR) is nowadays considered a tool of increasing importance in the product development process that can decrease development costs and increase competitiveness [10][11][14]. The application of VR tools and techniques can also improve customer satisfaction as it allows for design and applicability evaluation in the early stages of the product development process [10][11][13][14][17][21]. In [1] and [14] the authors present an extensive survey about the engineering applications of VR at different stages of the product life-cycle, as well as the hardware and software needed. While in the past only large companies, mainly from the automotive and aerospace industries, could afford themselves to finance the needed infrastructure for using VR [14], with VR hardware and software becoming more affordable in the last few years also small enterprises are able to take advantage of its potential.

Thanks to the active research in the past ten to twenty years in the field of integrating Computer Aided Design (CAD) and VR the connection between the two conceptually different technologies [6] is hardly any more only a downstream one-way process from CAD to VR as it was described fifteen years ago [4][5] with no possibilities for feedback from VR to CAD. Today terms like Immersive Engineering [11] and Virtual Reality Aided Design [14][16][21][22] are gaining popularity.

Still, the gap between CAD and VR is not completely closed [19][23]. The need for conversion between CAD and VR model descriptions remains, being through an intermediate file format or by direct meshing algorithms [12][16][18]. It is well known that this conversion may lead to errors in the tessellated model [8][9]. Although different repair algorithms have been developed [8][9][12][18], there are cases when manual repair of the mesh is required [17][19]. Furthermore, as discussed in the

next section, the results from the research in bringing changes made to the model in VR back to the CAD description of the model are still wanting.

In this paper we propose an application architecture and method for synchronizing VR and CAD model descriptions without restrictions to where (in which type of application) the changes to the model are made. We describe also our sample implementation developed to test the applicability of the proposed integration method using the CAD application Solidworks and our in-house developed VR software [25].

2 Related Work

Due to the popularity gained by VR in the last years and especially of its application in the product development process there has been an increased interest in the research community in the integration of CAD and VR. As pointed out by [23] there are two directions of CAD and VR integration. Most of the research is still looking at VR mainly as a CAD/CAE (Computer Aided Engineering) post-processing tool for viewing and assessing designs and simulations and reviewing analysis results. Some developments have resulted in very interesting and advanced applications of VR. To give just a few examples: simulation of assembly operations [2][16][17]; real-time VR exploration of very large CAD models [21]; using VR-based mechanical tools for part manipulation [3]; integration of CAD, Product Data Management and VR including automatic CAD to VR conversion and “geometrical healing” [12]; integration of VR visualization in the CAD application itself [6][23]. In [18] a solution allowing flexible interactions with CAD models and assemblies including model changes for “what-if” simulations and dynamic calculation and presentation of engineering analysis results after the change is presented. Most of the above mentioned sources handle the CAD *to* VR integration, i.e. transforming the CAD model into a VR one, allowing afterwards for different user interactions with the model.

Relatively less attention has been paid to the reverse process – making adjustments in VR and transferring the changes back to the CAD model, which would avoid the need for observing the needed changes in VR, making manually the modifications back in the CAD application and then again assessing the model in VR. One way to achieve VR to CAD feedback is presented in [20] – the information about CAD model splines and surfaces is exported from CAD and imported together with the model tessellation in the VR application. A link to the CAD description is contained together with the geometry in the scene graph of the VR application. The user is allowed to edit the splines / surfaces and then export the results in a file. [15] describes another approach using command exchange between the VR and CAD applications that allows the creation of basic primitives and modeling with boolean operations in VR. The VADE environment [2] supports automatic export of CAD data to VR, takes advantage of constraints specified in the CAD model during the assembly simulation in VR and allows the user to modify in VR pre-defined parameters of the CAD model. Changes are automatically conveyed to the CAD application which recalculates the model and the VR model is updated respectively. In [22] the authors describe their framework allowing extensive edition of CAD models in a Virtual Environment (VE),

including constraints and features, based on a Construction History Graph. Surprisingly it doesn't seem to support feedback to the CAD application. A specific solution for transferring cabling information from VR to CAD is presented in [17].

All solutions described above show some limitations when sending model data from VR (back) to CAD – usually in the type of information that can be kept consistent. Our approach, relying to some extend also on techniques described in the works mentioned above, allows for almost unrestricted CAD model editing in VR (to some extend the restrictions come from the possibilities for VR interaction for design - some of the issues have been discussed in [7]) and bidirectional transfer of changes between the CAD and VR applications the moment the changes are made.

3 Bidirectional Synchronization between CAD and VR

In order for the model information both in the CAD and VR applications to be kept consistent while working with the model the creation of a constant connection between the two applications is needed. Over this connection the two applications can then exchange data about the changes made to a model and update their own databases.

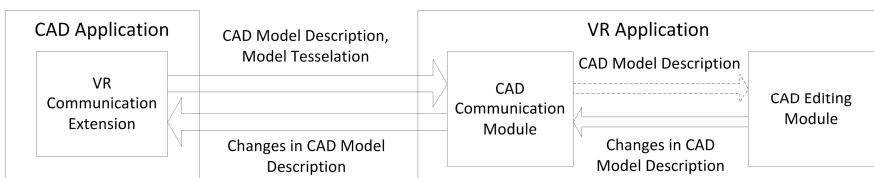


Fig. 1. Architecture for bidirectional synchronization between CAD and VR applications

Fig. 1 shows the designed architecture displaying only the modules introduced for the integration of CAD and VR. The CAD application is extended by a VR Communication Extension which is responsible for transferring model data (both CAD description and tessellation) from the CAD application to the VR one and receiving information about changes introduced to the CAD model in the VR application. Embedding an extension in the CAD application (similar approaches are described in [3] and [23]) gives access to CAD specific data like sketches, constraints, features and materials, as well as to CAD engine functions needed for model calculation. For the model tessellation CAD internal functions can be used or additional mesh generation and repair interfaces can be coupled (as proposed for example in [12] and [18]).

On the VR side a CAD Communication Module and a CAD Editing Module are introduced. The first one receives information sent from the CAD application and stores it in the VR scene data structure. The VR application can then immediately visualize the new model from the received model tessellation. If the user wants to edit a CAD model the CAD Editing Module is triggered with the CAD description of the model. The user could then make changes to the CAD model as he/she would in a CAD application but with VR interaction techniques. As the user makes changes to the CAD

model in VR the changes are simultaneously transferred to the CAD application where the new CAD model is calculated and the new CAD description and model tessellation are returned to the VR application so that it can display the updated model.

As CAD models can be very complex containing many parts, to avoid transferring large information sets between the CAD and VR applications, only information concerning the currently edited part is transferred. The partitioning of the model depends on the original parts of the CAD model. Transmitting only part of the model tessellation / description from the CAD to the VR application is probably not possible, as the changes made by the user could affect also other parts of the model depending, for example, on specific constraints or measurements.

4 Implementation and Tests

Following a sample implementation of the proposed in the previous chapter integration method based on the CAD application Solidworks with its Application Programming Interface (API) and our in-house developed VR software [25] is described.

The Solidworks CAD application is extended with an add-in (*SWToVR* on Fig. 2) that has access to the internal CAD model description as well as to the CAD engine through the API functions provided by Solidworks. Two libraries are developed to facilitate the communication between Solidworks and the VR application (see Fig. 2).

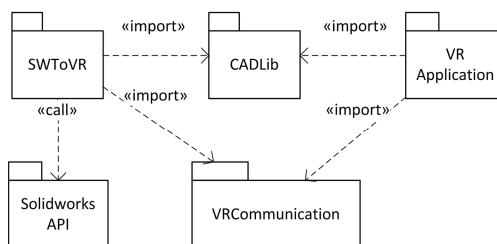


Fig. 2. Libraries developed for the CAD – VR communication and their dependencies

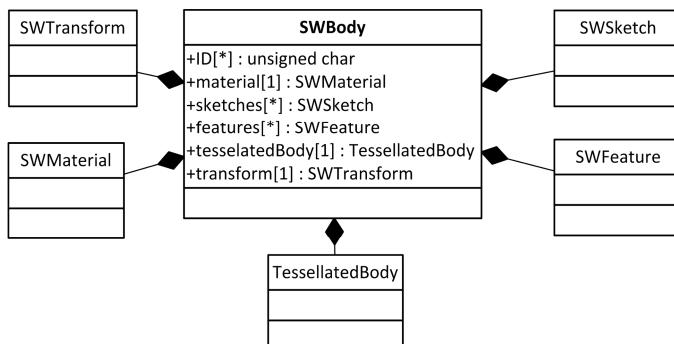


Fig. 3. Classes for transferring CAD model data between Solidworks and the VR application

The *VRCommunication* library contains classes used for the communication between different modules of the VR application (for example between visualization clients and management module). It has been extended to include classes for communication with the CAD application to facilitate the transfer of model data and changes.

CADLib contains classes that represent the CAD description of Solidworks parts (see Fig. 3). It is an intermediate library for transferring (“translating”) CAD descriptions between the two applications. To keep track of model correspondence in CAD and VR a unique persistent identifier assigned by Solidworks is used.

4.1 Storing the CAD Description in the VR Application

The information needed by a VR application for the visualization is usually stored in a scene graph (SG) – a notion first introduced by [24]. In order for the scene graph to be able to store the specific information about CAD models we have extended a standard geometry node to include also CAD description (see Fig. 4).

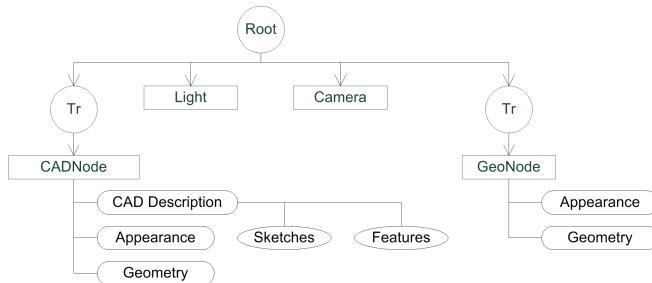


Fig. 4. Introducing a new node type in the SG structure to store information about CAD models

The data received from the CAD application can then be stored in a CADNode. Model tessellation and material information can be stored as usual in the Geometry and Appearance data of the node.

Because the CAD description is stored directly with the VR model itself in the SG the CAD information is directly available for display and editing when the user selects a model originating from CAD.

4.2 CAD and VR Synchronization

Starting a session between the CAD and VR applications requires an initial setup. In our case, as the communication succeeds over the TCP / IP protocol family, this includes specifying a listening port for the VR application to which the CAD application should connect and an IP address on which the VR application will be listening.

After the session has been initiated the user can send a CAD model (assembly, single or multiple parts) to the VR application which displays it. After that the user can continue working only with the VR application (see Fig. 5). When the user selects an object in the VR scene if it is an object originating from a CAD part the user can

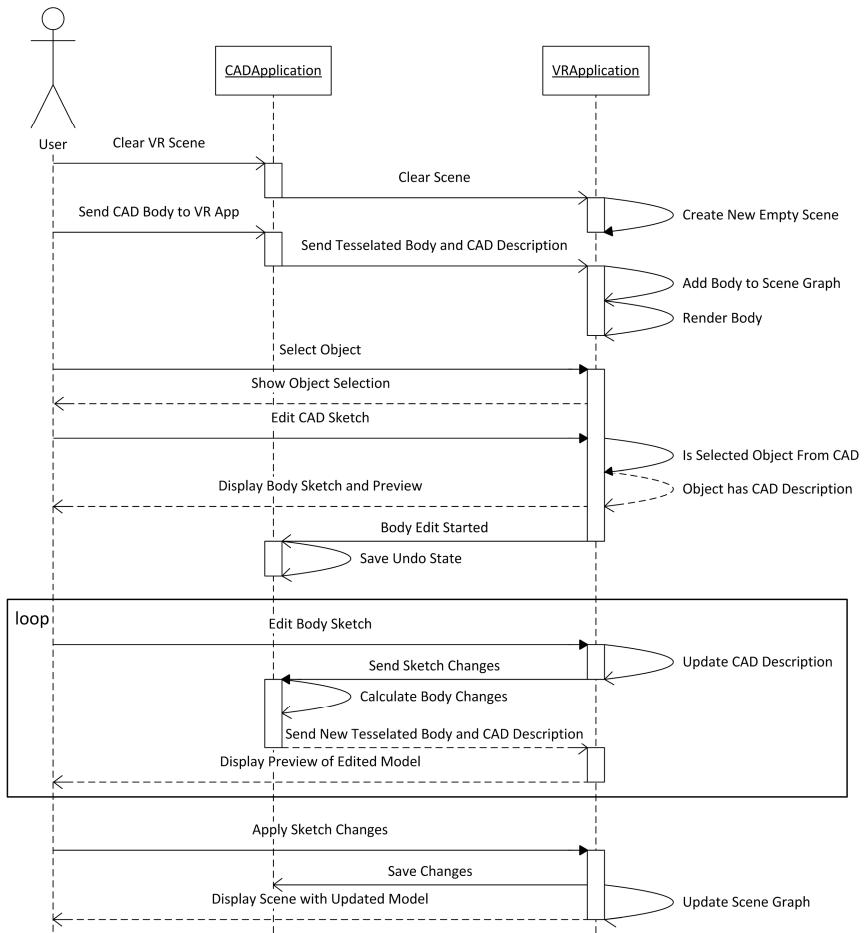


Fig. 5. Sequence diagram for the bidirectional synchronization between CAD and VR

choose to enter editing mode to modify the CAD description of the model. At this moment the VR application signals the CAD application that the user has started editing a CAD part so that the CAD application can save an undo state for the changes if the user decides to revoke them later. After that the user is presented with two views – one showing a 3D (stereo) preview of only the selected part and another allowing the modification of the CAD description. The latter view depends on what the user chooses to edit. For example, for sketch editing – the respective sketch is displayed. The stereo preview can be switched between the part currently being edited and the whole CAD model to allow the user to see how the changes are affecting the whole model. The changes to the edited part are made directly to the CAD description, i.e. when moving a sketch point in VR the position of the point in the CAD description is changed, or when changing an extrusion value, the extrusion value in the CAD description is modified, without the need for translating the changes to CAD commands.

While editing the model after each change made by the user a notification describing the modifications is sent from the VR to the CAD application. The CAD application recalculates the model and sends the updated CAD description and part tessellation to the VR one which displays the updated model. An update takes place after the user signals an end to the current operation. There is no update during the operation itself as this would lead to intensive network communication, thus degrading the interaction. At the end if the user decides to keep the changes the original scene but with the new model is displayed and the CAD application keeps the changes that have been made. Otherwise the CAD application invokes an undo operation reverting all changes made during editing mode and the VR application displays the original scene with the original model. Important here is that changes are propagated to all identical parts (if a part is present multiple times in the model). This is solved by referencing the same CAD node in the scene graph from different group nodes for repeating parts.

4.3 Tests

Following are some test cases used to test the applicability of the proposed solution.

Fig. 6 shows the modification of a simple part by editing its sketch. The image on the left displays the initial part – VR visualization client in the front and Solidworks in the back. The image on the right shows editing the sketch – the VR client displays a view where the user can select and move points of the sketch while displaying a preview at the same time. In the back the changes propagated to Solidworks can be seen.

Fig. 7 shows an example of a part with constraints. Top-left – initial state. Top-right – one of the sketches of the part is edited – although only one point of the sketch is moved, due to a constraint that the lines have to be parallel, the CAD application changes the model by modifying its extrusion along the full length of the part. The result appears accordingly in the VR application. Because of a dimensional constraint between the opening at the bottom and one of the edges the opening has become too large. To correct this, the sketch of the opening is edited to reduce its width (bottom-left). The bottom-right image shows the resultant model after accepting the changes.

Finally Fig. 8 shows an example for transferring a complex assembly together with its material data between CAD and VR in order to demonstrate the applicability of the proposed method for compound structures as well.

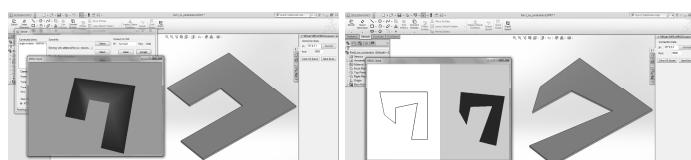


Fig. 6. Editing the sketch of a simple Solidworks CAD part

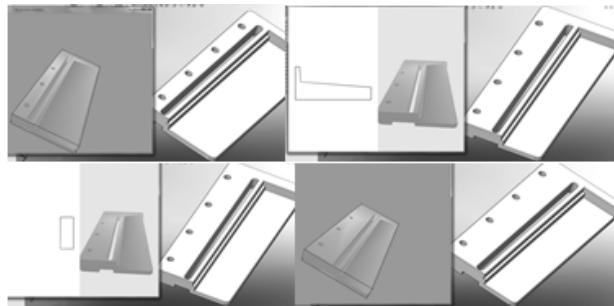


Fig. 7. Editing a model with constraints

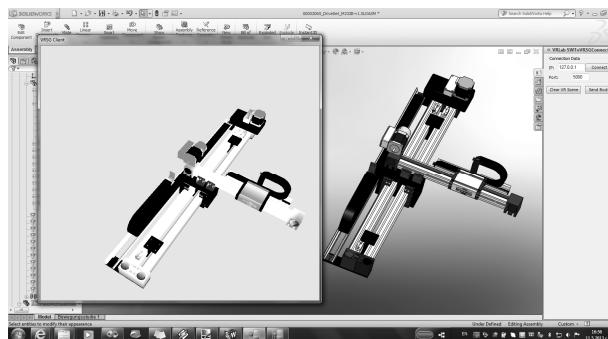


Fig. 8. A DriveSet M232B - a linear motion system part of DriveSet product family [14], transferred from Solidworks to a VR environment using the proposed method

5 Conclusion

In our opinion the here proposed method for CAD-VR model synchronization helps reduce the still existing gap in the integration of CAD and VR. The sample implementation and the performed tests have confirmed its applicability and as a result we can summarize its advantages as follows:

Using an extension native to the CAD application gives access to the full model description. After that the provided possibilities for editing the CAD model in VR depend on the completeness of the CAD description being transferred and the available VR manipulation and visualization modules and interfaces;

There is no difference if the model is modified in the CAD or the VR environment – the changes are propagated from one to the other in both directions the moment the change is made;

When a model is modified in the VR environment the calculation of the result is made by the CAD application and not by the VR application – this ensures that the result is the same as if the modification was made in the CAD application itself;

Avoids the need for implementation of CAD functions in VR or VR manipulation and visualization in CAD applications. Through the proposed method both applications can profit from the possibilities provided by the other.

Acknowledgments. The authors wish to thank for the support of the National Science Found at the Bulgarian Ministry of Education, Youth and Science received through grant DDBY02/67-2010.

References

1. Jayaram, S., Vance, J., Gadh, R., Jayaram, U., Srinivasan, H.: Assessment of VR Technology and its Applications to Engineering Problems. *Journal of Computing and Information Science in Engineering* 1, 83 (2001)
2. Jayaram, S., Wang, Y., Jayaram, U., Lyons, K., Hart, P.: A Virtual Assembly Design Environment. In: *Proceedings of the IEEE VRAIS Conference*, pp. 172–179 (1999)
3. Joshi, H., Jayaram, S., Jayaram, U., Varoz, L.: An Open Architecture for Embedding VR-Based Mechanical Tools into CAD Applications. In: *Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008* (2008)
4. Whyte, J., Bouchlaghem, N., Thorpe, A., McCaffer, R.: From CAD to virtual reality: modelling approaches, data exchange and interactive 3D building design tools. In: *Automation in Construction*, pp. 43–55. Elsevier (2000)
5. Steffan, R., Schull, U., Kuhlen, T.: Integration of Virtual Reality based Assembly Simulation into CAD/CAM environments. In: *Proceedings of the 24th Annual Conference of the IEEE, IECON 1998*, vol. 4, pp. 2535–2537 (1998)
6. Berta, J.: Integrating VR and CAD. *IEEE Computer Graphics and Applications* 19(5), 14–19 (1999)
7. Fiorentino, M., Monno, G., Uva, A.: CAD Interfaces in Virtual Reality: Issues and Solutions. Special Issue of: *Bulletin of The Transilvania University of Brasov* (2005)
8. Yang, J., Han, S., Park, S.: A method for verification of computer-aided design model errors. *Journal of Engineering Design* 16(3), 337–352 (2005)
9. Botsch, M., Kobelt, L., Pauly, M., Alliez, P., Levy, B.: *Polygon Mesh Processing*. A K Peters Ltd. (2010)
10. Bachvarov, A., Maleshkov, S., Boyadjiev, I.: Virtual Reality Based CAD-Model Exploration Method for the Design Engineering. In: *Proceedings of the International Conference on Information Technologies, InfoTech 2008*, vol. 2 (2008)
11. Kleiss, J.: Immersive Engineering. SCS (2010)
12. Kim, S., Weissmann, D.: Middleware-based Integration of Multiple CAD and PDM Systems into Virtual Reality Environment. *Computer-Aided Design & Applications* 3(5), 547–556 (2006)
13. Bachvarov, A., Maleshkov, S., Katicic, J., Stoyanova, P.: Design-by-the-Customer through Virtual Reality. In: *Advanced Research in Virtual and Rapid Prototyping. Advanced Research in Virtual and Rapid Prototyping*. CRC Press, Taylor and Francis Group, London (2010)
14. Zimmermann, P.: Virtual Reality Aided Design: A Survey of the Use of VR in Automotive Industry. In: Talabă, D., Amditis, A. (eds.) *Product Engineering: Tools and Methods Based on Virtual Reality*, pp. 277–296. Springer (2008)
15. Oancea, G., Gîrbacia, F., Nedelcu, A.: Software Module for Data Exchange Between Autocad and Virtual Reality Systems. In: Talabă, D., Amditis, A. (eds.) *Product Engineering: Tools and Methods Based on Virtual Reality*, pp. 383–394. Springer (2008)

16. Yue, C., Su, H.J., Alvarez, J.C., Ge, Q.J.: Enabling a High Fidelity Dynamics Simulation of CAD Assemblies in Virtual Environment for Machine Design. In: Proceedings of the ASME 2010 World Conference on Innovative Virtual Reality, WINVR 2010 (2010)
17. Barbieri, L., Bruno, F., Muzzupappa: Innovative integration techniques between Virtual Reality systems and CAx tools. *The International Journal of Advanced Manufacturing Technology*, 1085–1097 (2008)
18. Graf, H.: A “Change ‘n Play” Software Architecture Integrating CAD, CAE and Immersive Real-Time Environments. In: Proceedings of the 2011 12th International Conference on Computer-Aided Design and Computer Graphics, CADGRAPHICS 2011, pp. 3–10 (2011)
19. Raposo, A., Santos, I., Soares, L., Wagner, G., Corseuil, E., Gattass, M.: Environ: Integrating VR and CAD in Engineering Projects. *IEEE Computer Graphics and Applications* 29(6), 91–95 (2009)
20. Rabatje, R.: Integration of basic CAD functions into a VR environment. In: Proceedings of the Computer Graphics International, pp. 238–241 (1998)
21. Bennes, L., Bazzaro, F., Sagot, J.C.: Virtual reality as a support tool for ergonomic-style convergence: multidisciplinary interaction design methodology and case study. In: Proceedings of the 2012 Virtual Reality International Conference, VRIC 2012. ACM, New York (2012)
22. Bourdot, P., Convard, T., Picon, F., Ammi, M., Touraine, D., Vézien, J.-M.: VR-CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models. In: *Computer-Aided Design*, vol. 42(5), pp. 445–461. Butterworth-Heinemann Newton, MA (2010)
23. Guida, M., Leoncini, P.: Information-Preserving Procedural Translation of CAD Data to Dynamics-Simulated VR Environments. In: Proceedings of IDMME – Virtual Concept 2010, Bordeaux, France (2010)
24. Strauss, P., Carey, R.: An Object Oriented 3D Graphics Toolkit. In: *Computer Graphics Proceedings of SIGGRAPH 1992*, vol. 26, pp. 341–349. ACM (1992)
25. Maleshkov, S., Chotrov, D.: Affordable Virtual Reality System Architecture for Representation of Implicit Object Properties. *IJCSI International Journal of Computer Science Issues* 9(4(2)) (2012)

A Hand-Held 3-D Display System with Haptic Sensation

Kai Ki Lee¹, Kin-Hong Wong^{2,*}, Michael Ming-Yuen Chang¹, and Ying-Kin Yu

¹ Dept. of Information Engineering, The Chinese University of Hong Kong

² Dept. of Computer Science and Engineering, The Chinese University of Hong Kong

khwong@cse.cuhk.edu.hk

Abstract. We propose a projected based direct visual-haptic display system for virtual reality applications. In this work, we developed a hand-held 3-D display system in which the user can manipulate a 3-D object directly using his/her own hands and at the same time feels the weight and the movement of the displaying object. A projector mounted 1.5 meters above a handheld board is projecting images of virtual objects onto the board that the user is holding. A stationary camera is monitoring the board and the system calculates the pose of the board by computer vision techniques. So a motion-stereo effect can be generated by projected geometry methods for the user. Moreover, four motors are also pulling four strings attached to the corners of the board to generate the haptic feedback. Therefore, the user can feel the weight of the virtual 3-D object as well. The system is built and user tests were carried out with satisfactory results. This system has shown a new way of using projected reality, and provides a direct visual-haptic tool for developers to be used in interesting and useful virtual-reality applications.

Index Terms: Computer vision, Projected reality, Motion tracking, Haptic.

1 Introduction

In immersive virtual reality, there are two important issues: one is to create better user input control, such as the keyboard, mouse and magic-glove, etc. Another is haptic feedback, which allows the virtual system to return force feedback to the users. In this work, we propose a simple and reliable haptic display system built from low cost off-the-self devices. This paper will discuss a hand-held 3-D display system in which the user can manipulate a 3-D object using his/her own hands and simultaneously feels the weight and the movement of the displaying object. The system has a projector placed about 1.5 meters above a hand-held white board, while a computer-vision system is determining the pose of the board simultaneously. Then the system displays (through the projector) the corresponding 2-D images on the board depending on the pose to create a motion-stereo 3-D effect for the user. The board is also linked to a set of motors by wires attached to its four corners. As a result, the motors can send forces to the board to create the haptic effect to the one who is holding the board.

* Corresponding author.

By carefully coordinating the display and the forces created, we can simulate a situation where the user can manipulating a virtual 3-D object (placed on the handheld board) by his/her hands and feel the corresponding weight and motion.

The configuration of system is shown in Figure 1. The projector and camera are fixed on the top. Four motors are fixed at the four corners of the rig.

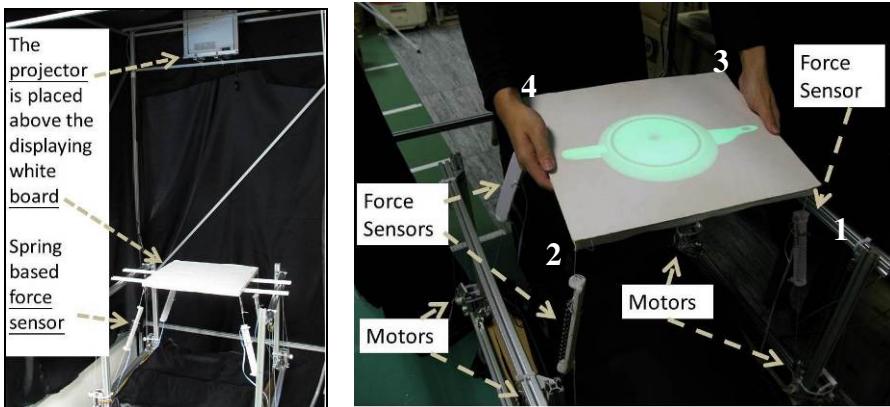


Fig. 1. Our proposed haptic system (left); A teapot is shown with our system (right)

In Figure 1, our proposed system can allow a user to view a 3-D virtual object in different angles while providing the user weight sensation through our haptic mechanism. The corners of the board are indexed as $i=1,2,3,4$ in Figure 1 (right).

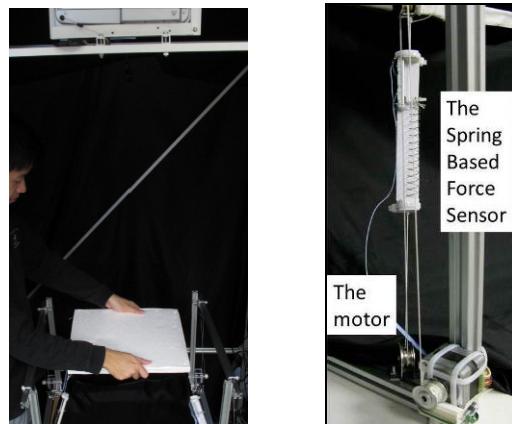


Fig. 2. The setup of our proposed system (left), the motor and spring-based force sensor (right)

Each motor is connected to one corner of the handheld displaying board via a string. The simulation of the weight of the displaying object is achieved via the dragging forces from the motors. According to the tracked position of the board, we adjust the force from these four motors to simulate the changing haptic sensation caused by

the movement of the board or the displaying object. In order to control the force along each string, we design a simple electrical force measurement device (spring-based force sensor) and attach it to the string. Based on the value measured by the force sensor, we adjust the motor to give the required tension on the string. As shown in Figure 3, the measuring device contains a sliding rheostat and a spring inside. By measuring the change of the resistance, we obtain the change of the force along the string. The whole system is based on off-the-shelf devices and the principle of the system is simple. We will explain the implementation details and evaluate the performance of the prototype system in following sections.

2 Related Work

Projector-based virtual reality and human machine interaction system is not new. The early versions include displaying static planar images corresponding to user input from keyboard and mouse. However, haptic feedback that includes force feedback is still relatively rare. That motivates us to research into building a projector-based immersive system that the user can feel the virtual object interactively.

2.1 Projector-Based System

Research into building projector-based augmented reality applications is always a popular topic. For example, the CAVE system [1] employs three rear projectors to project onto three side-walls of a cube-like room, creating a fully immersive virtual reality environment. As technology advances, the projectors are becoming smaller and lighter, making the system more economical and portable.

In a projector based human computer interactive (HCI) system, a camera is often used to track the motion of the user and the projector is responsible to display the corresponding images. For example, in [2], the Universal Media Book uses a computer vision approach to track a movable book and project pre-warped content onto it. Also Leung et al. [3] uses a projector to project appropriate content on a movable surface with a pre-calibrated projector-camera pair, Lee et al. [4] further improved it by carrying out image stretching and pre-warping to the projected image to create an immersive 3-D display.

2.2 Haptic Devices

Besides visual sensation, tactful sensation is an important for human. However, generating this kind of sensation accurately is not easy, and carefully designed mechanical structure is often required. It is worth mentioning that vibrating modules used in mobile phones and game machines are not able to produce authentic haptic effects. For example, in [5], a wearable haptic device is proposed to create the weight sensation of a virtual object. By deforming the finger-pad a user is wearing, this design presents realistic sensation of gravity to the user. In [6], a 7-DOF input device is proposed with haptic feedback, allowing the user to interact with virtual objects naturally by manipulating two hemispherical grips located in the center of the device frame. There is also

work which combines computer vision and haptic together to become an immersive system [7], however, the high building cost is the main drawback. Since the major components, such as projector, cameras and motors are not expensive at all at the time of writing, we believe it is possible to combine them to build a low cost and yet effective projector-based immersive system.

3 Design of the System

In conventional haptic interfaces, the user uses the hand to control a pointing device with force feedback to issue commands; however, the resulting effect is shown on a display unit located elsewhere. This is called indirect manipulation; the hand and eye are working separately at different locations and that is considered as unnatural. In an ideal scenario of direct manipulating, the user is acting upon an object where the visual and haptic effects are happening at the same spot. It is as if a virtual object is being manipulated and felt as if it is real. To provide the user with this direct manipulation experience, we propose a setup that simultaneously shows the visual content and force feedback in the same unit. In our proposed system, the user holds the cardboard in his hands, and different forces from the strings are applied onto the cardboard according to the position of the virtual object relative to the cardboard. A force calculation algorithm is designed to generate the force feedback to the user from the object being handled. During the operation, the user can move the cardboard to change the state of the virtual object and sense the force generated by the system. If programmed carefully, very interesting interactive games of handling moving objects (e.g. an animal such a moving cat) can be simulated. This may be the future research direction.

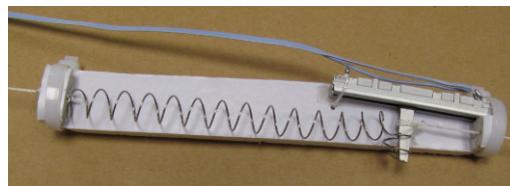


Fig. 3. Tension measuring device is composed of a spring and a sliding rheostat

3.1 Weight Display Unit

To allow the user to feel the weight of an object, we need a haptic interface that can generate a continuous downward force like the gravitational force. A string-based mechanism (see Figure 2) is designed to achieve this goal. The haptic interface proposed in this project uses four motor-string units that are placed at the lower corners to generate the required torque. A string from each motor is connected to one of the four corners of the visual display surface held by the user. The spring can serve as a tension feedback device, since the force F needed to scratch the spring is proportional to the extended length according to Hooke's law in equation (1).

$$F = -k\Delta x \quad (1)$$

where k is the elastic constant of the spring, Δx is the displacement.

The idea is based on the working principle of a spring scale used in our daily life. The length of the spring is monitored by a sliding rheostat (see Figure 3). When the spring extends, the slider of the rheostat will be moved to a new position that changes the resistance value. As the position is proportional to the resistance (2), we can measure the tension on the string by the following equations (3,4):

$$\Delta x \propto R \quad (2)$$

$$V_{force} = \frac{R}{R_{max}} V_{max} \quad (3)$$

$$V_{force} \propto F_{tension} \frac{V_{max}}{k \times R_{max}} \quad (4)$$

where R is the resistance of the sliding rheostat, V_{force} is the voltage on the rheostat, R_{max} and V_{max} are the maximum resistance and voltage respectively, $F_{tension}$ is the estimated force on the string.

The spring constant k can be determined by calibration with a known mass. Therefore, the system can measure the tension on the string by measuring voltage across the rheostat.

Moreover, the spring can serve as a passive element to generate continuous force on the visual surface (the displaying white board, see Figure 1). To explain the working principle of this motor-spring configuration, we divide the operation of the module into three modes. (1) Contraction mode: Consider a situation that we want to change the tension of the string from 2N to 10N. The force feedback module will enter the “contraction mode”, the motor rolls up the string until the tension reaches 10N. (2) Holding mode: In this “holding mode”, for example, a constant tension (e.g. 10N) is maintained. This state simulates the case when the user holds the board without any movement. (3) Releasing mode: If the user lifts up the visual surface then the tension increases because the spring is stretched by the user. The “releasing mode” unrolls the string to reduce the tension on the string until it reaches 10N again to simulate the weight of the object under gravity. For all modes, the motor rolling and unrolling action is controlled by a Proportion-Integral-Derivative (PID) controller to reduce overshoot and oscillation.

Although there is a slight time lag between the user action and the motor action, the motor-spring operation is still acceptable in producing the realistic feeling to the user of holding an object with weight. In addition, extra force is programmed to simulate upward acceleration when the displaying board is being lifted up.

3.2 Virtual Object Projection

In our projection system, the camera center is the world center. Each 3-D point in the space P is expressed as

$$P = [X \quad Y \quad Z \quad 1]^T \quad (5)$$

Each 2-D point on the projector image plane I is expressed as

$$I = [u \quad v]^T \quad (6)$$

The projector and camera do not share the same projection center. Even the pose between the board and the camera can be found, it is not enough to calculate the correct image projection (by the projector) of the virtual object onto the board. The reason is that there is a small transformation between the camera and the projector, which needed to be taken care of. The correction can be achieved by using a 3×4 matrix G_p described in Leung & Lee's paper [3]. The projection matrix G_p brings the 3-D point P to its correct image position I by using equation (7):

$$I \propto G_p P \quad (7)$$

In order to find the 3D space point on the board during runtime, we have to estimate the 3D pose of the tracked board. This can be done by applying the tracking algorithm mentioned in Lee's display system [2]. The board is detected and identified from the Hough diagram, and then a relative 3D pose to the board is estimated and tracked by particle filtering. A simple random walk model based on a uniform density describing the previous state is used in the dynamic model of the filter. To facilitate our discussion, we use a virtual 3-D ball rolling on a board to illustrate our idea. First, the estimated pose of the board is found by computer vision. Then the velocity v and horizontal acceleration a_x and vertical acceleration a_y of a free rolling ball on a board can be calculated using Newton's laws. The method is briefly described below.

The 3D corners \bar{P}_c of the cardboard can be represented by

$$\bar{P}_c = [P_{c1} \quad P_{c2} \quad P_{c3} \quad P_{c4}]^T. \quad (8)$$

$$P_{ci} = [X_{ci} \quad Y_{ci} \quad Z_{ci} \quad 1]^T \quad (9)$$

In equation (9), $[X_{ci} \quad Y_{ci} \quad Z_{ci} \quad 1]^T$ is the 3-D position of corner i , where $i=1,2,3$ and 4 (see Figure 1 for the locations of the corners). Since we do not attach any sensors onto the cardboard, the acceleration of the ball is calculated from the 3D corners \bar{P}_c . The acceleration of the ball along the x-axis of the 2D board is denoted as a_x in (10a), and the acceleration of the ball along the y-axis of the 2D board is denoted as a_y in (10b).

$$a_x \propto \frac{Z_{c2} - Z_{c1}}{\sqrt{(X_{c2} - X_{c1})^2 + (Y_{c2} - Y_{c1})^2}} \quad (10a)$$

$$a_y \propto \frac{Z_{c3} - Z_{c1}}{\sqrt{(X_{c3} - X_{c1})^2 + (Y_{c3} - Y_{c1})^2}} \quad (10b)$$

The tension force F of each string is calculated from the 2D location of the ball on the planar surface of the board, the mass m and the acceleration of gravity g by the Newton's Second Law of Motion in (11).

$$F = mg \quad (11)$$

After that, we need to create the corresponding projection image, which can be projected correctly onto the cardboard and observed by the user. This can be achieved by pre-warping the intermediate image into the projection image. From the tracking result of our system, we should have the relative pose between the camera and the cardboard at each frame. Then we can compute the 3D location of the four corners of the cardboard in 3-D. By using the calibrated projection matrix G_p as described in [3], we can find the four corresponding 2D points on the projection image plane according to equation (7).

Through the above procedures, the display content can be projected onto the cardboard correctly. The board will act as a light movable hand-held display device.

4 Experiment

Our proposed system is built with a projection unit and a force feedback unit. The computational unit is a computer installed with a 2.16GHz dual core processor and 1GB memory. The projection unit is a calibrated projector-camera pair which comprises of a DLP projector with resolution of 1280×1024 and a Logitech Quickcam Pro 4000 webcam with resolution of 320×240 . The force feedback unit is a haptic device, which comprises of four off-the-shelf stepping motors and four spring based force sensor designed by us. We use a white cardboard with $350mm \times 300mm$ dimension as our movable display surface. The webcam capturing rate is about 30 frame per second (FPS), and our average tracking updating rate is about 25 frame per second (FPS).

Three different user tests are conducted to evaluate the performance of the proposed haptic display system. We have invited nine people to test our system.

In the first test, a virtual cube is projected randomly on either left or right side of the display surface. Users need to close their eyes and determine the cube position from the treatment group test, which is with force feedback. For the control group test, there is no force feedback. The result is shown in Figure 4.

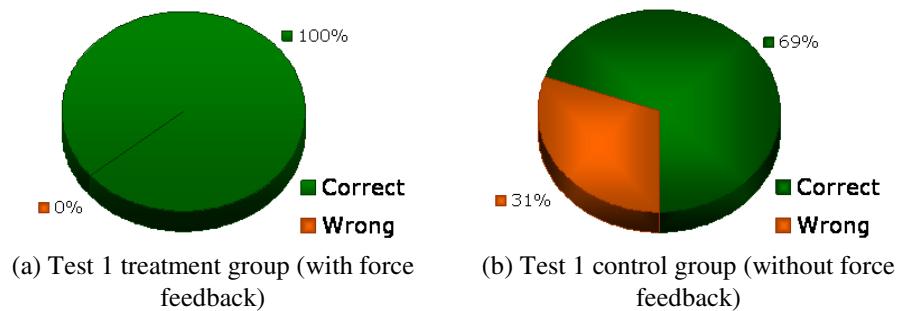


Fig. 4. . Results of User Test 1

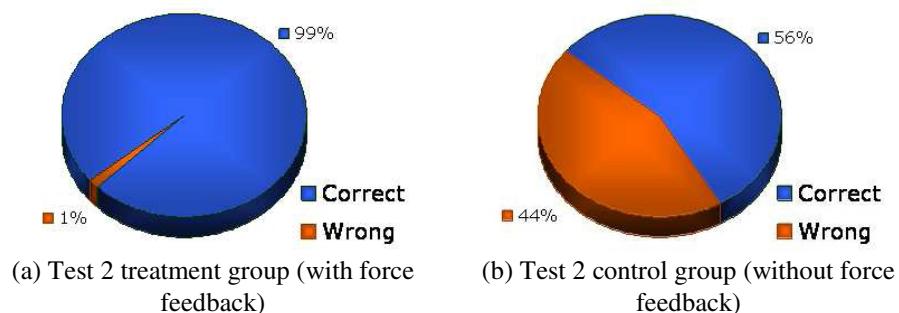


Fig. 5. Results of User Test 2

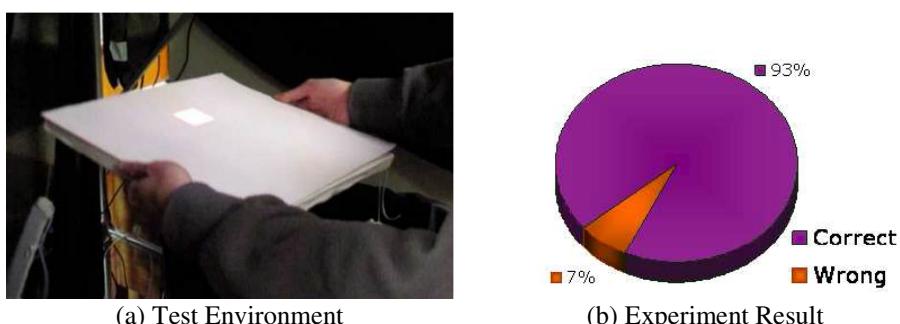


Fig. 6. Results of User Test 3

In the second test, the cube is projected randomly on either top or bottom side of the display surface. The result is shown in Figure 5.

In the third test, users need to distinguish a heavier virtual solid cube from a lighter virtual hollow cube in the virtual environment. The result in Figure 6 shows that most users are able to complete the task correctly.

5 Conclusion

In this paper, we have proposed a simple but effective visual-haptic display system made from low-cost devices. Our system consists of two parts, a projection module and a haptic sensation simulation module. The projection module displays the virtual object depending on the motion of the board while the haptic module simulates the haptic sensation of the moving object. The combination of the motion parallax and the haptic sensation gives the user a rich immersive experience.

We believe the proposed method can be applied to many Human Computer Interactive HCI systems for various interesting and useful applications. For example, we can develop an object balancing game that the user needs to stabilize an inverted pendulum. On the other hand, it may be used in a virtual shopping system that the customer can feel the weight of a certain product. We see a bright future of combining 3-D computer vision, projector and haptic feedback in building useful virtual reality systems.

Acknowledgement. The work described in this paper was supported by a direct grant from the Faculty of Engineering of the Chinese University of Hong Kong (Project No. 2050455). We would like to thank all the people who have helped us in this project: Hoi-Fung Ko, Zhaorong Lee, Simon Wong and Yibo Gong.

References

1. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A.: Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In: SIGGRAPH 1993: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 135–142. ACM, New York (1993)
2. Gupta, S., Jaynes, C.: The universal media book: tracking and augmenting moving surfaces with projected information. In: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 177–180. IEEE Computer Society (2006)
3. Leung, M.C., Lee, K.K., Wong, K.H., Chang, M.: A projector-based movable hand-held display system. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1109–1114 (2009)
4. Lee, K.K., Leung, M.C., Wong, K.H., Chang, M.Y.: A hand-held 3D display system that facilitates direct manipulation of 3D virtual objects. In: Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry, pp. 65–70. ACM (2009)
5. Minamizawa, K., Fukamachi, S., Kajimoto, H., Kawakami, N., Tachi, S.: Gravity grabber: wearable haptic display to present virtual mass sensation. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 Emerging Technologies, p. 8. ACM, New York (2007)

6. Kim, S., Hasegawa, S., Koike, Y., Sato, M.: Tension based 7-dof force feedback device: Spidar-g. In: Proceedings of the Virtual Reality, pp. 283–284. IEEE (2002)
7. Hashimoto, N., Jeong, S., Takeyama, Y., Sato, M.: Immersive multi-projector display on hybrid screens with human-scale haptic and locomotion interfaces. In: CW 2004: Proceedings of the 2004 International Conference on Cyberworlds, pp. 361–368. IEEE Computer Society, Washington, DC (2004)
8. Iwata, H., Yano, H., Tomiyoshi, M.: String walker. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 Emerging Technologies, p. 20. ACM, New York (2007)

Primitive Human Action Recognition Based on Partitioned Silhouette Block Matching*

Toru Abe¹, Masaru Fukushi², and Daisuke Ueda³

¹ Tohoku University, Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan
`beto@isc.tohoku.ac.jp`

² Yamaguchi University, Tokiwadai 2-16-1, Ube, Yamaguchi 755-8611, Japan
`mfukushi@yamaguchi-u.ac.jp`

³ SAXA, Inc., Miyashimo 3-14-15, Chuo-ku, Sagamihara 252-5221, Japan
`ueda.d@saxa.co.jp`

Abstract. This paper deals with the issue of recognizing primitive human actions through template matching with time series silhouette images. Although existing methods based on this simple approach can recognize a subject's action from a low-resolution image sequence, which is a basic requirement for surveillance applications, their recognition accuracy decreases considerably for corrupted silhouettes due to occlusion. To deal with this problem while keeping algorithm simplicity, we propose a novel method, which integrates template matching results for temporally and spatially partitioned silhouette blocks. Experimental results indicate that our method outperforms the existing methods in the accuracy of action recognition for corrupted silhouettes.

1 Introduction

Recognizing human actions from image sequences is an important issue in computer vision. Many methods have been proposed for various applications such as surveillance, control (i.e. controlling something by recognized actions), and analysis (e.g. orthopedic diagnosis from recognized actions) [1], [2], [3], [4]. In this paper, we focus on primitive human action recognition for surveillance applications.

While surveillance applications do not always need the precise locations and/or poses of subjects, those applications must work autonomously for long periods of time and recognize a subject's action within a permissible time from a low-quality image sequence. For the purpose of surveillance, methods have been proposed for recognizing primitive human actions through template matching with time series silhouette images [5], [6]. These methods suit surveillance applications for the following reasons: silhouettes of the subjects can be easily obtained from low-resolution image sequences; template matching can be implemented as a simple program competent for avoiding abnormal program termination and improving

* This work was supported in part by the Japan Society for the Promotion of Science (JSPS) under a Grant-in-Aid for Scientific Research (C) (No.23500201).

recognition speed. However, recognition accuracy of these methods decreases considerably for corrupted silhouettes due to occlusion. In the actual environment, corrupted silhouettes are often contained in input image sequences, since part of a human body is easily occluded by some obstacles such as guardrails, streetlights, and boxes.

To solve this problem while keeping algorithm simplicity, we propose a novel method for improving the accuracy of recognizing primitive human actions, which is based on template matching with partitioned silhouette blocks. Integrating the results of partial template matching for temporally and spatially partitioned silhouettes, our method can effectively use the temporal-spatial features in image sequences. Experimental results indicate that our method outperforms the existing methods in the accuracy of action recognition for corrupted silhouettes.

2 Action Recognition through Template Matching with Silhouette Images

To recognize primitive human actions from image sequences, Wang et al. have proposed a method based on template matching with time series silhouette images [5]. This method assumes that a subject is alone in each image sequence and his/her silhouettes can be extracted stably. Several approaches were examined in [5] for representing silhouette images (e.g. normal binary, distance transformed, and edge detected representations). In the following, we focus on the fundamental but most effective one; namely, normal binary representation.

As shown in Fig. 1, this method obtains silhouettes of a moving subject and regions of interest (ROI) from frames in an image sequence. These ROI are normalized so that they contain as much silhouettes as possible, keep aspect

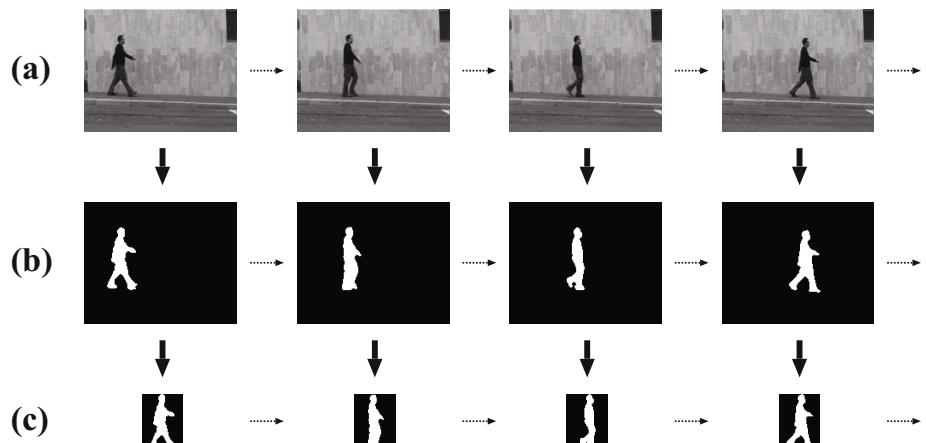


Fig. 1. Image sequence ‘walk’ [7]: (a) image frames, (b) silhouette images, (c) normalized ROI

ratio property, and have equal dimensions. The normalized ROI are represented as binary vectors, and thus an image sequence of T frames is represented as a vector sequence $\mathbf{V} = \{\mathbf{v}(1), \mathbf{v}(2), \dots, \mathbf{v}(T)\}$, where $\mathbf{v}(t)$ denotes the vector obtained from the t -th frame. To reduce the dimensions of vector representations, different techniques (e.g. principal component analysis: PCA, linear discriminant analysis: LDA, and locality preserving projections: LPP) were compared in [5].

Suppose that M types of actions A_m ($m = 1, 2, \dots, M$) are recognized from an input image sequence by using N reference image sequences of known actions. As shown in Eq. (1), let the input and reference image sequences be represented as vector sequences \mathbf{VI} and \mathbf{VR}_n ($n = 1, 2, \dots, N$),

$$\begin{aligned}\mathbf{VI} &= \{\mathbf{vi}(1), \mathbf{vi}(2), \dots, \mathbf{vi}(TI)\}, \\ \mathbf{VR}_n &= \{\mathbf{vr}_n(1), \mathbf{vr}_n(2), \dots, \mathbf{vr}_n(TR_n)\}.\end{aligned}\quad (1)$$

To recognize a subject's action in \mathbf{VI} , the sequence \mathbf{VR}_{n^*} closest to \mathbf{VI} is determined. The distance d between \mathbf{VI} and \mathbf{VR}_n is computed as the median Hausdorff distance by

$$d(\mathbf{VI}, \mathbf{VR}_n) = h(\mathbf{VI}, \mathbf{VR}_n) + h(\mathbf{VR}_n, \mathbf{VI}), \quad (2)$$

$$h(\mathbf{VI}, \mathbf{VR}_n) = \text{med}_{ti} (\min_{tr_n} \|\mathbf{vi}(ti) - \mathbf{vr}_n(tr_n)\|). \quad (3)$$

When $d(\mathbf{VI}, \mathbf{VR}_n)$ is minimal at $n = n^*$, the subject's action in \mathbf{VI} is recognized as the action corresponding to \mathbf{VR}_{n^*} .

As described above, this method uses silhouettes of a moving subject, which can be obtained readily from a low-resolution image sequence. Moreover, since template matching in this method is carried out for each silhouette image (binary vector), the normalization of action speed and duration is not necessary. Consequently, this method can be implemented as a simple procedure (i.e. template matching without any subject models), which is easy to avoid the abnormal procedure termination and improve recognition speed. However, results of this method are directly influenced by corruption of subjects' silhouettes. In the actual environment, silhouettes are easily corrupted by occlusion of the subject, which considerably decreases the accuracy of action recognition.

3 Template Matching with Partitioned Silhouette Blocks

3.1 Partitioned Silhouette Blocks

To cope with the decrease in action recognition accuracy for corrupted silhouettes, our method carries out template matching for temporally and spatially partitioned silhouette images.

Firstly, a silhouette image sequence is partitioned temporally, and sets of ROI are extracted. As shown in Fig. 2, for the t -th set of L consecutive $(t, t+1, \dots, t+(L-1))$ images, ROI of the same size is extracted from the same position in each image so that the silhouettes in the image set are circumscribed by the ROI set. The extracted ROI are normalized with respect to their dimensions. We refer to

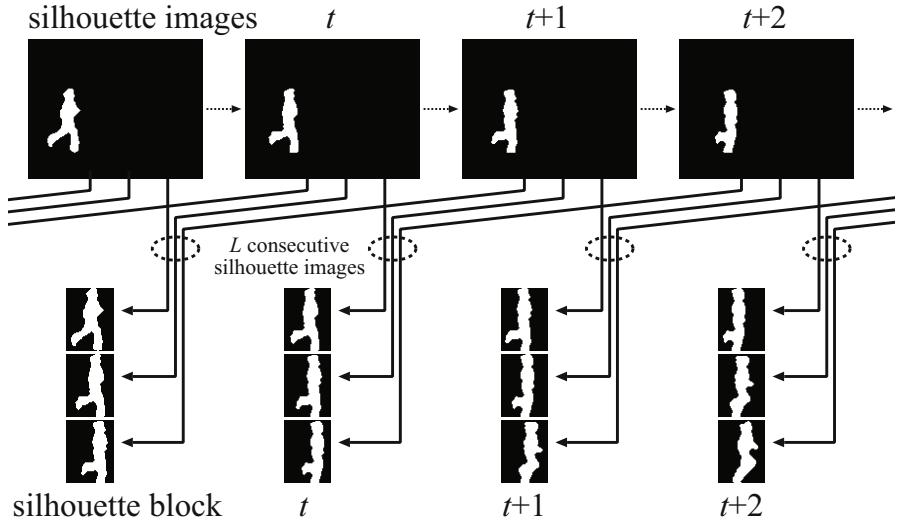


Fig. 2. Silhouette block from L consecutive images ($L = 3$)

this normalized ROI set as a silhouette block (SB). Let represent the t -th SB as a binary vector $\mathbf{v}(t)$ and the vector sequence obtained from an image sequence of T frames as $\mathbf{V} = \{\mathbf{v}(1), \mathbf{v}(2), \dots, \mathbf{v}(T - (L - 1))\}$. Secondly, as shown in Fig. 3, each SB is partitioned spatially into P partial blocks. Consequently, each $\mathbf{v}(t)$ and its sequence \mathbf{V} are also partitioned into $\mathbf{v}^{(p)}(t)$ and $\mathbf{V}^{(p)}$ ($p = 1, 2, \dots, P$). We refer to this $\mathbf{v}^{(p)}(t)$ as a partitioned silhouette block (PSB).

Our method converts input and reference image sequences into vector sequences $\mathbf{VI}^{(p)}$ and $\mathbf{VR}_n^{(p)}$. Between $\mathbf{VI}^{(p)}$ and $\mathbf{VR}_n^{(p)}$, distances $d(\mathbf{VI}^{(p)}, \mathbf{VR}_n^{(p)})$ are computed by Eq. (2). From those results of partial template matching, a subject's action in the input sequence \mathbf{VI} is recognized. Unlike a single

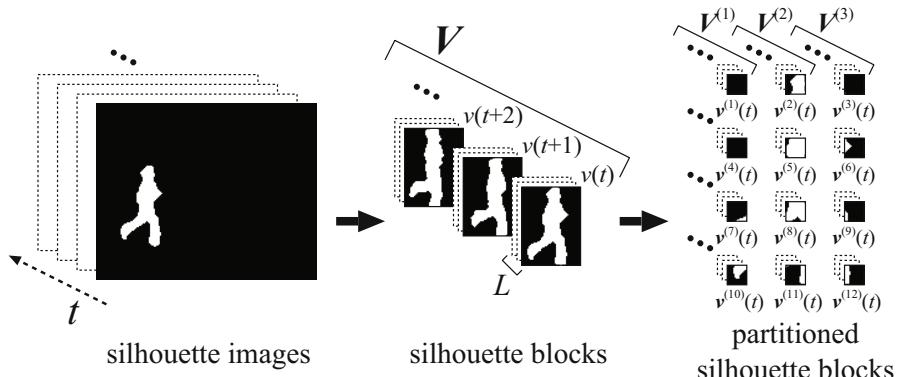


Fig. 3. Sequences of silhouette blocks \mathbf{V} and partitioned silhouette blocks $\mathbf{V}^{(p)}$

ROI, each SB can contain features of not only the subject's pose but also the temporally-localized subject's motion. If L ($L > 1$) is sufficiently-small, SBs are less affected by the difference in action speed and duration, whereas such SBs enable template matching to distinguish among different actions with similar poses (e.g. sitting down and sitting up). Furthermore, the subject's features are spatially localized in each PSB. Therefore, even if part of silhouettes in \mathbf{VI} are corrupted, some $\mathbf{VI}^{(p)}$ remain nearly unaffected and they are expected to be still effective for accurate action recognition.

3.2 Integrating Partial Template Matching Results

In our method, since template matching is carried out for each input vector sequence $\mathbf{VI}^{(p)}$ ($p = 1, 2, \dots, P$), these P partial matching results are integrated to recognize the subject's action in an input sequence. We propose three types of integration procedures. In the following, we suppose that reference vector sequences $\mathbf{VR}_n^{(p)}$ ($n = 1, 2, \dots, N$) of known actions A_m ($m = 1, 2, \dots, M$) are used in the procedures.

Proc. 1. To reduce the effect of corrupted $\mathbf{VI}^{(p)}$, the subject's action in an input sequence is recognized by voting. The sequence $\mathbf{VR}_{n^*}^{(p)}$ closest to each $\mathbf{VI}^{(p)}$ and the action (candidate) A_m corresponding to $\mathbf{VR}_{n^*}^{(p)}$ are determined. The subject's action is recognized as A_{m^*} which is the most frequent candidate for all $\mathbf{VI}^{(p)}$.

Proc. 2. The subject's action is recognized from the sum of the distances to all $\mathbf{VI}^{(p)}$. For every A_m , the minimum distance $d(\mathbf{VI}^{(p)}, \mathbf{VR}_n^{(p)})$ for each $\mathbf{VI}^{(p)}$ is determined, and they are summed up in D_m by

$$D_m = \sum_{p=1}^P \min_{\mathbf{VR}_n^{(p)} \in \mathcal{S}_m^{(p)}} d(\mathbf{VI}^{(p)}, \mathbf{VR}_n^{(p)}), \quad (4)$$

where $\mathcal{S}_m^{(p)}$ is a set of $\mathbf{VR}_n^{(p)}$ corresponding to the action A_m . The subject's action is recognized as $A_{m^{**}}$ whose D_m ($m = m^{**}$) is minimal for all m .

Proc. 3. Firstly, voting is carried out by Proc. 1. If the most frequent candidate A_{m^*} wins a majority in all $\mathbf{VI}^{(p)}$, then the subject's action is recognized as A_{m^*} . Otherwise, the subject's action is recognized from D_m by Proc. 2. This intends to exploit the advantages of both Procs. 1 and 2.

4 Experiments of Primitive Human Action Recognition

To demonstrate the effectiveness of our method, we conducted experiments of primitive human action recognition.

4.1 Image Sequences

The data set of image sequences opened to the public by [7] is used in the experiments. It contains 90 sequences (180 × 144 pixels, 25 fps), where nine different

subjects individually perform ten actions (walk, run, jump, gallop sideways, skip, bend, one-hand wave, two-hands wave, jump in place, and jumping jack). While subjects move horizontally in five actions (walk, run, jump, gallop sideways, and skip), they do not do that in the other five actions. As shown in Fig. 1, each sequence is converted to a series of silhouette images (those silhouette images are also included in the data set [7]).

As shown in Fig. 4, each silhouette block (SB) is extracted from L consecutive images. Their ROI are normalized to 48×64 pixels; therefore, the dimension of the vector $\mathbf{v}(t)$ representing each SB is $48 \times 64 \times L$. By partitioning every $\mathbf{v}(t)$

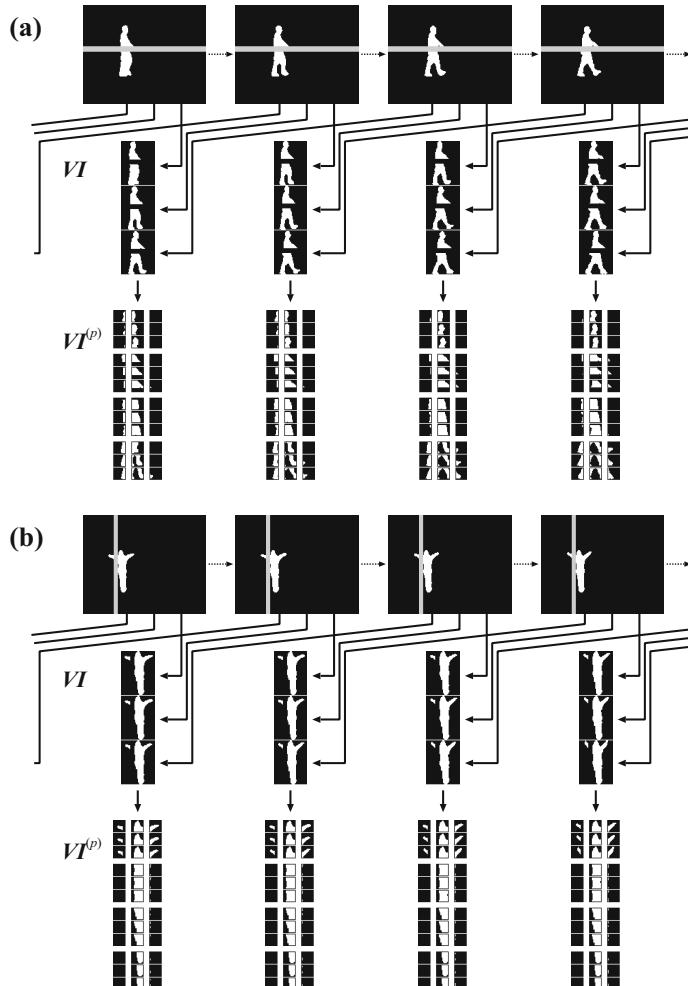


Fig. 4. Corrupted sequences, silhouette blocks \mathbf{VI} ($L = 3$), and partitioned silhouette blocks $\mathbf{VI}^{(p)}$: (a) 'walk' with a horizontal region, (b) 'two-hands wave' with a vertical region

spatially, 3×4 partitioned silhouette blocks (PSBs) $\mathbf{v}^{(p)}(t)$ ($p = 1, 2, \dots, 12$), each of which is $16 \times 16 \times L$ in dimension, are obtained. In [5], the dimensions of vector representations were reduced by using various techniques (e.g. PCA, LDA, and LPP) for improving recognition speed. In our experiments, since the evaluation of recognition accuracy is a primary concern, the dimensions of vector representations are not reduced in any methods; namely, raw vectors are used.

4.2 Experimental Condition

In the experiments, every image sequence is chosen from the data set as an input, and the other 89 sequences are used as references; therefore, eight references contain the same action as the input contains and the other 81 references do not.

While incorrupt image sequences are used as references, each image sequence chosen as an input is corrupted. To corrupt the input sequences as in the actual environment, silhouettes are occluded by superimposing two types of regions varying widths and positions. As shown in Fig. 4, the two types of regions, horizontal and vertical, are modeled on obstacles such as guardrails and streetlights, respectively. In some cases, despite the differences in widths and positions of superimposed regions, exactly the same corrupted sequences are generated. Such redundant sequences are excluded from the experiments. Furthermore, silhouette images without occlusion are removed from the sequences. By these procedures, 1667 input sequences (541, 565, 561 sequences for the region widths $w = 4, 8, 12$ pixels) and 1486 (453, 489, 544 for $w = 3, 6, 9$) sequences are generated with horizontal and vertical regions, respectively.

4.3 Experimental Results

Firstly, we carry out experiments to evaluate the effectiveness of template matching with SBs. Since each SB $\mathbf{v}(t)$ from a set of L consecutive images is not partitioned spatially, the method described in Section 2 is used for template matching. Therefore, when $L = 1$ (i.e., each SB consists of a single ROI), it corresponds to Wang's method with raw vectors [5]. The subjects' actions are recognized for all input sequences, i.e., their silhouette images are occluded by no (without occlusion), horizontal, or vertical regions.

The experimental results, the total recognition accuracy by varying L , are summarized in Fig. 5. As can be seen from Fig. 5, the accuracy decreases for corrupted sequences, particularly for silhouettes occluded by vertical regions. This reason comes from the fact that occlusion areas of vertical regions are mostly larger than those of horizontal regions for the same w because subjects' silhouettes are vertically long.

SBs of $L > 1$ obtain higher accuracy than those of $L = 1$. These results indicate that our method, which extracts each SB from more than one consecutive image without the normalization of action speed and duration, can effectively introduce features of the temporally-localized subject's motion into template matching. Since the differences in speed and duration of the subjects' actions

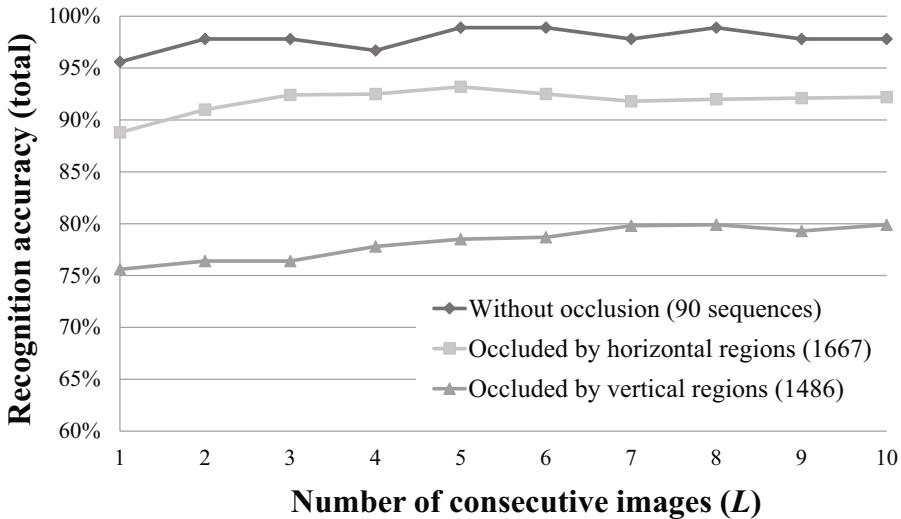


Fig. 5. Experimental results through template matching with SBs varying the number L of consecutive images

have significant influence on template matching with rather large L , the accuracy is not always proportional to L . However, SBs of sufficiently-small L ($L > 1$) are less affected by the difference in action speed and duration, whereas such SBs improve the accuracy of action recognition. This implies an interesting research issue; that is, there will be an appropriate number for L ($L > 1$). We leave this issue in future work.

Secondly, we conduct experiments to evaluate the effectiveness of template matching with PSBs. In the experiments, each SB $v(t)$ is extracted by $L = 3$, and partitioned into PSBs $v^{(p)}(t)$. To recognize the subjects' actions in input sequences, template matching is carried out with these $v^{(p)}(t)$, and their partial matching results are integrated by Procs. 1, 2, and 3.

Experimental results are summarized in Table 1. Because the areas of silhouettes corrupted due to occlusion are approximately proportional to w , as can be seen from this table, the recognition accuracy decrease considerably with w . In most cases, template matching with PSBs outperforms template matching with SBs in the recognition accuracy. These results show the effectiveness of our proposed method using PSBs, where subjects' features in the sequences are spatially localized, for the corruption of images due to occlusion.

In integrating partial matching results, Proc. 3 mostly provides higher recognition accuracy than Proc. 1 or 2. Compared to the results for SBs of $L = 3$ (without spatial partition), our method with Proc. 3 has 3.0 and 7.9 points improvements in the total recognition accuracy for input sequences corrupted by horizontal and vertical regions, respectively. Also, compared to the results for SBs of $L = 1$ (i.e. corresponding to Wang's method), our method has 6.6 and 8.7 points improvements.

Table 1. Recognition accuracy with SBs and PSBs

(a) Occluded by no regions (without occlusion)

Region width	(Input sequences)	(L = 1) L = 3	with PSBs, L = 3		
			Proc. 1	Proc. 2	Proc. 3
w = 0 pixels	(90)	(95.6) 97.8%	96.7%	98.9%	98.9%

(b) Occluded by horizontal regions

w = 4 pixels	(541)	(94.6) 97.0%	96.9%	98.9%	99.4%
w = 8	(565)	(90.3) 93.5%	94.3%	96.5%	96.5%
w = 12	(561)	(81.8) 87.0%	89.7%	90.4%	90.6%
Total	(1667)	(88.8) 92.4%	93.6%	95.2%	95.4%

(c) Occluded by vertical regions

w = 3 pixels	(453)	(92.3) 90.3%	90.9%	94.5%	95.4%
w = 6	(489)	(76.1) 78.5%	83.6%	83.6%	85.3%
w = 9	(544)	(61.4) 62.9%	75.7%	73.3%	74.1%
Total	(1486)	(75.6) 76.4%	83.0%	83.2%	84.3%

5 Conclusions

We propose a method for recognizing primitive human actions through template matching with time series silhouettes from low-resolution image sequences. Our method partitions silhouette images into temporal-spatial blocks and integrates partial template matching results with them, which improves the accuracy of action recognition by simple procedures (i.e. without the normalization of action speed and duration). The simplicity of our method will lead to low cost of implementation. Experimental results indicate that our method outperforms the existing methods in recognition accuracy especially for corrupted silhouettes.

To improve recognition accuracy, extracting each silhouette block from an appropriate number L of consecutive images and partitioning spatially each block into an appropriate number P of partial blocks (i.e. adjusting L and P for different types of actions) are promising strategies. In future work, we plan to investigate methods for achieving these strategies.

References

1. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding* 81, 231–268 (2001)
2. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding* 104, 90–126 (2006)

3. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Technol.* 18, 1473–1488 (2008)
4. Poppe, R.: A survey on vision-based human action recognition. *Image and Vision Computing* 28, 976–990 (2010)
5. Wang, L., Suter, D.: Learning and matching of dynamic shape manifolds for human action recognition. *IEEE Trans. Image Process.* 16, 1646–1661 (2007)
6. Weinland, D., Boyer, E.: Action recognition using exemplar-based embedding. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1–7 (2008)
7. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes (2007),
<http://www.wisdom.weizmann.ac.il/~vision/SpaceTime-Actions.html> (online; accessed May 1, 2013)

Fast and Accurate Unknown Object Segmentation for Robotic Systems

Lazaros Nalpantidis, Bjarne Großmann, and Volker Krüger

Robotics, Vision and Machine Intelligence Lab.,
Department of Mechanical and Manufacturing Engineering,
Aalborg University Copenhagen, Denmark
`{lanalpa,bjarne,vok}@m-tech.aau.dk`

Abstract. Object segmentation is the first step towards more advanced robotic behaviors, as robots need to localize objects before attempting tasks such as grasping or manipulation. A robot vision system should be able to provide accurate object hypotheses in reasonably high frame rates, using images and possibly also depth data. This work proposes a fixation-based object segmentation algorithm able to cope with unknown objects, and run on a real-time robot. We show that a balanced combination of moderately accurate, when considered independently, but at the same time computationally inexpensive building modules can yield remarkable results both in terms of accuracy, but also of execution speed. We describe our algorithm and present both qualitative and quantitative experimental results that indicate significant speed-up over the state-of-the-art.

Keywords: object segmentation, unknown objects, robot vision, Graph Cut, Grab Cut.

1 Introduction

One of the major high-level tasks of robot vision systems is the process of object detection and recognition. The ability to detect previously unseen or unknown objects in a household or office environment is the first step towards many sophisticated robotic behaviors [13]. A robot capable of generating object hypotheses through visual segmentation can use these for further interaction with the scene through pushing or grasping [12].

This work builds upon the recent work of Mishra et al. in active segmentation using fixation points [9,10]. A fixation point indicates a point on an object to be segmented, and can be considered as an intuitive human-robot interaction (HRI) paradigm for indicating an object to a system (e.g. through pointing, or using touch panels). The use of fixation points turns the ill-posed problem of finding a universal approach for image segmentation, into the tractable task of identifying the boundary of one object, i.e. the closed contour around the fixation point. More precisely, the authors rephrase the general segmentation problem as a binary labeling problem in polar space for a single object, and apply Graph Cut

[2]. Their results rely highly on the complex initial computation of a probabilistic boundary map, as it is the crucial first step for an accurate segmentation. However, due to the computational complexity of the probabilistic boundary map generation, the algorithm is difficult to be used in real-time systems.

The major *contribution* of the work in hand is to extend this approach in order to enable the new algorithm to run on a robotic real-time system. Therefore, the method proposed in this paper refines the ideas of the original approach by implementing several optimizations and additions. The basic idea is to reduce the computational load by combining much simpler building blocks. Thus, even if the proposed algorithm actually uses more stages, it ends up reducing the execution time by a factor of 7 compared to the original one.

This work shows that accurate segmentation results can be obtained by this algorithm even with the use of very simple edge detectors. What is needed in order to counterbalance the initially coarse results is one additional optimization step – the Grab Cut algorithm [14]. By using the output of the Graph Cut to initialize the Grab Cut, the framework of Mishra et al. becomes an intermediate step for computing an appropriate input mask. This allows to achieve high-quality results even for problematic segmentations made by the Graph Cut. As the Grab Cut is less dependent on the quality of the edge detection, the computational complexity of this process can be reduced drastically, hence boosting the overall performance of the object segmentation algorithm.

1.1 Considerations

The algorithm proposed by [9] gives good results only for simple objects without any “holes”, and furthermore cannot achieve practically useful frame rates for robotic applications. Clearly, the used “Globalized Probability of Boundary” (gPb) edge detection algorithm is the bottleneck of that segmentation algorithm, as it requires 6 seconds to compute on a quad-core 32-bit processor.

The method proposed in our work reduces the overall computational load by redistributing it more evenly among various modules. The idea is to find a suitable balance between quality and performance. Increasing the execution speed of the edge detection implies decrease of the quality of the edge map. That in turn raises the error ratio of the Graph Cut algorithm, especially for regions with blurry contours. However, the Grab Cut algorithm, used in the end to segment the image, can handle certain degree of errors in the edge map by using multiple iterations and sophisticated color models.

1.2 Related Work

Detecting objects in a scene has been a challenge since the beginning of computer vision. Even though the more general problem of image segmentation has received a lot of attention [4], segmentation of unknown object still constitutes an open issue among the computer and robot vision communities. The recent work of Mishra and Aloimonos [9,8] proposed the use of polar transformation

and then Graph Cut segmentation of objects depicted in single images, achieving very accurate results. This algorithm largely relies on accurate detection of edges in the image. The used gPb edge detector [7] is considered to be among the most robust and accurate edge detectors. It combines local information derived from brightness, color, and texture cues, with global information derived from spectral partitioning to provide response focusing on salient contours. Its main drawback is its computational complexity that results in long execution times. Even though, GPU-accelerated implementations of this detector have been reported [3], they demand highly specialized hardware.

Within the framework of Mishra et al., the possibility of including disparity or optical flow information has been also considered. On the other hand, the work of [11] proposes a simple way to combine information coming from sequences of images, gathered by mobile robots. Along the same path, the work of [6] is using the same underlying algorithm as the previous works, but also proposes a symmetry-based technique to choose suitable fixation points. Furthermore, the works of [5] and [1] perform accurate object segmentation, but again they rely on an initial rich 3D representation of the scene. Finally, GrabCut [14] is an efficient segmentation method, but it requires the definition of a coarse mask containing the object in order the segmentation process to be initialized.

2 Proposed Algorithm

The algorithm proposed in this work uses the Graph Cut segmentation results as an intermediate step and applies Grab Cut on them. This yields a new algorithm that doesn't involve computational bottlenecks, such as the gPb computation, and that can be used in robotic real-time systems. The block diagram of the proposed segmentation method is shown in Fig. 1.

The proposed algorithm consists of a number of modules. The input is a color image of a scene and optionally a depth or disparity map, when using 3D sensors,

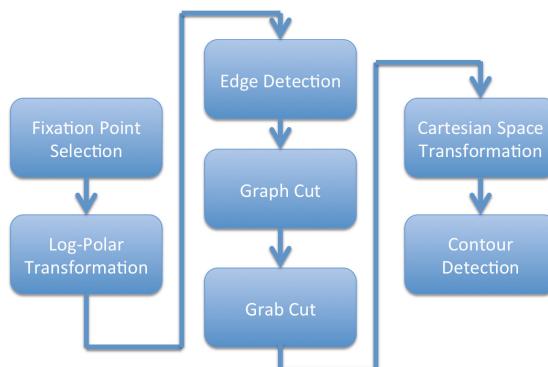


Fig. 1. Block diagram of the proposed object segmentation algorithm

such as the Microsoft Kinect or a stereo camera. First, the fixation point is selected. For what concerns our prototype implementation the user is choosing one (or more) point(s) in the images. Of course, other HRI or automatic approaches can be easily used instead for this step. The image is then transformed to the log-polar space. The log-polar space was used instead of the polar, as it not only blurs textures close to the fixation point, but also allows to generate the color models for the Grab Cut more precisely. That is because the log-polar representation increases the object region. Moreover, a growing kernel size following the increasing cell-size of the log-polar grid for the edge detection is implemented, thereby imitating an aspect of the human visual system: the blurred vision outside the focus. The edge detection computes an edge map from the given color and depth images. The edge detection is a simple Difference of Gaussians (DoG) kernel, applied both to the data of the color and the depth image. The results are then linearly combined into one edge map. Afterwards, the Graph Cut algorithm finds an approximate solution for the object's contour as a labeled map which is then used as the input for the Grab Cut algorithm.

Grab Cut employs color similarities to build up Gaussian Mixture Models. It uses the color- and depth-based edge map to calculate the weights of the n-links, and can handle certain degree of errors in the edge map. The segmentation is iteratively refined using the edge map and the built color model. The Grab Cut, being an extension of the Graph Cut, also profits from all the scale invariance feature of the log-polar space transformation. Finally, the processed image is transformed back in the cartesian space. The final result is the contour of the fixated object in the scene.

3 Experimental Validation

3.1 Hardware and Software Setup

The hardware setup used to evaluate the segmentation algorithm is important when the speed of execution needs to be estimated. Even though the time is not comparable to other implementations of segmentation algorithms, it can be a helpful indicator for the performance of the modules of the algorithm in relation to each other. All tests have been performed using a notebook computer with an *Intel Core i7-2630QM @ 2GHz* and 4 GB of RAM with a *Windows 7* 64-bit operating system. The camera device for video input was a *Microsoft XBOX 360 Kinect-Sensor* with an *RGB* camera and a depth sensor. The output of the camera and the depth sensor are respectively an *RGB* and a grayscale image with a resolution of 640×480 . Finally, the software of the proposed algorithm was developed in a modular way. The implementation of the modules made excessive use of the OpenCV library methods, especially in low-level algorithms.

3.2 Segmentation Results

The proposed segmentation algorithm is designed to detect objects that can be used in robot manipulation scenaria. Therefore, test scenes were set up with a



Fig. 2. Segmentation results of different objects. The “x” marks the chosen fixation point for each object respectively. The objects’ contours are marked in green.

variety of objects having different properties. The test results for various segmented objects in different scenes are summarized in Fig. 2.

Figure 2(a) demonstrates the capability of the algorithm to handle various objects. However, most objects in the scene are arranged in a way to generate strong depth information such that the contours are easier to extract. Therefore, another more casual scene is segmented in Fig. 2(b). Instead of using solid objects, this scene shows more soft and deformable objects which generate not only varying depth information but additionally have almost no depth difference at the contact boundaries.

The objects in Fig. 2(c) are elongated everyday objects, and an uncommonly shaped mug. Please note that the empty region in the center of the mug’s handle is identified as not being part of the object. This feature is due to the introduction of the Grab Cut algorithm. The original algorithm of Mishra et al. is meant to

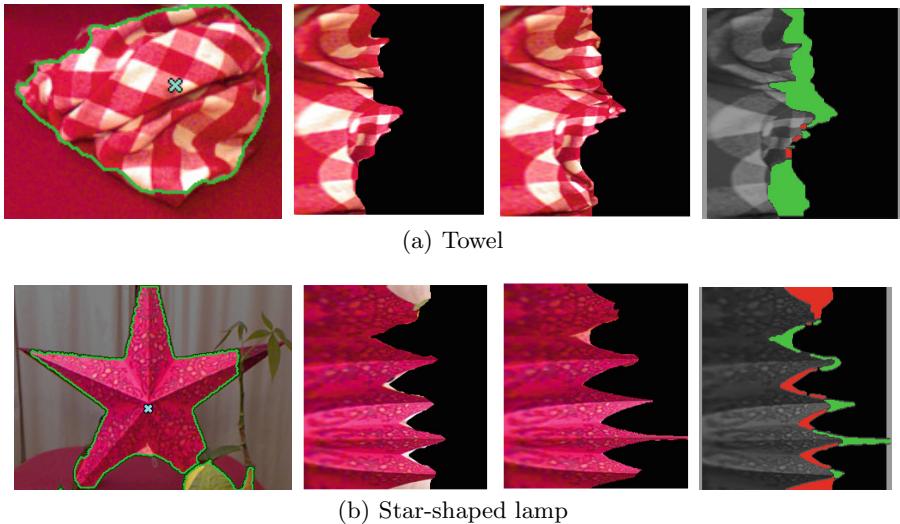


Fig. 3. Intermediate results and refinement of the Graph Cut results after applying the Grab Cut algorithm with four iterations. The 2nd column shows the Graph Cut results. The 3rd column shows the Grab Cut results. The rightmost column shows the added regions in green and the subtracted regions in red, as resulted from this refinement.

only deal with objects not containing any holes, and would have failed to identify this fine but important for grasping detail.

Elongated objects are more difficult to segment, especially using Graph Cut-based algorithms. When transformed to log-polar space, such objects are deformed to long and thin horizontal “spikes”, as they are bend around the fixation point. However, the Graph Cut algorithm tries to find a short vertical cut to minimize the energy. Therefore, long objects and objects with sharp edges tend to get cut off. The Grab Cut algorithm suffers from the same problem, but it can counterbalance this behavior by using color information. In fact, the use of Grab Cut within our method was motivated because of exactly this feature. It allows to optimize the result of the Graph Cut algorithm, especially in these cases, as shown in Fig. 3.

Some elucidating intermediate results of the operation of the proposed algorithm are also given in Fig. 3. Figure 3(a) focuses on the dish towel of Fig. 2(a), while Fig. 3(b) focuses on the star-shaped lamp of Fig. 2(d). The 2nd column shows the Graph Cut results. The 3rd column shows the Grab Cut results. Finally, the rightmost column shows the added regions in green and the subtracted regions in red, as resulted from the refinement. The final segmentation can be seen overlaid on the initial images in the leftmost column.

3.3 Execution Speed

The computational load of the presented algorithm is an important factor for its use in a robot system. Within this context, the object segmentation module should be able to achieve frame rates that allow smooth robot operation, avoiding long pauses for processing.

The duration of the processing varies depending on the object and the environment. Moreover, the measured time for the segmentation process depends much on the hardware used. Therefore, the evaluation presents the absolute duration of the algorithm on the setup described in section 3.1. Besides that, it also compares the relative durations of the different steps of the algorithm which are then compared to the relative durations of the original approach by [9].

The measurements are taken for the whole algorithm as well as for each step: the edge detection, the Graph Cut and the Grab Cut algorithm. Table 1 shows the average values of ca. 100 segmentations on a medium sized object using a CPU implementation of our algorithm.

The whole algorithm takes about six seconds to compute the segmentation of an object on the used notebook computer. These timing results are expected to be reduced at least by an order of magnitude with a GPU accelerated version of the same algorithm. The result of the optimization of the edge detector has to be pointed out in particular. In this approach, it takes less than half a second to compute the image and depth edge map. The most time consuming steps are the Graph Cut and the Grab Cut algorithm. The Graph Cut has the most work to do, as the minimum cut calculation is based on mostly unknown labels. In total, the Grab Cut takes longer than the Graph Cut. However, it has to be taken into account that the Grab Cut's total time includes four iterations, including the learning process of the Gaussian Mixture Models. This means that a single iteration of the Grab Cut algorithm takes less than a second. It performs much better than the single Graph Cut as the input mask is already an approximation of the final result.

The algorithm mainly is a single thread application (except for the image grabbing routine) and uses only 21% of the CPU capacity. The memory consumption varies around an average of 100 MB, when, besides the video player and the output window, no additional observation window is opened. The low CPU consumption is a good starting point for further optimizations.

Table 1. Measured absolute and relative durations for the different steps for a CPU implementation of the proposed algorithm

Module	Duration (sec)	Duration (%)
Edge Detection	0.40	6.38
Graph Cut	2.27	36.20
Grab Cut	3.46	55.18
Cleaning	0.14	2.24
Overall time	6.27	100.00

Table 2. The indicated duration for the different steps of the algorithm proposed by [8]

Module	Duration (sec)	Duration (%)
Edge Detection	6.00	15.00
Graph Cut (x2)	2.00	5.00
Optical flow	24.00	60.00
Others	8.00	20.00
Overall time	40.00	100.00

As the proposed algorithm is an optimization and extension of the algorithm described by [9], a comparison between the two of them is evident. A direct comparison is difficult, however by taking the Graph Cut algorithm as a reference point, a comparison can be made to roughly classify this framework. Therefore, the given duration of the original algorithm, as described in [8], are shown in Table 2.

The Graph Cut algorithm is computed twice in the algorithm. The second time, a 3D-histogram is used to incorporate the color information into the Graph Cut algorithm. Therefore, it seems that the Graph Cut implementation takes about one second to compute on the used computer system, as described in Sec. 3.1. By using the Graph Cut duration as a base unit, it is possible to compare the two implementations, as shown in Table 3.

The comparison of the normalized durations in Table 3 leads to the conclusion that the implementation of the proposed algorithm runs more than seven times faster than the algorithm proposed by [8]. Even though this calculation depends on many unknown variables, at least the edge detection can be roughly compared, as the implementation is the same as in [9]. The performance gain for the edge detection is even higher: the edge detector is more than 16 times faster than the original approach. Using a faster desktop computer and a GPU implementation of certain parts of our algorithm, i.e. edge detection, Graph Cut computation, real time response can be obtained.

Table 3. The duration of the implementations as multiples of the Graph Cut duration and as a factor of the performance gain for the proposed implementation

Module	Normalized Duration		Improvement Ratio
	Proposed	Mishra et al.	
Edge Detection	0.18	3.00	16.67
Graph Cut	1.00	1.00	1.00
Other	1.58	16.00	10.13
Overall time	2.76	20.00	7.25

4 Conclusion and Discussion

The algorithm proposed in this work incorporates several optimizations and extensions over the algorithm presented by [9]. It involves an additional Grab Cut module for the segmentation result given by the original approach, which thereby becomes an intermediate step of the new proposed algorithm. This leads to a more balanced process, as the final result rely less on the edge detection. By distributing the liability more evenly across the modules of the algorithm, a foundation for supplementary optimization processes can be established, as minor errors occurring in the intermediate steps can be absorbed by the following Grab Cut algorithm.

The execution time is reduced and the algorithm can now be used in real-time robot systems. As the developed algorithm is very robust to erroneous intermediate results, the edge detection step could be simplified a lot. This allowed for the reduction of the computational complexity of the whole algorithm, while the high quality of the algorithm's results remained practically unchanged.

The polar space representation for the contour retrieval is exchanged in favor of the log-polar space. Besides the useful scale invariance property of the polar space, this transformation has two additional advantages: On the one hand, the image gets smoothed near the object of interest which results in less frequent intensity changes of the texture in the object region. This enhances the results of the edge detection, as it will find less edges near the object caused by texturing. On the other hand, the object region in log-polar space is much larger than in polar space. This enhances the results of the Grab Cut algorithm, as more pixels are included to build the color models which are used to refine the object's contour.

An interesting feature of the proposed algorithm was its ability to identify "holes" on objects. The empty part of the handle of the mug in Fig. 2(c), commonly used for grasping, was correctly identified as background and was not part of the object segmentation. This characteristic is owed to the the added Grab Cut module, and is not easily achievable without it.

Even though the Grab Cut algorithm is able to refine the contours of the objects, it still has its limits. If the object is too thin, i.e. the edges of the boundary are too close, this method will not work, as seen in the segmentation of the plant with the thin stem and the endings of the spikes of the star-lamp. The difficulties arising when segmenting objects with thin or spiked shapes are caused by multiple reasons. First, the resolution of the depth map is not sufficient to detect very thin objects. Last, elongated objects are deformed during the log-polar transformation getting unfavorable shapes for the graph-based algorithms.

Acknowledgements. This work has been supported by the European Commission research project "Robotics-enabled logistics and assistive services for the transformable factory of the future (TAPAS)", FP7-ICT-260026.

References

1. Björkman, M., Kragic, D.: Active 3D segmentation through fixation of previously unseen objects. In: Proceedings of the British Machine Vision Conference, pp. 119.1–119.11. BMVA Press (2010)
2. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 359–374 (2001)
3. Catanzaro, B.C., Su, B.Y., Sundaram, N., Lee, Y., Murphy, M., Keutzer, K.: Efficient, high-quality image contour detection. In: International Conference on Computer Vision (ICCV), pp. 2381–2388 (2009)
4. Illea, D.E., Whelan, P.F.: Image segmentation based on the integration of colour-texture descriptors - A review. *Pattern Recognition* 44(10-11), 2479–2501 (2011)
5. Johnson-Roberson, M., Bohg, J., Björkman, M., Kragic, D.: Attention-based active 3D point cloud segmentation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1165–1170 (October 2010)
6. Kootstra, G., Bergström, N., Kragic, D.: Fast and automatic detection and segmentation of unknown objects. In: IEEE-RAS International Conference on Humanoid Robots, pp. 442–447 (2010)
7. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (June 2008)
8. Mishra, A., Aloimonos, Y.: Visual segmentation of simple objects for robots. In: Robotics: Science and Systems, Los Angeles, CA, USA (June 2011)
9. Mishra, A., Aloimonos, Y., Fah, C.L.: Active segmentation with fixation. In: IEEE International Conference on Computer Vision, pp. 468–475 (2009)
10. Mishra, A.K., Aloimonos, Y., Cheong, L.F., Kassim, A.: Active segmentation with fixation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(4), 639–653 (2012)
11. Nalpantidis, L., Björkman, M., Kragic, D.: Yes - yet another object segmentation: exploiting camera movement. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Algarve, Portugal (2012)
12. Rasolzadeh, B., Björkman, M., Huebner, K., Kragic, D.: An active vision system for detecting, fixating and manipulating objects in real world. *International Journal of Robotics Research* 29(2-3), 133–154 (2009)
13. Richtsfeld, A., Morwald, T., Prankl, J., Zillich, M., Vincze, M.: Segmentation of unknown objects in indoor environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4791–4796 (2012)
14. Rother, C., Kolmogorov, V., Blake, A.: “GrabCut”: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* 23(3), 309–314 (2004)

Differential Progressive Path Tracing for High-Quality Previsualization and Relighting in Augmented Reality

Peter Kán and Hannes Kaufmann

Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria
peterkan@peterkan.com, kaufmann@ims.tuwien.ac.at

Abstract. In this paper we present a novel method for real-time high quality previsualization and cinematic relighting. The physically based Path Tracing algorithm is used within an Augmented Reality setup to preview high-quality light transport. A novel differential version of progressive path tracing is proposed, which calculates two global light transport solutions that are required for differential rendering. A real-time previsualization framework is presented, which renders the solution with a low number of samples during interaction and allows for progressive quality improvement. If a user requests the high-quality solution of a certain view, the tracking is stopped and the algorithm progressively converges to an accurate solution. The problem of rendering complex light paths is solved by using photon mapping. Specular global illumination effects like caustics can easily be rendered. Our framework utilizes the massive parallel power of modern GPUs to achieve fast rendering with complex global illumination, a depth of field effect, and antialiasing.

1 Introduction

Inserting synthetic objects into real videos is required by many real-time applications in computer graphics. Accurate offline algorithms which are capable of producing images indistinguishable from reality were developed. These methods are based on mathematical models describing light transport. Computationally expensive calculations are required to produce a full solution of global light transport involving multidimensional integration which is described by the rendering equation [1]. Therefore physically based algorithms have not been suitable for real-time applications. Applications including cinematic relighting, movie previsualization, and others can benefit from real-time light transport computation.

Direct illumination is traditionally used for the real-time preview of mixed virtual and real scenes. However, the reflected indirect light component is missing in the rendering. We propose a rendering framework using a physically based algorithm to render the composited video in preview quality during interaction. In addition it supports progressive refinement to converge to the full solution. Modern GPUs are employed to increase the speed of the path tracing algorithm.

We use an Augmented Reality (AR) scenario to allow users to interact with virtual objects inserted into the real world. This scenario can be especially useful during movie production where virtual and real content is mixed. A novel one-pass differential progressive path tracing algorithm is introduced which quickly calculates two illumination solutions needed for compositing. Our framework operates in two main modes allowing interaction and high-quality convergence: An interactive preview mode and a progressive refinement mode. The problem of noise in the interactive preview mode is solved by allowing users to increase the quality of the result by increasing the sampling rate. Users can switch to the progressive refinement mode any time to see the full quality solution (Figure 1).

Rendering synthetic objects into a real scene requires the estimation of real lighting. For this purpose we use a camera with a fish eye lens to capture the environment illumination. Two lighting algorithms are available in our framework: (1) Light source estimation by processing an environment map or (2) image based lighting, where the whole captured environment map is used to light the scene. In our rendering framework a physically based camera model with finite sized aperture is used, which enables the simulation of a high-quality depth of field effect (Figure 1). Difficult light paths needed to generate caustics are hard or in some cases impossible to simulate by path tracing. We overcome this problem by using Photon Mapping to handle these light paths separately (Figure 2). Thus our framework is capable of simulating complex global lighting between real and virtual scenes. Our framework naturally supports reflection and

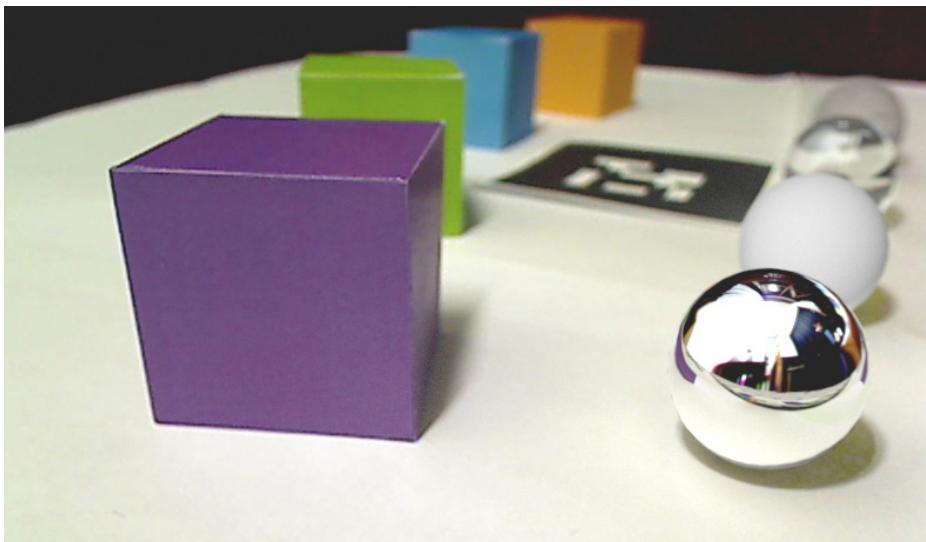


Fig. 1. Converged AR image in progressive refinement mode. Cubes on the left side of the table are real and spheres on right are virtual. The converged image was progressively rendered within 1 minute. Note the similar depth of field effect on real cubes and virtual spheres. The real environment is correctly reflected in the virtual metal ball.

refraction on specular surfaces. Moreover antialiasing can be enabled by supersampling an area of pixels and it is inherently supported in progressive refinement mode. Artificial lights can be added to the rendering to support artistic lighting situations in the process of movie production. If a high quality result is needed, the captured video, lighting and camera position can be recorded during interaction. Afterwards the scene can be rendered in high-quality in post-processing to create the full movie quality.

Our main contributions of this paper are the novel differential progressive path tracing algorithm and the overall framework for previsualization and cinematic relighting. Differential progressive path tracing utilizes the massive parallel power of GPUs and produces a real-time preview of global light transport. It progressively renders a fully converged image within a short time. Our interactive framework uses this rendering algorithm together with photon mapping to simulate the full range of Global Illumination (GI) effects.

2 Related Work

The simulation of global illumination in AR is required for proper lighting of virtual objects and for achieving visual coherence between virtual and real scenes. Several methods were presented in previous research to solve this problem and to calculate the interreflections between real and virtual worlds. In this section we refer to recent research about global illumination in AR and we mention the approaches for previsualization and relighting.

2.1 Global Illumination in Augmented Reality

A pioneering work in light transport for AR was presented by Fournier [2]. It was later extended by Debevec [3] using an image based lighting approach to simulate the natural appearance of virtual objects. Caustics and accurate refractions for AR were first presented in differential photon mapping [4]. Recently high-quality augmentations of photographs with virtual objects were presented [5]. This work is similar to ours in the sense that it calculates the physically based lighting solution and it uses a user driven approach. All mentioned approaches require offline calculation. The advantage of our work is that we present a real-time approach allowing interaction of a user with the scene and we include the automatic light source estimation.

Recently rasterization based solutions for light transport in AR were presented. These solutions can achieve real-time performance with the cost of an approximation error. The work of Knecht et al. demonstrates diffuse global illumination in AR by using Differential Instant Radiosity [6]. This approach achieves real-time performance, but is limited to diffuse global illumination. Recently Lensing and Broll [7] proposed a method for calculating fast indirect illumination in mixed scenes with automatic 3D scene reconstruction. The disadvantages of both approaches are that they introduce approximation errors and are limited to diffuse global illumination. Our approach can calculate the

full range of global illumination effects. In our solution the error is introduced in the form of noise, which can be reduced by increasing the sampling rate. After request the solution quickly converges to the noise-free image. A real environment map for image-based lighting of virtual objects was used in [8,9]. The disadvantage of these techniques is that they ignore visibility in the irradiance calculation and therefore introduce an approximation error. In our solution we calculate interreflections between virtual and real worlds in both ways in a physically based fashion. We solve the visibility problem accurately by ray-tracing. Near-field illumination in AR was correctly simulated by Grosch et al. [10]. The disadvantage of their method is that a vast precomputation of irradiance is required and big objects have to be static to not change the irradiance. The advantage of our method is that no precomputation is required and we support fully dynamic geometry, materials and lighting.

2.2 Previsualization and Cinematic Relighting

Real world videos mixed with computer generated content are often used in movie production. Previsualization solutions are used on set while shooting the film to see a real-time preview of the final compositing of virtual and real worlds. A previsualization system for filmmaking using MR was proposed by Ikeda et al. [11] and Northam et al. [12]. The authors used rasterization-based rendering and did not calculate any light transport between real and virtual worlds. A good overview of previsualization techniques can be found in [13].

Systems for cinematic relighting handle a huge amount of geometry and produce an image with global light transport. A ray-traced occlusion caching of massive scenes for cinematic relighting was used by Pantaleoni et al. [14]. Sparse directional occlusion caches were precomputed in their system to accelerate a lighting pipeline working in the spherical harmonics domain. A radiosity caching algorithm to preview the final quality in movies was proposed by Christensen et al. [15]. Authors used three resolutions of radiosity and the appropriate resolution was chosen depending on the ray differentials during rendering. The mentioned approaches for cinematic relighting focus on rendering virtual content. None of them addresses the combination of real and virtual objects. Our solution can be used both for previsualization and relighting in movie production.

3 Differential Progressive Path Tracing

The core of our framework is the differential progressive path tracing algorithm running on the GPU. This algorithm uses Monte Carlo integration in ray-tracing to evaluate the global light transport in a mixed reality scene. Two solutions of light transport are needed for differential rendering to composite the final image. The first solution is the light transport in a real scene only and the second global illumination solution is within a mixed scene taking into account the geometry of both real and virtual objects. These GI results are composited with the image taken from the camera using the following equation [3,5]:

$$L_{of} = M \odot L_{om} + (1 - M) \odot (L_{oc} + L_{om} - L_{or}) \quad (1)$$

L_{of} denotes the radiance of the final composite image for one pixel. This radiance is tonemapped to fit the low dynamic range (LDR) displaying capabilities of current displays. L_{om} is the mixed radiance result where both real and virtual objects are used. L_{or} is the lighting solution that only uses real objects. The term M denotes the mask, which defines the amount of blending between virtual objects and the real world. The differential radiance $L_{om} - L_{or}$ is added to the radiance obtained from the camera image L_{oc} . This difference presents the changes in lighting caused by adding virtual objects. Inverse tonemapping is applied to the captured LDR camera image to obtain the high dynamic range (HDR) value L_{oc} .

We use one pass algorithm to calculate both mixed and real radiance together. Four ray types are used to enable the calculation of two rendering solutions: **a mixed radiance ray, a real radiance ray, a mixed shadow ray, and a real shadow ray**. The mixed radiance ray returns both mixed and real radiances while the real radiance ray calculates only real radiance. The mixed radiance ray type is always used in primary rays shot from the camera, because both mixed and real radiances have to be evaluated. The ray type can change if the ray intersects geometry. Our algorithm is described in detail in [16].

To calculate the light exiting from a scene point x in a direction ω_o towards the camera, the integration of incoming light at this point is necessary. The integral equation can be written as [1]:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_o, \omega_i) L_i(x, \omega_i) |n \cdot \omega_i| d\omega_i \quad (2)$$

where L_o is the outgoing radiance coming from point x to direction ω_o , L_e is the radiance emitted from point x towards direction ω_o and L_i is radiance of incident light incoming from direction $-\omega_i$ to point x , while the incoming light radiance is integrated on the hemisphere Ω . f_r denotes the BRDF function of the surface at position x , with surface normal n . Monte Carlo integration can be used to solve the recurrent rendering equation. The recursive path tracing algorithm [1] is used to perform this numerical integration. In original differential rendering the integral in Equation 2 has to be evaluated two times to be used in the compositing equation (Equation 1). Our one pass differential rendering algorithm can produce both solutions together. The algorithm starts by shooting the mixed radiance rays from the camera towards the scene. The numerical integration is performed on the hitpoints of primary camera rays with geometry to evaluate the contribution of light reflected from a surface point. The random light paths are sampled from this point and the visibility of a light source is tested in every hitpoint to calculate the light contribution. Mixed radiance rays will evaluate both mixed and real radiances which are needed for the differential rendering equation. The user can set up the number of samples taken during interaction to control the performance and quality of the live preview. If a user requests the full quality image, the progressive algorithm starts to run and all samples are averaged to obtain the final value.

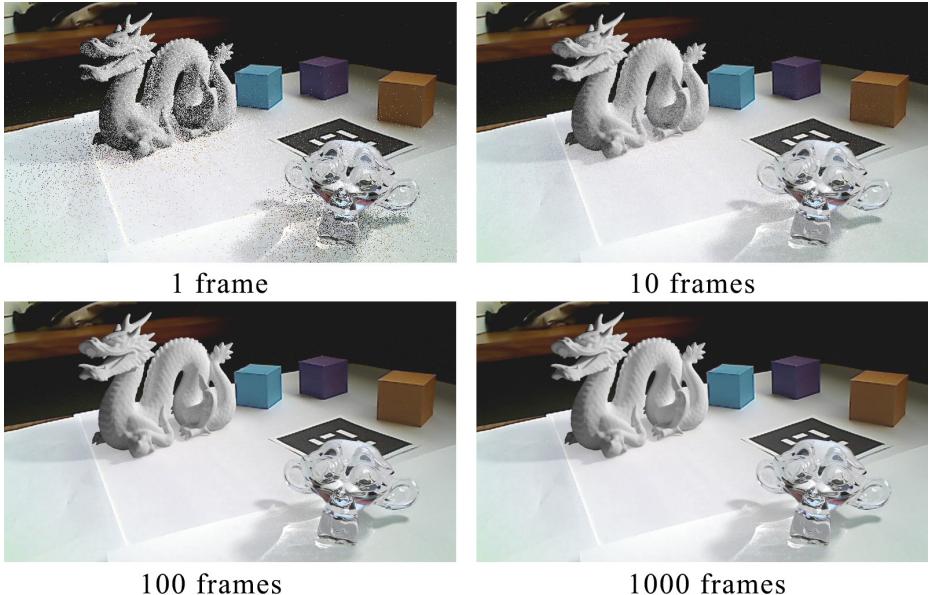


Fig. 2. Rendering by differential progressive path tracing. The scene contains a virtual monkey and a virtual dragon (27K triangles). The solution after 1, 10, 100 and 1000 frames of refinement is shown. Caustics are simulated by photon mapping.

When the user requests to see the converged AR image, the system stops tracking and camera capturing and progressively converges to a high-quality image without noise. The last camera image, pose and lighting is used for progressive rendering. Newly calculated frames are accumulated to the accumulation buffer and the average solution of all samples is progressively calculated. The new calculated radiances are blended with the accumulated values:

$$L_{acc} = \frac{(n - 1)L_{acc} + L_{of}}{n} \quad (3)$$

where L_{acc} stands for the accumulated value. This value is calculated as an average of all samples of previous frames. L_{of} is the result of the compositing equation of the differential rendering algorithm (Equation 1) and n is the number of frames in progressive rendering mode.

The light transport calculation handles all radiance values as float numbers in HDR. The tonemapping operator is applied to the accumulated value to convert HDR to LDR values for displaying the composited frame on the screen. We use an inverse tonemapping operator to convert LDR camera image into HDR radiance [17]. Differential progressive path tracing is implemented using the Optix [18] ray-tracing engine. The results of progressive rendering in AR can be seen in figure 2.

4 Previsualization Framework

We created a novel previsualization and relighting framework capable of delivering a high-quality rendering result by using the differential progressive path tracing algorithm. The presented framework can operate in two different modes:

- **Interactive preview-mode:** The 3D pose of the camera is tracked and the user can interact with the scene in real-time. The quality of output can be controlled by the number of samples per pixel. The preview mode suffers from noise caused by the high variance in Monte Carlo integration.
- **Progressive refinement mode:** This mode is used when a high-quality converged image is required. The interaction is paused and the last captured camera image is used for compositing. The synthetic image is progressively refined as more samples are used. In our experiments a fully converged image could be obtained within 2 minutes in average.

The following stages are performed per frame if the interactive preview mode is enabled: The environment lighting is captured by a camera with a fish eye lens and the positions of light sources are estimated if light source estimation is enabled. The observer camera image is taken and a 3D pose is calculated by the tracking system. Data from previous stages are sent to the ray-tracing engine running on the GPU. Mixed radiance rays are sent from the camera position towards the scene to evaluate both mixed and real radiances and to composite virtual objects with the real camera image. Direct illumination is evaluated at hit points of rays with the geometry. Indirect reflected light is estimated by shooting rays in random directions within the hemisphere above the hit point and using Monte Carlo numerical integration. A low number of ray samples (1-50) is used during interaction which leads to a noisy preview image. Temporal coherence between successive frames is achieved by using the same random parameters across each frame. A video can be recorded during the interactive session and the full quality movie can be generated in postprocessing.

When the user switches to the progressive refinement mode, the tracking of the camera, light source estimation and camera capturing is paused. The ray tracing engine uses the data from the last interaction frame and starts to accumulate the calculated light values. The progressive refinement is displayed to the user. The user can either wait until the image is fully converged to see the final result or can interrupt the progressive refinement to continue in interactive mode.

4.1 Estimation of Illumination

The estimation of real light is an important step to achieve visual coherence between lighting of virtual and real objects. Two different lighting methods are available in our framework. The first is Image Based Lighting [3]. There the whole environment map that is captured by a camera with a fish eye lens is used as a source of light. Inverse tonemapping is applied to the environment image in every frame to obtain HDR radiance values. This radiance is later accessed in the

ray miss program when a ray misses any geometry. This method can simulate natural lighting but requires more samples to calculate the converged solution.

The second option is light source reconstruction using image processing. Our framework utilizes the method proposed in [19] which uses connected component analysis to extract the positions of light sources. We run environment image capturing and light source estimation asynchronously in a separate thread.

4.2 Interaction

During the interactive preview mode a user is able to interact with virtual objects to see the rendering result in preview quality. The user can switch between interactive preview mode and progressive refinement mode by pressing a button. We use marker-based visual tracking to estimate the 3D pose of the camera although any other tracking system could be used. Our rendering framework supports dynamic geometry, materials, lighting and camera. Therefore interaction between virtual and real worlds can be achieved in real-time. Our current implementation does not use the movement of real objects, because predefined phantom objects are used. The system can be extended by automatic scene reconstruction.

4.3 Caustics

It is difficult to correctly simulate some light paths by path tracing. Caustics are especially problematic because the probability that the specular reflection will hit the light source is low in case of area light sources and it is zero in case of point light sources. Therefore we employ Photon Mapping [20] to simulate caustics. We use a GPU implementation of photon mapping using the OptiX ray-tracing engine [18] in order to achieve interactive frame rates while keeping quality of the created caustics high. We extended the interactive photon mapping implementation from [19] to work in path tracing. Caustic light is integrated in path tracing by using kernel density estimation at each ray hit point.

5 Results

Rendering results with our framework can be seen in figure 3. Each row shows a different scene, frame rates are depicted below the pictures. The first column shows interactive rendering with 1 ray sample shot per pixel. The second column contains interactive results with 9 rays per pixel and the third column shows the converged rendering solution in the progressive refinement mode. The scene in the first row contains three real paper cubes on a table and a virtual box with a metallic ball and a bunny (17K triangles). An image based lighting approach was used to lit the scene. The scene in the second row contains real cubes on the left and virtual spheres on the right side (4K triangles). Light source estimation by image processing was used here. Note the very fast convergence when simplified lighting conditions are used (10 s). The third row shows a real scene with inserted Buddha statues (581K triangles). The image based lighting approach

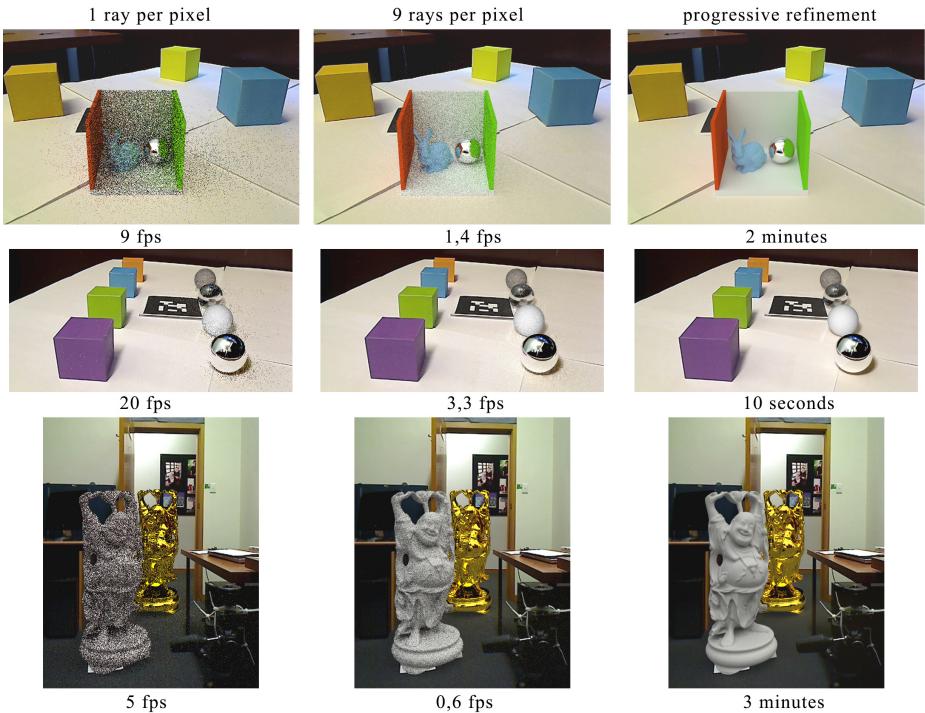


Fig. 3. Rendering results with differential progressive path tracing in our framework

was used. Interactivity can be still achieved despite the very high complexity of this scene. The results show the interaction of light between virtual and real objects. We analyzed the dependence of rendering speed on output resolution which is depicted in figure 4. We can see that our path tracing based rendering is interactive even for full HD resolution. It can be seen that our system is well scalable both in terms of triangle count and output resolution. All tests were performed on a laptop with quad-core CPU and a GeForce GTX 680M graphics card. A resolution of 800x600 was used in all evaluations except figure 4.

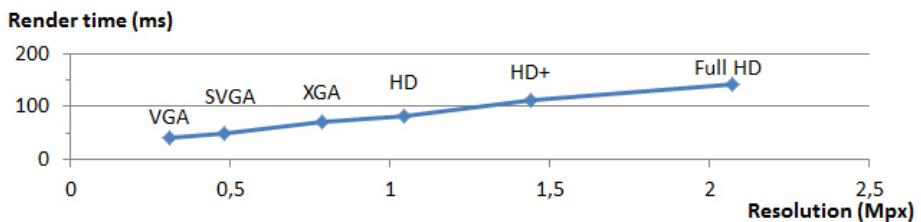


Fig. 4. Dependence of rendering time on resolution. A 3D model consisting of 20K triangles was used. 1 ray per pixel was shot to render the scene.

6 Conclusion and Future Work

In this paper we propose a novel progressive rendering algorithm for augmented reality based on path tracing and differential rendering. A framework capable of simulating complex global light transport between virtual and real worlds is presented. Our framework can possibly be used in movie previsualization, cinematic relighting or other fields.

In the future we plan to extend the framework with automatic real-time 3D scene reconstruction which will allow physical interaction between real and virtual worlds. Moreover we plan to improve our path tracing algorithm with advanced filtering approaches to reduce noise caused by high variance [21]. We believe that due to the improvements of graphics hardware, ray-tracing based rendering will become the standard rendering method for AR applications.

Acknowledgements. 3D models of Bunny, Dragon and Happy Buddha are courtesy of Stanford Computer Graphics Laboratory. Monkey model is courtesy of Blender. Peter Kán is financially supported by the Vienna PhD School of Informatics.

References

1. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986, pp. 143–150. ACM, New York (1986)
2. Fournier, A., Gunawan, A.S., Romanzin, C.: Common illumination between real and computer generated scenes. In: Proceedings of Graphics Interface 1993, Toronto, ON, Canada, pp. 254–262 (1993)
3. Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998, pp. 189–198. ACM, New York (1998)
4. Gorsch, T.: Differential photon mapping: Consistent augmentation of photographs with correction of all light paths. Eurographics Short Papers, Dublin (2005)
5. Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. In: Proceedings of the 2011 SIGGRAPH Asia Conference, SA 2011, pp. 1–12. ACM, New York (2011)
6. Knecht, M., Traxler, C., Mattausch, O., Purgathofer, W., Wimmer, M.: Differential instant radiosity for mixed reality. In: Proceedings of the 2010 IEEE International Symposium on Mixed and Augmented Reality, pp. 99–107 (2010)
7. Lensing, P., Broll, W.: Instant indirect illumination for dynamic mixed reality scenes. In: Proceedings of the 11th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2012, pp. 109–118. IEEE (2012)
8. Pessoa, S., Moura, G., Lima, J., Teichrieb, V., Kelner, J.: Photorealistic rendering for augmented reality: A global illumination and brdf solution. In: 2010 IEEE Virtual Reality Conference, VR, pp. 3–10 (2010)

9. Agusanto, K., Li, L., Chuangui, Z., Sing, N.W.: Photorealistic rendering for augmented reality using environment illumination. In: Proceedings of the International Symposium on Mixed and Augmented Reality, ISMAR 2003, pp. 208–218. IEEE Computer Society, Washington, DC (2003)
10. Gorsch, T., Eble, T., Mueller, S.: Consistent interactive augmentation of live camera images with correct near-field illumination. In: Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology, VRST 2007, pp. 125–132. ACM, New York (2007)
11. Ikeda, S., Taketomi, T., Okumura, B., Sato, T., Kanbara, M., Yokoya, N., Chihara, K.: Real-time outdoor pre-visualization method for videographers. In: 2008 IEEE International Conference on Multimedia and Expo, pp. 949–952 (2008)
12. Northam, L., Istead, J., Kaplan, C.S.: A collaborative real time previsualization tool for video games and film. In: ACM SIGGRAPH 2012 Posters. SIGGRAPH 2012. ACM, New York (2012)
13. Wong, H.H.: Previsualization: assisting filmmakers in realizing their vision. In: SIGGRAPH Asia 2012 Courses, pp. 1–20. ACM, New York (2012)
14. Pantaleoni, J., Fascione, L., Hill, M., Aila, T.: Pantaray: fast ray-traced occlusion caching of massive scenes. In: ACM SIGGRAPH 2010 papers. SIGGRAPH 2010, pp. 1–10. ACM, New York (2010)
15. Christensen, P.H., Harker, G., Shade, J., Schubert, B., Batali, D.: Multiresolution radiosity caching for global illumination in movies. In: ACM SIGGRAPH 2012 Talks. SIGGRAPH 2012. ACM, New York (2012)
16. Kán, P., Kaufmann, H.: Physically-based depth of field in augmented reality. In: EG 2012, pp. 89–92. Eurographics Association, Cagliari (2012)
17. Banterle, F., Ledda, P., Debattista, K., Chalmers, A.: Inverse tone mapping. In: Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia, GRAPHITE 2006, pp. 349–356. ACM, New York (2006)
18. Parker, S.G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., Stich, M.: Optix: a general purpose ray tracing engine. ACM Trans. Graph. 29 (2010)
19. Kán, P., Kaufmann, H.: High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In: Proceedings of International Symposium on Mixed and Augmented Reality, pp. 99–108. ACM Press (2012)
20. Jensen, H.: Realistic Image Synthesis Using Photon Mapping. Ak Peters Series. A K Peters, Limited (2009)
21. Gastal, E.S.L., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. ACM TOG 31, 1–13 (2012); Proceedings of SIGGRAPH 2012

Projection on Suitable Sub-surface Selected in Indoor Environment

Shafaq Mussadiq and Rehan Hafiz

National University of Sciences and Technology, Islamabad, Pakistan
`{shafaq.mussadiq,rehan.hafiz}@seecs.edu.pk`

Abstract. Traditional projection environments were characterized by static projectors projecting at specialized surfaces. The recent advent of pico projector-camera pair embedded into mobile devices provides flexibility to project in environments previously considered as unsuitable. One of the major challenges for such systems is to automatically select the best portion from a cluttered environment suitable for projection. In this paper we propose a scheme to select the most appropriate projection surface and then make a confined projection in the selected area. The decision for suitability is taken based upon the color homogeneity and the size of the candidate surface; while for finding the correspondence between the surface and projection contents an existing perfect map based spatial encoded pattern is suggested. By performing spatial decoding in only regions suitable for projection; the search space for spatial decoding is greatly reduced. Experiments show encouraging results.

Keywords: pico projectors, projection surface, spatially encoded pattern, feature correspondence.

1 Introduction

The increase in the number and capabilities of portable smart devices such as smart phones, smart cameras, personal digital assistants and anticipated smart watches has resulted in massive visual data that the user wish to view and share among each other. However, the small screen size of such devices puts a limitation on their usage. The recent advent of pico-projectors however enables these portable devices to provide larger visual displays for increased interactivity and productivity. Unlike traditional projection scenarios where projection surfaces were purpose built the users of pico-projectors have to find a suitable surface in their surroundings for the projection. A recent study [14] investigated the social response of users to explore the type of projection surface they prefer, and the contents they wish to share using Experience Sampling Method (ESM) and claimed that typically users selected large planar surfaces such as walls, tables, floor and ceiling for projection. However, in cluttered environments users faced difficulties: for example in making a confined projection on a train seat, undistorted projection on chair back, and some had comments related to the color of projection surface. Thus number of factors may affect the quality of projection

like surface color, surface texture, surface size and geometry of the surface that should be considered in order to provide the user with an appropriate projection. The use of pico-projectors with mobile robots for the purpose of assisted augmented display has been explored in [8] [6] where the authors used a mobile robot coupled with a pico-projector to display additional information related to paintings in a museum with prior information about the exact map of the museum and the location of projection surface. This provides further motivation for the case where a mobile robot may encounter a totally new and unfamiliar place where it has to select a suitable projection surface and then make a confined projection in the available surface as well. Unlike most other previous work, where the emphasis is on undistorted projection of imaging contents on non-planar and irregular surfaces, we propose a scheme to find the best projection surface in cluttered environment for the case where the projector device has been casually placed in the environment. The scheme assumes the presence of a camera embedded into the same device such that both projector and camera are having similar extrinsic parameters differing only by translation due to the physical position of projector and camera. The only other assumption being the larger field of view (FOV) of camera as compared to projector assuring the camera view to encompass the maximum projection area. The remaining paper is organized as follows. In section 2, the work related to the camera projector displays in literature is described briefly. The proposed scheme for projection on suitable surface selected in indoor environment is illustrated in section 3. The section 4 describes the experimental setup and results, followed by the conclusion and future work in section 5.

2 Related Work

Mobile devices featuring pico-projectors allow the user to casually project contents on a variety of surfaces, such as workplace areas or office walls that may provide a cluttered environment. We propose to break the problem in two phases: Selection of an appropriate portion of the environment, and confinement of the projection in the selected region. Confinement in the selected region of interest shall however require the correspondence to be known between the projected content and the projection surface. This correspondence shall be than used to estimate the required geometric correction to be applied to the content. A number of schemes have been proposed in the past to perform geometric correction and are typically classified as active and passive schemes [16]. Active systems employ the use of structured light patterns for estimating the projection surface while passive schemes make use of imperceptible patterns. The structured light techniques have been broadly classified into three; direct codification, time-multiplexing and spatial encoding based [11]. Direct codification makes use of distinct colors/gray levels so that each pixel is uniquely identified. This is not suitable for our case where pico-projectors are characterized by very low brightness values (10-20 Lumens) and the colors are not as vibrant as normal projectors. Also, the assisting cameras are mobile-grade with

low resolution. The decoding of the perceived color is further effected by the color of the projection surface which may not be necessarily white in color. Figure 1 shows a color coded pattern [2] projected for different lighting and background conditions (Figure 1(f)). Figure 1 (b, c, d and e) show the actual decoded colors in the pattern of Figure 1(a). Figure 1(g, h, i and j) show the corresponding decoded colors for the different cases of Fig 1(a). Variations in the decoded colors from that of (b, c, d and e) provides clear evidence of the challenges involved in the usage of color coded patterns for arbitrary projection surfaces.

Another possible scheme is to make use of Time-multiplexing to project patterns over time so that they are conveniently decoded. As for example the 3D reconstruction is done by using the time multiplexing combined with spatial encoding scheme [5]. The proposed method successfully reconstructed the objects but they have defined a particular background screen in their proposed architecture which is not possible in our case. Similarly [9] also used the spatio-temporal encoding scheme for 3D reconstruction but they have used the colored pattern which have shown to be not suitable for our case (Figure 1). Spatial encoding methods [11] make use of a single coded pattern. A number of spatially encoded patterns has been developed. The simplest are the checkerboard patterns [10] [18] and point-grid [13] based patterns that have been employed to find the projector-surface correspondences. One of the limitations of these patterns is that the complete pattern is to be made visible otherwise it would be difficult to tell the exact patch location in the entire image. Most of the geometric corrections schemes typically assume a pre-defined model such as quadric or swept surfaces [10] [13] and in general are not suitable for cluttered environments; also they have a single surface for projection as opposed to our scheme where we have to search for the surface that best fits on our defined criteria. Other spatial encoding methods are based on mathematical formulations that overcome the limitation of previous ones by assigning a unique code to the image pixels based on their neighborhood. Thus, even if the portion of pattern is visible, its exact location can be identified in the original image. Some examples of these patterns

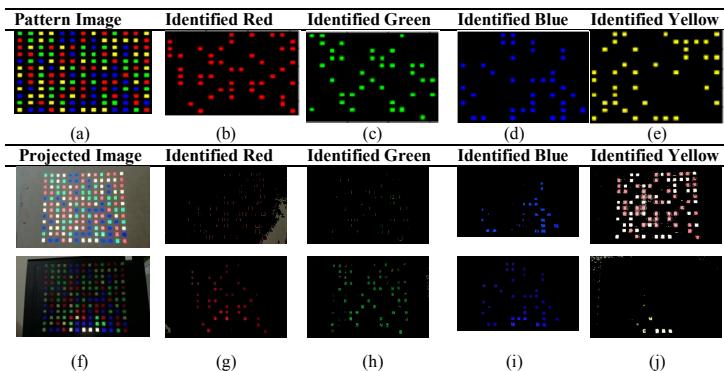


Fig. 1. Projection of Color Coded Pattern

are the single axis de-bruijn or pseudo random sequences [15] [7] and the two dimensional M arrays or perfect maps [1] [17]. We shall be employing a perfect map based scheme for our methodology.

3 Proposed Scheme

We have developed a system that automatically selects the best surface and then makes a confined, undistorted projection over the selected surface. The two steps are illustrated in detail in the following sections, with the experimentation results.

3.1 Suitable Surface Selection

We propose to tackle the problem of selection of suitable projection surface in a cluttered environment in two-steps. Firstly, identification of all possible projection surfaces in the camera view and secondly identifying the largest projection surface that falls in the projection area. The second step is achieved by finding the correspondence between the surfaces available and the projected contents.

Since indoor cluttered environments are typically characterized by objects of different shapes, sizes, textures and reflection properties and may be placed at varying depths from the projector-camera pair. We define a surface suitable for projection as having at least two desirable characteristics: Color homogeneity and considerable size for useful projection. A surface with uniform color consistency and texture will be a preferred candidate for projection; since it would give equal color gain to all points on the projection surface. For achieving the desired characteristics we need to perform the color based image segmentation. Experimentation has been performed by using k-means, particle swarm optimization (PSO) based [4] and mean-shift image segmentation [3]. The results of using all three methods of segmentation are shown in Figure 2 below. Although k-means and mean-shift are based on clustering technique, one of the limitations of k-means is that it requires a prior assessment about the number of clusters and that the confinement of the cluster's shape. On the other hand, PSO based techniques require fine tuning of the parameters to get appropriate results. Also unlike k-means, mean-shift works well when there are too many clusters and the clusters may vary in size, which may happen usually in indoor environments and provides results very true to the predicted segments as are shown in Figure 2.

In particular we propose to use the mean-shift based color segmentation to segment out regions in the environment. The motivation for selection of mean-shift based segmentation is also provided by [12]; where mean shift based segmentation was employed as a preprocessing step for planar surface estimation for robot navigation.

The efficiency of mean-shift is also tested under varying illumination conditions and compared with k-means and PSO. Although the results of PSO are close to the mean-shift in the Figure 2(a), but when both the methods are compared under varying illumination conditions, mean-shift shows encouraging

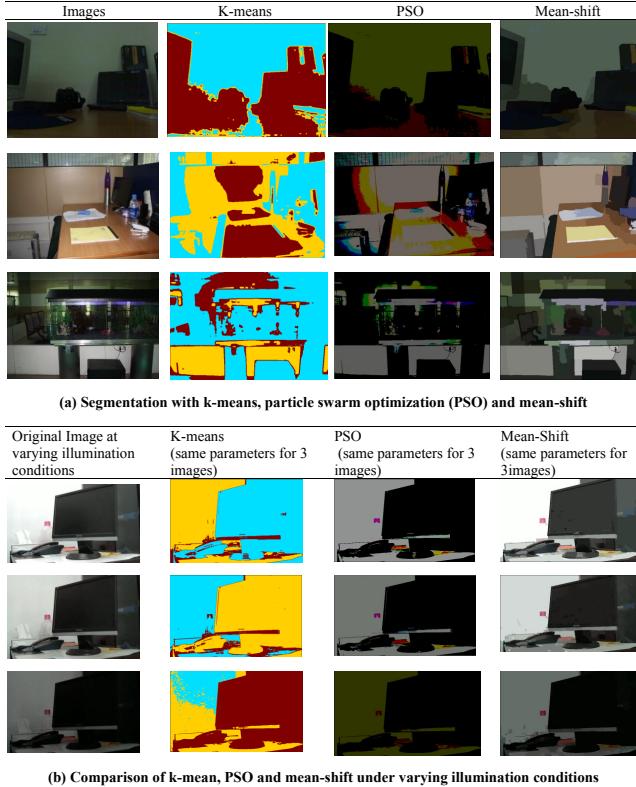


Fig. 2. Segmentation with K-means,PSO and Mean-shift

results (with reference to Figure 2(b)). As for example, in the Figure 2(b), the mean-shift has segmented out the monitor as a separate region with respect to its background cupboard in all 3 cases, whereas the PSO has merged them altogether in a single segment that can deviate our decision of largest surface selection to a false detected largest region.

3.2 Feature Correspondence

Once, the most suitable projection surface is identified, the next step is to find the correspondence between the environment and the contents being projected. Projection of structured light pattern has been used in the literature as described earlier. We are not considering the time multiplexing encoding methods, because they require a number of pattern images to be projected and a code word for the pixels is determined by the combination of the projected corresponding pixels along the time. Likewise the direct codification may be affected by the low intensity and low color quality of pico-projector's output. For cluttered environment where only a portion of projected pattern may be visible; we thus suggest to use

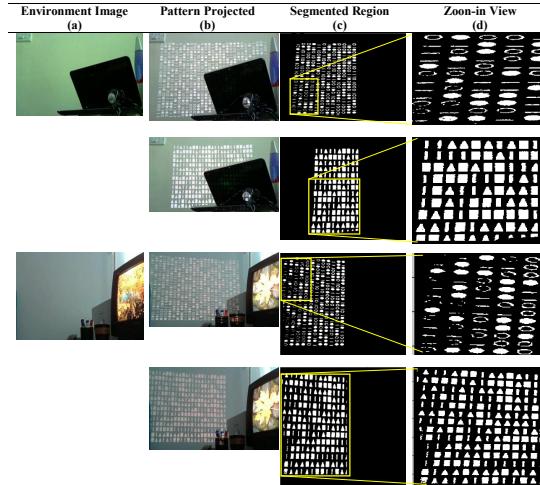


Fig. 3. Comparative Illustration of Patterns Presented in[1] and [17]

spatial encoding scheme. Specifically, two previously presented spatial encoded patterns [1][17] were tested for the cluttered environments using pico-projectors.

We compared the efficiency of both these schemes for a variety of indoor cluttered environments (Figure 3). The encoded pattern image (P_{proj}) containing mxn objects is defined by mxn encoding matrix (M_{proj}). Each element of M_{proj} corresponds with an object location in P_{proj} , and is assigned a nine digit code uniquely defined by its eight surrounding neighbors [1]. The encoded pattern image (P_{proj}) is projected and captured using the camera, which is then decoded to find out the correspondence between the projected and the captured image. Let the captured pattern image (by the camera) be I_{proj} . The captured image is masked with the previously segmented maximum suitable rectangular projection region ($rect_{max}$) (Figure 3(c)). This helps in two ways. Firstly, the pattern objects being projected on regions with homogenous color experience less noise and secondly, the decoding only needs to be done in the region specified by $rect_{max}$, thus reducing the decoding search space. For decoding, the masked captured pattern image is segmented out to find the objects within it. Encoded pattern by [17] is characterized by solid objects while the pattern by [1] is characterized by thin boundaries encompassing holes. It was observed that objects with holes were more challenging to be detected correctly since their thin boundaries get disconnected due to relatively low quality and less resolution of low cost pico-projector-camera pair present on most mobile devices. In [17], the authors have decoded the primitive shapes based on the number of angles present in the shape and further considering their length to width ratio. For the case of pico-projectors however, detecting the correct number of angles can become a challenging task due to their low quality output as clear from the segmented objects in the Zoom in View of Figure 3(d). We thus employed

a different decoding scheme as compared with [17]. An object is identified as a triangle if the ratio of the area of the shape (solid filled pixels) to the area of its bounding box (i.e. the ratio of white to black pixels) is less than constant ' T_1 ', otherwise the primitive/symbol is considered as a stripe or rectangle, but in case of stripe; its length-to-width ratio, ' LW_{ratio} ' is less than constant ' T_2 ', and otherwise for the rectangle. The constants T_1 and T_2 are estimated, comparing with the ideal primitive shapes (i.e. triangle, stripe and rectangle), with added tolerance factor and are set to 0.7 and 0.5 respectively. These constant values are tested for number of scenarios for its verification. The scheme is summarized below in 1-4.

$$Decode(symbol) = \text{Triangle if ratio} < T_1 \quad (1)$$

$$Decode(symbol) = \text{Stripe} \begin{cases} \text{if ratio} > T_1 \\ \text{if } LW_{ratio} < T_2 \end{cases} \quad (2)$$

$$Decode(symbol) = \text{Rectangle} \begin{cases} \text{if ratio} > T_1 \\ \text{if } LW_{ratio} > T_2 \end{cases} \quad (3)$$

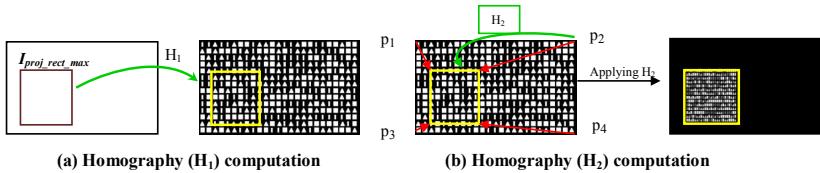
$$Decode(symbol) = \text{unidentified otherwise} \quad (4)$$

The following algorithm summarizes the proposed projection surface selection procedure.

1. Capture two images: I without projection and I_{proj} after projecting P_{proj} .
2. Do Mean Shift Segmentation of I into n number of regions:
 $(r_1, r_2, r_3, \dots, r_{n-1}, r_n)$.
3. Finding the r_i where $i = 1, 2, 3, \dots, n$, with $\max(\text{area}) = r_{max} \text{if } (\sum_b b_k > (\sum_l b_l)) \text{ for each } r_1, r_2, r_3, \dots, r_{n-1}, r_n$
4. Determining the largest rectangular area $rect_{max}$ of the region r_{max} .
5. Perform intersection (masking) of $rect_{max}$ and I_{proj} and call it $I_{proj_rect_max}$
6. Decode and find correspondence between $I_{proj_rect_max}$ and P_{proj} .

3.3 Projection on Selected Surface

Once we have selected the suitable surface, we now have to make a confined projection over the sub surface selected. The points in one plane are related to the corresponding points in the other plane through the homography matrix H . We first compute the homography (H_1) required to find the location in pattern image (P_{proj}) that corresponds to the largest rectangular region ($I_{proj_rect_max}$) by using the decoded patterns. To scale the projected contents to the largest rectangular region another homography (H_2) is computed that limits these projection to the confined area of the largest rectangular region (Figure 4).

**Fig. 4.** Homography H_1 and H_2

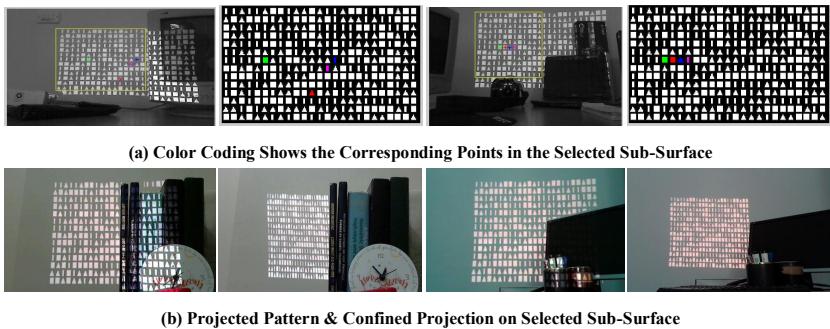
4 Experimental Results

We have used a microvisionpicop pocket projector (848x480) with a Logitech camera c310. The camera is placed on the projector to treat them as a single unit. Number of experiments were performed to verify the proposed scheme. In order to assess the technique of primitive detection, based on number of angles, presented by authors in [17], we performed a comparative analysis of angle based detection scheme with our proposed scheme.

Table 1 shows a comparison of correct decoding percentage populated by taking the average for 10 different indoor scenarios. It is shown that for the pico-projector case our proposed deciding scheme provides over 90% correct decoding of primitive shapes as compared to just over 30% correct decoding obtained by approach proposed by [17]. This claim is further strengthened by the fact that

Table 1. Percentage correct decoding of primitive shapes using pico projector (848x480), for angle based technique[17] and our proposed technique

Shapes	Angle Based Detection	Proposed Algorithm
Rectangle	31%	92%
Triangle	33%	94%
Stripe	32%	97%

**Fig. 5.** Suitable Surface Selection in Cluttered Environment

the authors in [17] have adopted a primitive correction procedure by observing N frames under their defined constraints. This however not applicable for our case where we make use of a single captured pattern to find the correspondence for estimation of Homography.

Figure 5(a) shows the selected suitable region (marked by rectangle) for the cases from cluttered office environments, using the techniques described in section 3.1 and 3.2. Figure 5(b) shows the confined projection in the largest subsurface selected, applying the procedure described in section 3.3.

5 Conclusion and Future Work

The work has been inspired by the pico projectors that allows the user to project the imaging contents anywhere; on any available surface. In our work we have presented a scheme for automatic selection of suitable surface for projection. The criterion used for selection is the largest possible surface present in the environment based on the color homogeneity of the surface. Once the surface is selected, next task is to find the corresponding feature points between the projected and original image in order to estimate the homography that relates the points in one image plane to the corresponding points in the other image. A coded structured light pattern has been used for correspondence estimation and its decoding scheme has been specifically updated to produce reliable results for pico-projector scenario. Experiments have shown encouraging results verifying the proposed scheme. The current scheme is specifically designed to have a confined projection on the planar surfaces selected in the environment. The work could be extended to deal with the non-planar and curved surfaces that require the estimation of the surface geometry and then prewarp the projected image accordingly.

References

1. Albitar, C., Doignon, C., Graebling, P.: Calibration of vision systems based on pseudo-random patterns. In: IEEE/RSJ International Conference on Intelligent Robots and Systems 2009, pp. 321–326 (October 2009)
2. Chen, H., Ma, S.: Feature points matching for face reconstruction based on the window unique property of pseudo-random coded image. Special Issue on Intelligent Mechatronics 22, 688–695 (2012)
3. Comaniciu, D., Meer, P.: Mean shift: A robust approach towards feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5) (May 2002)
4. Ghamisi, P., Couceiro, M.S., Benediktsson, J.A., Ferreira, N.M.F.: An efficient method for segmentation of images based on fractional calculus and natural selection. Expert Systems with Applications 39, 12407–12417 (2012)
5. Ishii, I., Yamamoto, K., Doi, K., Tsuji, T.: High-speed 3d image acquisition using coded structured light projection. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 925–930 (2007)

6. Kwon, E., Kim, G.J.: Humanoid robot vs. projector robot: Exploring an indirect approach to human robot interaction. In: Proceeding of 5th International Conference on Human-Robot Interaction, pp. 157–158 (2010)
7. Pages, J., Salvi, J., Collewet, C., Forest, J.: Optimised de bruijn patterns for one-shot shape acquisition. *Image and Vision Computing* 23, 707–720 (2005)
8. Park, J., Kim, G.J.: Interacting with robots "indirectly" through projected display. In: Proceeedings of Ubiquitous Robots and Ambient Intelligence (2008)
9. Payeur, P., Desjardins, D.: Structured light stereoscopic imaging with dynamic pseudo-random patterns. In: Kamel, M., Campilho, A. (eds.) ICIAR 2009. LNCS, vol. 5627, pp. 687–696. Springer, Heidelberg (2009)
10. Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., Forlines, C.: iLamps, geometrically aware and self configuring projectors. In: ACM SIGGRAPH (2003)
11. Salvi, J., Fernandez, S., Pribanic, T., Llado, X.: A state of the art in structured light patterns for surface profilometry. *Pattern Recognition* 43, 2666–2680 (2010)
12. Tang, H., Zhu, Z., Xiao, J.: Stereovision-based 3d planar surface estimation for wall-climbing robots. In: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (October 2009)
13. van Baar, J., Raskar, R.: Flexible calibration of multiple projectors for low-cost curved screen displays. In: International Conference on Artificial Reality and Telexistence (ICAT) (November 2004)
14. Wilson, M.L., Robinson, S., Craggs, D., Brimble, K., Jones, M.: Picoing into the future of mobile projector phones. In: CHI 2010 Extended Abstracts on Human Factors in Computing Syatems, pp. 3997–4002. ACM, New York (2010)
15. Xu, J., Xi, N., Zhang, C., Shi, Q., Gregory, J.: Real-time 3d shape inspection system of automotive parts based on structured light pattern. *Optics and Laser Technology* 43, 1–8 (2011)
16. Yang, R., Welch, G.: Automatic projector display surface estimation using everyday imagery. In: 9th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (2001)
17. Yang, T.-J., Tsai, Y.-M., Chen, L.-G.: Smart display: A mobile self-adaptive projector-camera system. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (July 2011)
18. Zhou, J., Wang, L., Akbarzadeh, A., Yang, R.: Multi-projector display with continuous self-calibration. In: Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems (2008)

A Framework for the Visualization of Finite-Time Continuum Mechanics Effects in Time-Varying Flow

Alexy Agranovsky, Harald Obermaier, and Kenneth I. Joy

University of California, Davis

Abstract. Integration-based flow visualization provides important visual cues about fluid transport. Analyzing the behavior of infinitesimal volumes as opposed to the behavior of rigid particles provides additional details valuable to flow visualization research. Our work concentrates on examining the local velocity gradient tensor along the path of a particle seeded within time-varying flow to produce a visualization highlighting temporal characteristics of particle behaviors, such as deformation. We present a framework for the analysis and visualization of such characteristics, focused on providing concise representations of physically meaningful flow features such as separation regions and vorticity. We apply the derived techniques to two data sets, highlighting the importance of such higher order Lagrangian analysis techniques to time-varying flow analysis.

1 Introduction

Modern flow simulations often output time-varying vector fields from which a wealth of information can be ascertained. Key techniques in visual flow analysis place particles in the flow field and follow their path as they are advected over time, thus visualizing the translational behavior of such particles.

Our work focuses on the intrinsic volumetric effects of a region of space surrounding a particle as it is advected through the field. More specifically, we are interested in secondary effects of Continuum Mechanics such as the stretching, shearing, and rotation characteristics of the flow experienced within the region. We develop a framework to isolate and visualize these individual fluid metrics, providing further insight into particle interactions within a local neighborhood. We demonstrate how a visualization of these metrics is a crucial tool to discover and understand behaviors such as mixing.

Local instantaneous transformations of flow particles are represented by the velocity gradient tensor. To examine the transformation of a volumetric particle over a (finite) amount of time, this information has to be accumulated as it is advected through the flow field. We present a unified framework for the accumulation and visualization of a number of Continuum Mechanics effects, such as shear, stretching, and rotation. After a finite time interval, our results display the varying flow field effects that have acted on the particle. In summary, the contributions of this work to the area of flow visualization are as follows: 1) We provide a theoretical framework for a number of advanced continuum mechanics descriptors. 2) We show how existing methods fit into this framework and develop novel descriptors of particle transformations. 3) We encode and combine descriptors as scalars for visualizations that enhance the deformation traits of particles.

2 Related Work

Considerable effort has been put into identifying flow features into one graphical representation [1–3], commonly referred to as glyphs, by visualizing geometric primitives [4]. While our work also identifies local flow behavior, we do not aim to show all of the metrics at once. A different visualization of flow characteristics is given by work that studies higher order effects of flow fields in the form of mixing [5] and volume deformations [6, 7] induced by the velocity gradient tensor. Our work in part makes use of the flow deformation definitions given in these papers.

The use of the *Finite-Time Lyapunov Exponent* (FTLE), a measure of exponential stretching, was introduced by Haller [8] to extract salient flow features. Soon after, techniques [9–11] provided robust FTLE calculations along with visualizations focusing solely on flow divergence. For these works, the FTLE is calculated strictly using a flow map approach, while our work is interested in the characteristics of flow deformations experienced along the particles pathline [12]. Kasten et al. [13] introduced the notion of localized FTLE, exchanging the standard flow map gradient tensor with an accumulation of the velocity gradient tensor along pathlines.

Work has also been performed to identify the particular rotation and strain experienced during deformation. The rotation given by the velocity gradient tensor has been associated with vortex core extraction [14], while the strain component can identify shearing [15]. Both tensor components have been combined to identify flow features [16], with local strain used with FTLE for flow structure identification [17]. The visualization of local neighborhood deformation along a particle’s trajectory has been captured using polygons [18], streamtubes [19], and predicates used to categorize trajectories [20, 21] according to chosen flow properties. While our work also concentrates on extracting local flow field changes, the visualization accumulates these deformations into one overall image, providing a framework for identifying specific causes for con-tortion through arbitrary vector fields. Fuchs et al. [22] accumulate vorticity, a form of rotation, as an additional measure for vortex extraction and Obermaier et al. [23] accu-mulate strain, a variation of shear, in mantle flow fields for automatic strain analysis for 3D geophysical data.

Our work explores all facets of the deformation, for example magnitude, strain, and rotation within one consistent algorithmic framework. By combining the deformation experienced by a particle’s neighborhood into one visualization, we can ascertain the primary fluid features along the particle’s path. Furthermore, the provided results show that the examination of particle transformations in turbulent flow allows for the isolation and relation of multiple flow characteristics, thereby highlighting notable features.

3 Framework Concept

The goal of our work is to present a unified framework for the transformations experienced by particle neighborhoods moving along trajectories through the flow field. Over the lifetime of such a neighborhood, the overall deformations may be complex, but can be broken down into a set of metrics with distinct physical meanings that help explain the transformation. More specifically, we give examples of how to examine the accumulation of rotation, shear, fractional anisotropy, and stretching to better understand fluid

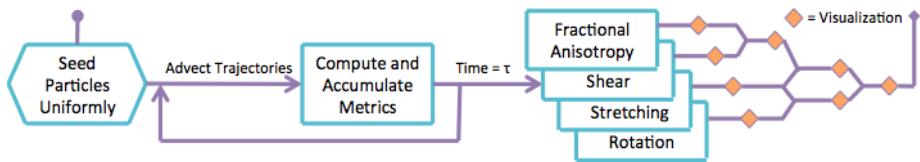


Fig. 1. Workflow of the proposed analysis and visualization framework. Particles are seeded uniformly and advected until time τ , during which the metrics are calculated and accumulated along the trajectory. These metrics can then be visualized independently or in combination.

mixing behavior. Particles are seeded uniformly throughout the data set and trajectories are advected until a time τ , during which the desired metrics are calculated and accumulated. They are then visualized at time τ both independently and in combination, as we will explore after an in-depth look at the metrics. In the next section, we describe the mathematical process involved with the calculation and accumulation of these metrics.

4 Velocity Gradient Tensors along Trajectories

Our work explores the velocity field effects on a particle advected through the flow, specifically targeting transformations due to stretching, rotation, and shear on the particle's neighborhood (Figure 2(a)). The majority of work on post-simulation analysis focuses on fluid transport, with a much smaller subset examining in isolation, individual fluid mixing characteristics (Figure 2(b)). Our work stresses the importance of both transport and various transformation metrics, exploring how in combination, these effects give a greater insight into flow behavior. In the following, we establish the context in which we analyze the flow field and provide the required mathematical background. To achieve notational differentiation, vectors are denoted as bold face lower-case letters \mathbf{v} to distinguish them from scalars $s \in \mathbb{R}$.

4.1 Velocity Gradient Tensor

Our main focus is on 3D velocity analysis, i.e., $\mathbf{v} = (\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t}, \frac{\delta z}{\delta t})$. For our purposes, this velocity is obtained directly from a time-dependent, 3D vector field over a finite space domain $U \in \mathbb{R}^3$ and a finite temporal domain $I \in \mathbb{R}$.

To begin analysis over the entire data set, particles are seeded in a uniform distribution over all axes. Using a Runge-Kutta 4th/5th order integration scheme, the particles

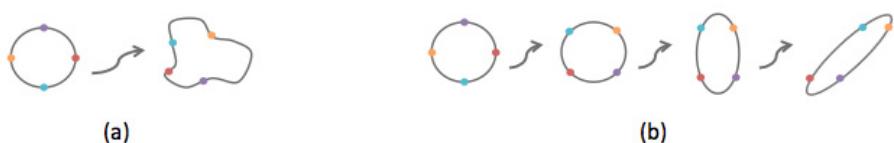


Fig. 2. Neighborhood transformations. Figure (a) shows the combined effects that rotation, stretching, and shearing have on a neighborhood while (b) exemplifies them individually.

are advected for a time t through the vector field v . To capture instantaneous change in a neighborhood around a particle position $\mathbf{x} = (x, y, z)$, the spatial gradient of the flow field is obtained at every time step using central differencing.

This gradient captures local changes within the flow field. As a 3×3 second-order tensor, it is a linear mapping between vectors \mathbf{v} and $\mathbf{w} \in \mathbb{R}^3$. In our case, this so called *velocity gradient tensor* $\nabla \mathbf{v}$ of a flow field $v : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ describes a local linearized rate of change in velocity:

$$\nabla \mathbf{v}_{ij} = \frac{\partial v_i}{\partial x_j}$$

with components corresponding to the central differencing calculation. To quantify the spatial changes given by the velocity gradient tensor, we take a closer look at its *eigenvectors* and *eigenvalues*, which give the instantaneous maximal and minimal directions and magnitude, providing principal components of the deformation.

We aim to analyze flow fields based on the transport of particles. Examining the velocity gradient tensor, instantaneous changes in the flow direction can be extracted at any point location. By calculating $\nabla \mathbf{v}$ frequently at positions along a particle's trajectory, the deformation along a specific path through the fluid is made apparent. This instantaneous change in velocity allows for the local extraction of the desired deformation descriptors. Accumulating these changes will summarize the deformations experienced by a particle throughout its lifetime, exposing flow features. In contrast to purely stationary analysis of velocity gradient components, this approach allows for the investigation of truly time-varying behaviors in a Lagrangian setting.

4.2 Neighborhood Gradient Tensor

To map the instantaneous change in velocity back to a deformation of the particle neighborhood after a single time step, consider a mapping $\mathbf{g}(\mathbf{x}) = (g_x(\mathbf{x}), g_y(\mathbf{x}), g_z(\mathbf{x})) \in \mathbb{R}^3$ that holds the position value of arbitrary points \mathbf{x} in a particle neighborhood after the particle was advected for Δt . The corresponding *deformation gradient tensor*

$$\mathbf{D}_{ij} = \frac{\partial g_i}{\partial x_j}$$

is a second-order tensor describing the local deformation of line elements in space. The direct relation between velocity fields and the displacement function allows for the computation of the deformation gradient tensor \mathbf{D} using the relation

$$\mathbf{D} = e^{\nabla \mathbf{v} \cdot \Delta t} \tag{1}$$

to which a first order approximation of the matrix exponential is given by $\mathbf{I} + \nabla \mathbf{v} \cdot \Delta t$. Consider a pathline $p(t) = p(\mathbf{x}, t_0; t)$ for a particle seeded at location \mathbf{x} at time t_0 and advected till time t . The time evolution of the deviation of the flow field f between t_0 and t can be discretized in intervals of size Δt . Accumulating these changes, a matrix forms which holds a mapping of the neighborhood deformation at starting point $p(t_0)$ to the end point $p(t)$. \mathbf{D} then describes the state of the neighborhood as a volume, allowing for volumetric analysis to quantify desired descriptors of the deformation. In this paper, we are interested in the deformation that occurs up to time t , and therefore we accumulate the deformation gradient tensor to said time.

4.3 Accumulating the Deformation Gradient Tensor

To quantify the deformation at any given time step, the effects of previous deformation gradient tensors are accumulated up to the desired time. Given an arbitrary vector \mathbf{v}_i at time t_i , the effects of the deformation on \mathbf{v}_i are quantified as $\mathbf{v}_{i+1} = \mathbf{D}_i \cdot \mathbf{v}_i$ where \mathbf{v}_{i+1} is the vector \mathbf{v}_i after experiencing the deformation governed by \mathbf{D}_i . To accumulate these transformations over the lifetime of a particle, the deformation gradient tensor must be applied to the original vector \mathbf{v}_0 at all time steps:

$$\mathbf{v}_n = \prod_{i=0}^{n-1} \mathbf{D}_i \cdot \mathbf{v}_0. \quad (2)$$

The deformed vector \mathbf{v}_n is therefore defined by the initial vector and an accumulation of all transformations prior to step n . For an initially spherical neighborhood (input vectors \mathbf{v}_0 lie on the unit sphere), the accumulated deformation tensor is a direct representation of the deformed neighborhood at step n .

While this form of accumulation adds up (geometric) deformation information, other descriptors, such as scalar valued rotation must be accumulated in a slightly different form. Since an accumulated deformation will not allow for the extraction of descriptors like total rotation of a particle (due to positive and negative rotations or total rotations exceeding 2π), we extract such descriptors locally and use standard statistical measures for accumulation. Total rotation, for example, can be computed by summing up either absolute or signed local rotation angles. Similarly, average and maximal values of descriptors may be extracted along trajectories by examining local properties.

5 Scalar Descriptors of Volume Behavior

For a breakdown of the analysis process involved with a volume deformation, we first take a look at the instantaneous neighborhood itself, eliminating any temporal components. The neighborhood of a flow particle is represented abstractly in the form of a sphere, manipulated into an ellipse after experiencing a change in velocity. Let \mathbf{E} be the matrix that deforms the sphere into the ellipse. The axes of the ellipse are found by extracting the singular values of \mathbf{E} , revealing the directions of the deformation.

By associating a velocity with the neighborhood (tangential to the particle's trajectory), we can examine in detail the forces acting on the ellipse and further distinguish orientations. The exact combination of rotation and strain associated with the deformation is found by decomposing the velocity gradient tensor. Furthermore, by combining

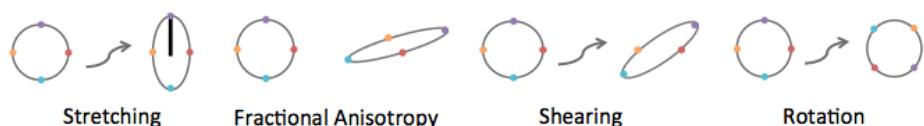


Fig. 3. Transformation effects on a spherical volume. This Figure shows the effects of stretching, shearing, and rotation on a spherical volume and also visually displays fractional anisotropy, where the left figure has a fractional anisotropy near 0 and the right near 1.

a sequence of these deformations, the overall change in the neighborhood can be analyzed with respect to direction and magnitude over a finite time, shown in Figure 2(b). By accumulating the deformation gradient tensor over the life-time of a particle and applying statistical measures, the total separation and rotation experienced by the local neighborhood can be quantified.

5.1 Stretching

We first begin by focusing on the magnitude of the maximal volume stretching. This notion is equivalent to how FTLE characterizes laminar and turbulent flow by measuring the separation and convergence rate of infinitesimally close particle trajectories. Given a finite-time interval, FTLE gives a measure of exponential stretching magnitude for areas of flow experiencing divergence or convergence (forward or backward time integration).

The most essential component to calculating the FTLE is the flow map gradient. In their work [13], Kasten et al. present a flow map free method for calculating the local particle deformation, replacing the flow map gradient with an accumulation of local separation measures. Without relying on a flow map, Kasten et al. presented the idea of a localized Finite-Time Lyapunov Exponent (L-FTLE) which relies solely on the local separation along an individual pathline.

Hence, over the life-time of a particle, the accumulated deformation gradient tensor can be directly applied to the calculation of the FTLE. If the tensor T_n represents the deformation accumulation after n time steps, applying the spectral norm $\|\cdot\|_\lambda$ to T_n gives the maximum stretching of the local neighborhood. This is equivalent to performing a singular value decomposition. Therefore, the normalized localized FTLE is defined by

$$L - \text{FTLE}(\mathbf{x}, t) = \frac{1}{t} \ln(\|T_n\|_\lambda) \quad (3)$$

giving a magnitude measure for the volume deformation along the pathline. Thus, the magnitude of maximal stretching is immediately encoded in the accumulated deformation gradient tensor.

5.2 Fractional Anisotropy

We are not only interested in the magnitude of stretching, but also the contortion itself. This section outlines *fractional anisotropy* (FA), which approximates how line-like a neighborhood becomes after a distortion. The FA is a scalar value between zero and one that describes the degree of anisotropy during the deformation process. A value of zero means that the deformation is fully isotropic or that it is unrestricted in all directions, see the left hand side of Figure 3 on fractional anisotropy. A value of zero may also occur if the deformation is equal in all directions. A value near one means that the deformation occurs along one axis only, see the right image of Figure 3. The equation for calculating the fractional anisotropy is

$$FA(\mathbf{T}) = \sqrt{\frac{3}{2}} \frac{\sqrt{(\lambda_1 - \text{tr}(\mathbf{T}))^2 + (\lambda_2 - \text{tr}(\mathbf{T}))^2 + (\lambda_3 - \text{tr}(\mathbf{T}))^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \quad (4)$$

where $tr(\mathbf{T})$ is the normalized trace of the accumulated deformation gradient tensor and the λ 's are the eigenvalues of the same tensor.

With the deformation pictured as an ellipse, the fractional anisotropy gives an idea as to how line-like the neighborhood is. A fractional anisotropy of zero would be a perfect sphere, while an FA of one would cause the ellipse to degenerate into a line. Thus, the scaling allows for two distinct filtering techniques, using the FA value to either visualize a range of deformations or to weight the importance of other flow features.

5.3 Strain and Shear

To further examine the velocity gradient tensor, we do a standard decomposition of ∇v into its symmetric component which describes the strain

$$\mathbf{S} = \frac{1}{2}(\nabla v + \nabla v^T) \quad (5)$$

where $S_{ij} = \frac{1}{2} \left(\frac{\delta u_i}{\delta x_j} + \frac{\delta u_j}{\delta x_i} \right)$ and rotational components equate to zero. When considering strain in fluid mixing, we are most interested in the shearing force driving the deformation. Shear occurs when two or more infinitesimally close vectors within the flow field f are parallel but of different magnitude. Particles seeded on these vectors travel along trajectories parallel to one another while the distance between them increases, see Figure 3. To give a measure for the difference in the vector field magnitude causing the shearing, the angle between the direction of strain and the direction of the particle is found [15]. It is important to distinguish shearing from other types of divergence in order to properly identify flow separation. Eliminating the rotational component of deformation, the maximum eigenvector of Eq. (5) will give the desired strain direction.

While the angle between the strain and particle direction provides a metric for shearing magnitude in 2D, it may not capture all shearing cases that occur in 3D. If the majority of the strain force is equally distributed along two of the three axes, while a smaller strain is experienced in the third, the maximum eigenvector of the strain tensor would give an arbitrary direction. To isolate the strain magnitude in 3D, we take a different approach, looking at the force of separation. Because we are most interested in deformation effects along a pathline, the greatest effect from a separation force would occur on the plane orthogonal to the particle direction. This plane can be found using the particle's position and its velocity as the plane normal. The separation magnitude along this plane is then just the largest eigenvalue of the projected deformation tensor. Shear is represented as $SEP(\mathbf{T}_S)$ at each time step where \mathbf{T}_S is the strain component of the deformation tensor. The accumulation of this metric is done by taking the average of the shear over all time steps.

5.4 Rotation

Completely eliminating separation forces, the anti-symmetric part of the velocity gradient tensor focuses on rotation. Decomposing ∇v into its anti-symmetric component describes the deformation due to rotation:

$$\boldsymbol{\Omega} = \frac{1}{2}(\nabla v - \nabla v^T) \quad (6)$$

where $\Omega_{ij} = \frac{1}{2} \left(\frac{\delta u_i}{\delta x_j} - \frac{\delta u_j}{\delta x_i} \right)$ which is commonly referred to as the *vorticity* or the tendency of particles in a fluid to rotate (Figure 3). The length of this vector gives the angular velocity of the rotation.

In a 2D setting, the vorticity describes the rotation on the plane itself and the accumulation of this metric gives total rotation of a particle around the z-axis. However, in 3D the physical meaning becomes unclear with respect to the pathline as the axis of rotation may change at every time step. Accumulating the angular velocity would have no clear meaning seeing as every plane of rotation could vary. Instead, we have chosen to focus on the twisting a particle experiences while traveling along its trajectory. In other words, we are interested in the angular velocity on the plane orthogonal to the pathline. Therefore, the deformation gradient tensor is projected onto this plane and the projection is then substituted into Eq. (6) for the calculation of vorticity. Summing the absolute vorticity at every time step now serves to show how the particle twirls along its trajectory, referred to as $ROT_A(\mathbf{T}_\Omega)$ where \mathbf{T}_Ω is the anti-symmetric component of the accumulated deformation tensor [19].

6 Metric Combination

Even when given only the basic characteristics of flow deformation, their combination can further the analysis of the fluid flow. Beginning with the strain component of the velocity gradient, the separation can be broken down into three cases. In the first case, shearing occurs along the particle's flow direction, thereby resulting in a low separation magnitude on the plane orthogonal to the particle's velocity vector. A high separation magnitude results from shearing that occurs orthogonal to the flow direction, categorized as the second case. The final case occurs when there is no notable shearing, resulting in a median separation magnitude, irrelevant to the metric purpose. However, this third case can be isolated using the fractional anisotropy. Volumetric tensor effects having a fractional anisotropy near 0 also categorize those volumes that experience little to no shearing. Weighting the shearing metric by fractional anisotropy using the following equation:

$$SEP_{FA}(\mathbf{T}_S) = FA(\mathbf{T})^3 \times SEP(\mathbf{T}_S) \quad (7)$$

filters out the third case from the separation metric. Furthermore, this filtered separation can then be applied to the divergence rate found by FTLE.

Recall that the FTLE gives the maximum separation rate of infinitesimally close particle trajectories over a finite time, referred to here as the maximum separation magnitude of the neighborhood surrounding a particle along its trajectory. While certain methods, especially Lagrangian Coherent Structure extraction, focus on the separation due to changes in flow direction, FTLE is also sensitive to changes in flow magnitude (Please refer to [15] for an in-depth explanation). The change in flow magnitude is isolated by the separation magnitude due to shearing. Therefore, filtering the FTLE by the SEP_{FA} metric can better express FTLE in terms of changes in flow direction. This is done by a direct linear filtering after the accumulated shearing magnitudes are normalized between 0 and 1:

$$FTLE_A(\mathbf{T}) = SEP_{FA}(\mathbf{T}_S) \times FTLE_A(\mathbf{T}) \quad (8)$$

where $FTLE_A(\mathbf{T})$ is the separation rate of the accumulated tensor \mathbf{T} along the trajectory.

Applying the rotation metric to FTLE can also provide insight into the flow field. Directly combining both metrics:

$$PLANAR_SEP(\mathbf{T}) = FTLE_A(\mathbf{T}) + ROT_A(\mathbf{T}_\Omega) \quad (9)$$

reveals the maximum separation due to planar separation.

7 Results

In the following we present examples that illustrate the velocity gradient-based descriptors extracted with our technique. We study these metrics using two time-varying data sets, one depicting a jet flow and the other a Karman vortex street. The Jet Flow data set has dimensions $128 \times 256 \times 128$, while the Karman data set is of size $167 \times 34 \times 34$. Both are represented as uniform grids. Particles are seeded uniformly to cover all grid spaces and advected using Runge-Kutta 4th/5th order integration for a time $T = 5$. The velocity gradient tensor is calculated 10 times per time step (a total of 50 per trajectory), using central differencing. The deformation gradient tensor is then accumulated, along with shear and rotation statistical measures. With respect to performance, the additional workload of computing the velocity gradient and solving for the eigenvectors and eigenvalues is nominal compared to the traditional computational load associated with finding nearest neighbors for pathline advection, i.e. cell location. All images were created using the Voreen volume visualizer [24].

7.1 Shear and Fractional Anisotropy

To visualize shearing effects, we compute the separation magnitude due to shearing in the flow, computing this magnitude on the plane orthogonal to the particle's trajectory. Mentioned earlier, this descriptor also provides magnitudes in areas of little to no shear, colored in cyan in Figure 4(d). This coloring is given a slight opacity to show that these values occur in areas outside of the flow structure boundary. Filtering the shearing magnitude by the fractional anisotropy (SEP_{FA}) eliminates the unwanted values, giving way to Figure 4(b). Stretching orthogonal to the flow direction is shown in purple, while stretching along the particle's trajectory is shown in orange. We see that the highest amount of shearing along the particle's trajectory occurs in the middle of the plume, revealing the difference in magnitude as the flow billows from the jet. The highest amount of shearing orthogonal to the trajectory is shown to occur on the outside of the plume, outlining areas where the moving flow interacts with the still surroundings, causing vortex-like features. Fractional anisotropy can also be used as a filter for FTLE in compressible data sets, allowing to distinguish between divergence and uniform expansion.

While providing insightful information into the fluid flow on its own, the shearing magnitude can also be used as a filter for the stretching magnitude found using the FTLE. In certain applications, flow divergence due to time-varying separation is preferred over general flow divergence. However, the FTLE does not distinguish between

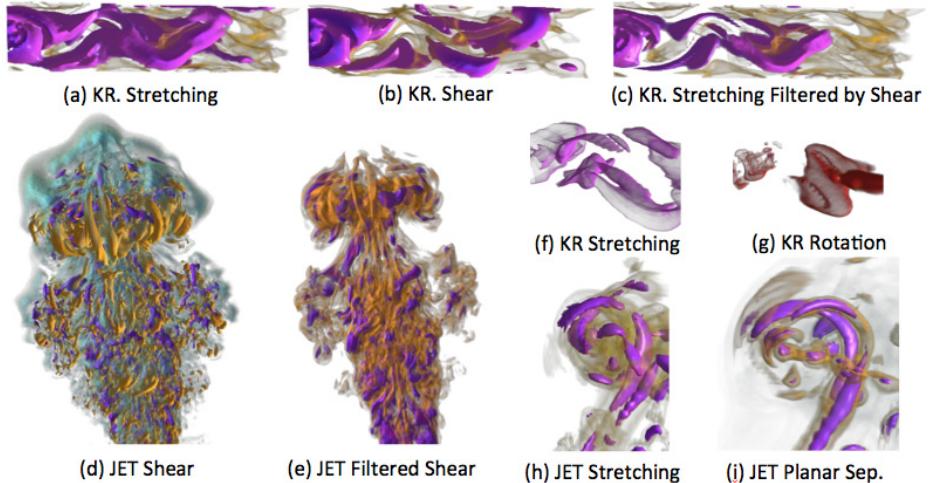


Fig. 4. Karman (KR) and Jet data sets (Low - cyan, mid - orange, high - purple). (a) shows mid to high FTLE values. (b) shows shearing filtered by fractional anisotropy. (c) shows the filtering of FTLE to eliminate divergence due to shearing, clearly displaying the two manifolds. (d) shows separation magnitude due to shearing. Shearing along the particle's trajectory is shown in orange, orthogonal shearing in purple, and areas of no shearing in cyan. (e) shows the same shearing descriptor filtered by fractional anisotropy to eliminate areas of no shear. (f) shows the FTLE which clearly discerns between two manifolds formed by flow going around the obstacle. (g) shows the rotation (in red) that occurs when particles following the divergence of both manifolds meet and rotate due to differences in speed. (h) shows the FTLE metric of the top of the plume hinting at rotation. (i) exposes the rotating flow features in more detail by showing planar separation.

the two. Filtering the FTLE by the separation magnitude due to shearing can eliminate the later component due only to differences in flow magnitude ($FTLE_A$). Figure 4(a) shows the FTLE metric of the Karman data set, with mid to high values generously covered in purple. Figure 4(b) shows the separation magnitude due to shearing, already filtered by fractional anisotropy. Following the filtering scheme described in Section 6, the result of filtering the FTLE descriptor by the shear descriptor is shown in Figure 4(c). The exact same volume transfer function is used in both 4(a) and 4(c), highlighting the same range of FTLE values in purple, instantly revealing the two distinct manifolds in the latter by eliminating divergence due to shearing. This filtering technique can be applied to the mixing of multiple fluids, capturing the boundary between them caused by a difference in velocity during fluid interaction, while eliminating the separation due to shear slightly beyond the boundary layer.

7.2 Rotation and Stretching

Rotation is not synonymous with stretching and therefore can illustrate flow effects not present in an FTLE-based visualization, a method favored for the visual classification of turbulent flow. Particles that diverge according to the two prominent manifolds in the Karman data set should, in theory, meet and interact. If one front moves faster than the other, this interaction should result in a rotation. Figure 4(f) shows only the flow

divergence in purple, while Figure 4(g) shows only rotation in red. This image shows rotation occurring when the particles following the two divergence structures interact, confirming the theory. Being able to identify isolated rotating pockets is crucial for analysis and optimization, where such islands tend to degrade mixing quality.

The jet flow data set, more turbulent than the Karman, does not often have areas where rotation is completely separate from stretching. Therefore, the planar separation (*PLANAR_SEP*) is a more interesting measure, combining the stretching and rotation descriptors. Figure 4(h)(i) shows two images side by side, both of the same area covering a section of the plume. Figure 4(h) is the FTLE while Figure 4(i) shows high values of planar separation. All high values are colored purple for both. By extracting the planar separation, the rotational separation formed by the plume when it closes in on itself is better displayed, connecting the divergence structures in question. Figure 4(i) visualizes the twisting effects experienced by particles traversing the plume.

8 Conclusion and Future Work

We have presented a unified framework for the identification and extraction of meaningful flow features along a particle’s trajectory as it passes through a time-varying vector field. Adding to the commonly studied transport of fluid, we have shown that measuring rotation and shear along a particle’s trajectory can give insight into the mixing behavior. Furthermore, coupled with flow divergence, these metrics help identify specific areas of interest, highlighting areas of divergence due to shearing and observing areas of lower divergence with an emphasis on rotation.

We plan to extend this work to methods that use the velocity gradient as a building block in a Lagrangian setting. The metrics we have chosen to focus on are not the only meaningful characteristics of flow deformation that can be extracted from the velocity gradient tensor. For instance, Lagrangian Coherent Structures (LCS) and vortex cores are detected using components of the velocity gradient tensor. Along with an extension to other gradient methods, we plan to explore various flow visualization techniques for the deformation metrics to better combine them with the method’s results.

Acknowledgment. This work was supported in part by the NSF (IIS 0916289 and IIS 1018097), and the Office of Advanced Scientific Computing Research through DOE ‘Scientific Discovery through Advanced Computing’ (SciDAC) contract DE-FC02-06ER25780, and contract DE-FC02-12ER26072 (SDAV).

References

1. Globus, A., Levit, C., Lasinski, T.: A tool for visualizing the topology of three-dimensional vector fields. In: IEEE Conf. on Vis. 1991, pp. 33–40, 408 (1991)
2. Ropinski, T., Oeltze, S., Preim, B.: Visual computing in biology and medicine: Survey of glyph-based visualization techniques for spatial multivariate medical data. Comp. and Graph. 35, 392–401 (2011)
3. Wiebel, A., Koch, S., Scheuermann, G.: Glyphs for non-linear vector field singularities. In: TopoInVis, pp. 177–190. Springer, Heidelberg (2012)

4. de Leeuw, W.C., van Wijk, J.J.: A probe for local flow field visualization. In: Proc. of the 4th Conf. on Vis. 1993, pp. 39–45 (1993)
5. Haller, G.: Finding finite-time invariant manifolds in two-dimensional velocity fields. *CHAOS* 10, 99–108 (2000)
6. Obermaier, H., Hering-Bertram, M., Kuhnert, J., Hagen, H.: Volume deformations in gridless flow simulations. *Comp. Graph. Forum* 28, 879–886 (2009)
7. Obermaier, H., Joy, K.I.: Derived metric tensors for flow surface visualization. *IEEE Trans. on Vis. and Comp. Graph.* 18, 2149–2158 (2012)
8. Haller, G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena* 149, 248–277 (2001)
9. Sadlo, F., Peikert, R.: Visualizing lagrangian coherent structures and comparison to vector field topology. In: *TopoInVis* (2007)
10. Garth, C., Gerhardt, F., Tricoche, X., Hagen, H.: Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Trans. on Vis. and Comp. Graph.* 13, 1464–1471 (2007)
11. Garth, C., Wiebel, A., Tricoche, X., Joy, K., Scheuermann, G.: Lagrangian visualization of flow-embedded surface structures. *Comp. Graph. Forum* 27, 1007–1014 (2008)
12. Theisel, H., Weinkauf, T., Hege, H.C., Seidel, H.P.: Topological methods for 2d time-dependent vector fields based on stream lines and path lines. *IEEE Trans. on Vis. and Comp. Graph.* 11, 383–394 (2005)
13. Kasten, J., Petz, C., Hotz, I., Noack, B.R., Hege, H.C.: Localized finite-time lyapunov exponent for unsteady flow analysis. In: *VMV*, pp. 265–276 (2009)
14. Jiang, M., Machiraju, R., Thompson, D.: Detection and visualization of vortices. In: *The Vis. Handbook*, pp. 295–309. Academic Press (2005)
15. Pobitzer, A., Peikert, R., Fuchs, R., Theisel, H., Hauser, H.: Filtering of FTLE for visualizing spatial separation in unsteady 3d flow. In: Peikert, R., Hauser, H., Carr, H., Fuchs, R. (eds.) *TopoInVis*, pp. 237–253. Springer (2012)
16. Haimes, R., Kenwright, D.: On the velocity gradient tensor and fluid feature extraction. Technical Report AIAA Paper (1999)
17. Kasten, J., Hotz, I., Hege, H.C.: On the elusive concept of lagrangian coherent structures. In: *TopoInVis*, pp. 207–220. Springer, Heidelberg (2012)
18. Schroeder, W.J., Volpe, C.R., Lorensen, W.E.: The stream polygon: a technique for 3d vector field visualization. In: Proc. of the 2nd Conf. on Vis. 1991, pp. 126–132. IEEE Computer Society Press (1991)
19. Ueng, S.K., Sikorski, C., Ma, K.L.: Efficient streamline, streamribbon, and streamtube constructions on unstructured grids. *IEEE Trans. on Vis. and Comp. Graph.* 2, 100–110 (1996)
20. Salzbrunn, T., Scheuermann, G.: Streamline predicates. *IEEE Trans. on Vis. and Comp. Graph.* 12, 1601–1612 (2006)
21. Salzbrunn, T., Garth, C., Scheuermann, G., Meyer, J.: Pathline predicates and unsteady flow structures. *The Visual Comp.* 24, 1039–1051 (2008)
22. Fuchs, R., Peikert, R., Sadlo, F., Alsallakh, B., Gröller, E.: Delocalized unsteady vortex region detectors. In: *VMV*, pp. 81–90 (2008)
23. Obermaier, H., Billen, M.I., Hagen, H., Hering-Bertram, M., Hamann, B.: Visualizing strain anisotropy in mantle flow fields. *Comp. Graph. Forum*, 2301–2313 (2011)
24. Meyer-Spradow, J., Ropinski, T., Mensmann, J., Hinrichs, K.H.: Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Comp. Graph. and App.* 29, 6–13 (2009)

Visual Access to Optimization Problems in Strategic Environmental Assessment

Tobias Ruppert, Jürgen Bernard, Alex Ulmer,
Arjan Kuijper, and Jörn Kohlhammer

Department of Information Visualization and Visual Analytics,
Fraunhofer Institute for Computer Graphics Research,
Fraunhoferstrasse 5, 64283 Darmstadt, Germany
`{tobias.ruppert,juergen.bernard,alex.ulmer,arjan.kuijper,
joern.kohlhammer}@igd.fraunhofer.de`

Abstract. The complexity of actual decision making problems especially in the field of policy making is increasing due to conflicting aspects to be considered. Methods from the field of strategic environmental assessment consider environmental, economic, and social impacts caused by political decisions. This makes the analysis of reasonable decisions more complex. Mathematical models like optimization can help to balance conflicting aspects. Although they are not easy to understand, these complex models and the resulting policy options have to be reviewed by the decision makers. In this work we present a visual-interactive interface to an optimization system capable of solving multidimensional decision problems. The interface enables visual access to the complex optimization models, and the analysis of alternative solutions. As a result strategic environmental assessment can be included in the decision making process. An evaluation in the domain of regional energy planning underlines the usability and usefulness of the visual interface.

1 Introduction

Visualizing information helps humans to understand problems. Perception science has shown that the human visual system can grasp information faster if relevant aspects are visually highlighted [1]. This is especially important when difficult decisions have to be made and a good knowledge foundation is crucial. Besides supporting informed decisions, visualization can help in communicating important information during and after the deciding making process. This makes decisions more transparent for a broad group of stakeholders and enables constructive feedback and creative discussions.

Policy making is a domain where the scope of decisions is broad and the impact can affect large parts of the society. At the same time, influencing factors like environmental, economic, and social aspects have to be considered. In recent years environmental and social impacts gained more importance so that strategic environmental assessment (SEA) was enforced by law in many countries. The goal of SEA is to analyze impacts on the environment caused by political

decisions. This results in a complex set of relationships that demands a scientific analysis. Multiple authors from the field of SEA claim that the methods to analyze environmental impacts are not well integrated into the policy making cycle [2–4]. One reason for this is the complexity of SEA systems that usually depend on many influencing ‘dimensions’. The authors suggest that SEA concepts have to be presented more clearly and robust to the decision makers.

One possible way to address multidimensional decision problems is mathematical optimization. Established models and algorithms that support multidimensional problems exist, but they have to be transferred to the policy making process. Recent attempts in the optimization domain have created models which support multidimensional decision making with integrated environmental assessment, but most of them are lacking a visual interface to make the results accessible to non-experts [5–7]. SEA could provide even more value to the decision making process if decision makers could directly work with the models.

In our approach, we introduce a visual interface to multidimensional optimization models. Goal of this interface is to reduce the complexity of the underlying optimization models in order to provide non-IT-experts an intuitive access to advanced analysis functionality. The user can interactively adjust input parameters, and analyze the resulting alternative solutions. As a use case our visual interface is coupled with a SEA model designed for creating regional energy plans. The underlying optimization model concerns environmental, social and economic impacts of these energy plans on a regional level. The results of this optimization are interpreted by decision makers as well as domain experts, and may conclude in policy options to be considered in the policy making process. The visual interface is also designed to overcome knowledge gaps between different stakeholders. With our approach collaboration between decision makers and domain experts is facilitated due to a common information base provided by the visualization. The visual-interactive design makes use of an optimization model to compute solutions and provides clearly structured information about environmental impacts as requested by [3].

2 Related Work

We first review relevant information visualization and visual analytics approaches. Secondly, we target on strategic environmental assessment on the policy level, and optimization in general as the application domains of our approach.

2.1 Information Visualization and Visual Analytics

In our approach we focus on the visualization of quantitative data. Fundamental guidelines for visualization techniques in this field can be found in [8]. In this work, Few filtered and sorted the pre-attentive attributes described by Ware [1] by their precision of quantitative perception. He concludes that bar charts are an appropriate choice for the comparison of quantitative data.

Despite the need for visualization approaches in the policy making domain [9], only a small number of examples exist. One approach for supporting decision making with multiple stakeholders can be found within the Vismon application [10]. The application enables the users to analyze simulated fishery data. Decision makers can analyze the health of fish populations. The LEAP system implements an SEA approach in the field of energy planning [11]. It provides functions to analyze energy consumption, production and resource extraction while monitoring the resulting greenhouse gas emissions. Further approaches that are concerned with energy efficiency can be found in the survey of Markovic et al. [12]. A further work to be mentioned is the visual-interactive tool ComVis created by Matkovic et al. to assist the engine design process by optimizing diesel injection [13]. The optimal parameter set up for an engine construction is analyzed via visual-interactive simulation and optimization systems.

2.2 Strategic Environmental Assessment for Policy Making

Strategic environmental assessment (SEA) is a proactive approach for integrating environmental concerns into early phases of decision making. The target is to anticipate and prevent environmental, economic and social damage by predicting the impacts. Especially in the policy making process where contrary options have to be evaluated SEA concepts can be applied profitably. The first SEA system was already established in 1969 by The US National Environmental Policy Act (NEPA). With the first SEA system applied by the World Bank in 1989 the acknowledgement rose, and more countries started to make use of SEA [4, 14]. Nowadays, SEA is an approved methodology and is used in many countries [3]. In [2], Fischer postulates three meanings of SEA: a) a systematic decision support process, b) an evidence-based instrument for scientific assistance in policy making, and c) a framework for the better consideration of alternative policy options for sustainable development. In [3], Therivel additionally recommends an increasing participation and collaboration of multiple domains in the policy making process.

Optimization models can describe complex decision making problems. To make the results understandable for non-domain-experts visualization can help as suggested in Jones' work [15]. Multiple approaches that consider environmental, economic and social impacts were submitted by the domain of optimization, since multi-objective optimization models are able to support these problems [6, 16, 17]. Further authors have combined the policy making process with optimization and environmental assessment like You et al. for optimizing bio-fuel supply chains [6], or Lim et al. for water infrastructure optimization [7]. They have committed models that are able to solve multidimensional decision making problems. Yet, they lack visual interfaces that could enable the involvement of decision makers into the process. The target of SEA should be to deliver robust data and clearly presented information [3]. In [18], the authors describe how information visualization and visual analytics techniques may contribute to the policy making process by bridging the knowledge gaps between different stakeholders involved in the policy making process.

3 Approach

In our approach we introduce a visual interface to provide access to multidimensional optimization models for SEA. In our use case the optimization model tackles the problem of defining an optimal energy plan on a regional level. This complex mathematical model is difficult to understand by non-modeling-experts like policy makers. To address this challenge, we connect visualizations to the model which facilitates the access for policy makers. In the following, we first briefly describe the model with its variables, dependencies, possible target functions, and constraints. Then, we summarize the user requirements coming from policy makers and domain experts. And finally, we present the visual designs to enable the visual access to the multidimensional optimization problem.

3.1 Modeling a Regional Energy Plan

In this section, we describe our collaborative approach with the goal to find an optimal energy plan on a regional level. The resulting energy plan consists of a set of energy sources (primary activities), that each produces a specific amount of energy. The plan also includes secondary activities needed for the installation of the respective energy sources (e.g. aerial power lines, dams, etc.). Multiple aspects have to be considered in this scenario. The government has only a limited budget to incentivize the construction of new plants. Still, a defined minimum of energy has to be produced. There are multiple types of energy sources that can be installed. Some are more efficient, others are more sustainable. Moreover, each region has geographical characteristics that restrict some types of energy sources. In addition, governmental laws have to be observed. Often, they aim at the protection of nature and prohibit the extensive use of polluting energy plants. This is also a demand of the society, which is directly affected by the impact of new policies. In summary, considering all dependencies and finding a solution that satisfies all constraints results in a multidimensional decision problem.

In our approach, we made use of an optimization model to address this multidimensional decision problem. More specifically, a linear optimization model is used, designed by modeling experts, which can be reviewed in [5]. Please note, that our visual designs only consider input parameters and output data of the optimization model. Hence, it can be easily adapted to other linear, and even non-linear optimization problems. A linear optimization problem can be mathematically described as $\max(\mathbf{c}^T \mathbf{x} | A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0)$. In our case the vector \mathbf{x} to be optimized consists of the amount of energy to be produced by each of the energy sources included in the model. $\mathbf{c}^T \mathbf{x}$ defines the target function to be maximized. Thereby, \mathbf{c} encodes the target of the optimization problem, e.g. overall energy produced, overall cost, impact on an environmental receptor, etc. $A\mathbf{x} \leq \mathbf{b}$ encodes the constraints on the problem. Thereby, similar to \mathbf{c} , each row of the matrix A together with the boundary value comprised by \mathbf{b} describes a constraint. After the definition of the optimization model, an optimal solution vector \mathbf{x} can be calculated, if a solution exists. This vector comprises the optimal amount of energy to be produced by each energy source.

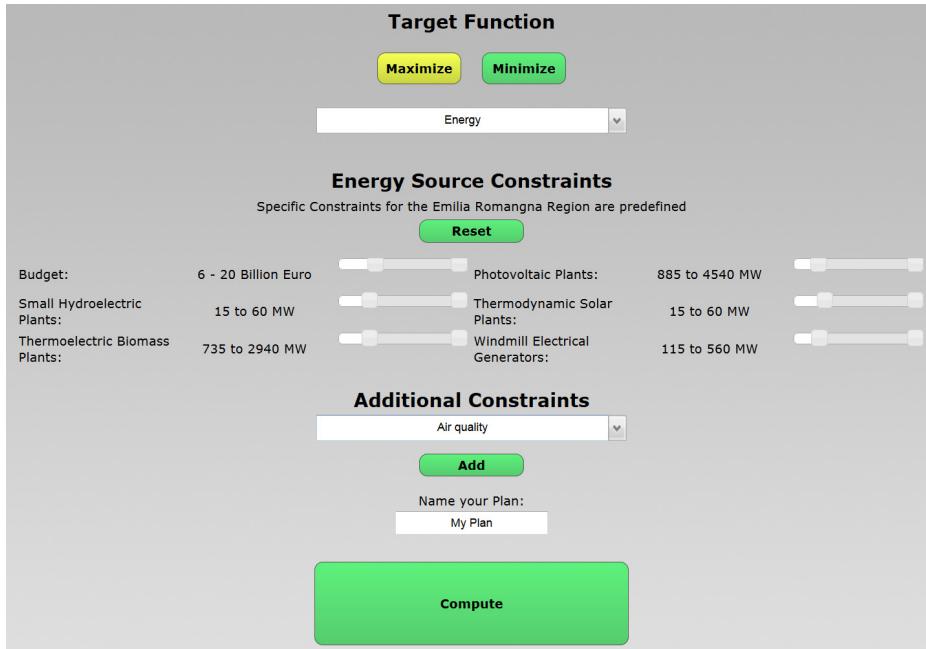


Fig. 1. The Input Interface for specifying target function (e.g. maximize energy), constraints on budget and energy source capacities (e.g. biomass plants capacities), and constraints on environmental impacts caused by the energy plan (e.g. air quality)

3.2 User Requirements

At the beginning of our approach, requirements were identified with the user groups of policy makers, domain experts and modeling experts. Moreover, a questionnaire was sent to the potential user groups to confirm the identified requirements, and determine further refinements. As a result of this requirements analysis, the final requirements for this approach have been defined as follows:

- R₁** Visual definition of target function and constraints
- R₂** Visualization of calculated optimized energy plan
- R₃** Comparison of energy plans
- R₄** Consideration of environmental, economic and social impacts

3.3 Visual Designs

Based on the results of the requirements analysis phase, we present a web-based system for the visual access to multidimensional optimization models in the application field of strategic environmental assessment. The various input parameters of the model can be defined in a visual-interactive manner. This encapsulation of the optimization model itself via visual interfaces helps to reduce

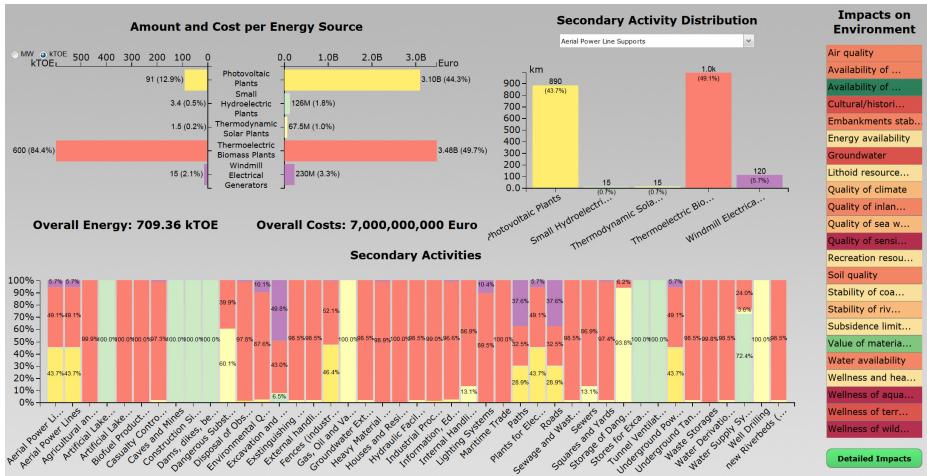


Fig. 2. The Optimized Plan View visualizes the output data of the optimization. It depicts the quantity and costs of energy to be produced per source, impacts on the environment, and additional secondary activities needed for the realization of the plan.

the complexity of the input space. The output space of the model is represented in a two-stage design. Firstly, a result visualization gains insight into the output data of the model. Secondly, an interface to visually compare results of different parameterizations helps to determine optimal input parameter setups. In the following, the visual designs of the web-system are described.

The Input Interface (see Figure 1) makes available all possible degrees of freedom of the optimization model. The user is enabled to define target function and constraints to specify the optimization problem (see \mathbf{R}_1). The visualization provides three sections to address these tasks. In the upper part the target variable to be maximized or minimized can be chosen. Below constraints on the energy sources can be specified. Please note, that in our use case as a maximum value for each energy source the available regional capacity of each respective source is set. Hence, the user can refine these constraints within the range of 0 and the maximal regional capacity. Moreover, additional constraints on the environmental, social, and economical impacts can be set. Finally, the specified parameters can be labeled as a plan, and the optimal solution can be calculated.

The Optimized Plan View (see Figure 2) visualizes the output data of the calculated energy plan based on the inputs defined in the Input Interface. This addresses requirement \mathbf{R}_2 . It gives an overview of the amount of energy (in megawatt or kilotons of oil equivalent) and the costs to be produced by the plan (top left). Additionally, environmental, social and economical impacts are displayed (top right), which addresses requirement \mathbf{R}_4 . Moreover, the secondary

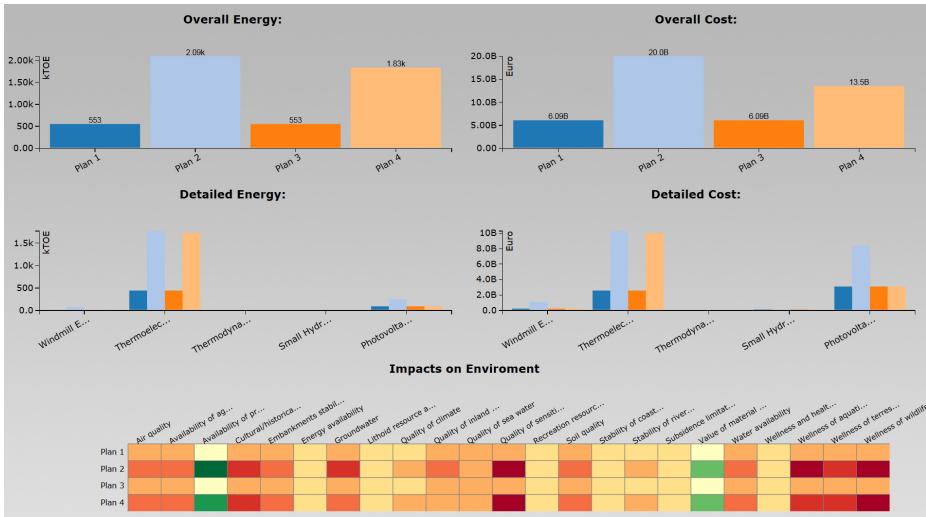


Fig. 3. The Compare View enables the user to compare plans calculated with different input parameters. Here, four different plans are depicted. The user can compare the overall energy and costs to be produced by each plan, detailed information about the energy source mix, and impacts on the environment.

activities needed for the installation of the energy sources are depicted (top middle and bottom).

The information in this view consists of nominal and quantitative data. Hence, we chose bar charts as visualization technique, as proposed in the literature [8]. Moreover, this technique is easy to understand for non-IT-experts. A normalized stacked bar chart depicts the percentages of secondary activities needed for the installation of primary activities (energy sources). The impacts are depicted via a heatmap to save display space. The values of the different impact types cannot be compared because they are measured in different units. If requested, more detail on the impacts can be viewed in the Impacts View, an additional matrix heatmap visualization mapping secondary activities (rows) on impacts (columns). For the visualizations we used an evaluated categorical color map [19] to depict the nominal data labeling the distinct energy sources. A diverging color map is used to depict the quantitative impact values in the heatmap; negative (red), neutral (white), or positive (green).

The Compare View (see Figure 3) visualizes a set of energy plans the user wants to compare. This view covers functional requirement R_3 . To compare the different facets of the plans each variable is visualized separately. The top layer allows the user to get a fast overview of the compared plans by presenting the overall energy, and the overall costs produced by each plan via bar charts. The middle layer of the visualization splits the energy produced and the costs

into the energy sources and displays them as grouped bar charts. The heat map, also presented in the Optimized Plan View, shows the different impacts on environment, society and economy, and therefore addresses requirement **R**₄.

4 Case Study

We conducted a case study to evaluate the outcome of our approach. In the following, we describe the methodology and the results of our evaluation.

4.1 Experimental Design

The system evaluation was hampered by the fact that, to the best of our knowledge, no comparable visual interfaces for SEA could serve as a baseline to evaluate against. Neither the targeted policy analysis domain, nor the applied model to solve multidimensional optimization problems have been presented in a comparable manner. For this reason, the experiment was designed towards an evaluation of the targeted purpose of this approach. We considered a) the validation of the functional user requirements and b) the verification of the visual encodings of the system as the two important factors to prove the success of the system.

We conducted an informal summative case study based on task-completion tests and a user questionnaire. The field-based case study was conducted with our web-based application to enable a real-world setup. The unsupervised environment enabled the participants to perform the tests without being influenced, in order to give credible feedback. Model parameterizations for the respective tasks were carefully selected and tested in a previously applied test-run to control the resulting data values and their visual representations. The dependent variable of the case study was the task-completion. Additionally, qualitative feedback based on user preference was gathered within the user questionnaire.

Task-Completion Test. We designed the tasks to validate the functional user requirements (cf. Section 3.2). The tasks were chosen by means of covering each functional component. By defining the task setup, attention was paid to ensure that each task covers an ‘atomic’ functional unit of the analysis workflow. This enabled the identification of bottlenecks in the analytical workflow and a target-oriented enhancement thereof. We were interested in what way non-expert users would be able to comprehend the analytical context of the domain-specific model. In this way, statements regarding the ‘generalizeability’ of the case study became possible. First of all, the user had to calculate an energy plan with the default input parameters. Then, she had to solve task 1 and task 2 with the Optimized Plan View (**R**₂). The first task should test, whether she is able to find the relevant information displayed. The second task should test whether the user discovered the additional information in the heatmap by using the mouse-over tooltip. Next, the user had to apply the detailed impact view to solve task 3 (**R**₄). Task 4 could be solved by understanding the remaining visualization of the Optimized Plan View (**R**₂). Afterwards, the user had to specify an alternative plan in the Input

Interface by changing some of the default constraints (\mathbf{R}_1). The two calculated plans were compared by the user in the Compare View to solve tasks 5 and 6 (\mathbf{R}_3). The six tasks to cover the functional requirements were:

1. Which energy source in the plan produces the lowest amount of energy?
2. On which receptor the most negative impact is induced? What is its value?
3. What has the most negative impact on ‘quality of sensitive landscapes’?
4. Which source needs the highest amount of ‘aerial power line supports’?
5. Which energy plan has a more positive impact on the air quality?
6. Which of the two plans produces more energy?

Questionnaire. In addition to the task-completion tests, we provided a questionnaire that enabled the participants to give qualitative feedback. It targeted the usability of the system. Questions of concern were the verification of the visual encodings and interactive capabilities. First, questions about the participants’ profession, their domain of expertise and their common analysis tasks, were asked. Additionally, the questionnaire incorporated eleven closed questions about the visual encodings and the usefulness of each view. Finally, we provided open questions to gather informal feedback of individual user preferences.

Participants. Ten non-domain experts agreed to participate voluntarily in the task-completion test. All of them had a profound background in information visualization. None of them had expertise in the energy domain. Thus, the results of the task-completion test were not influenced by domain knowledge which allows the generalization of the results. The questionnaire was answered by all non-experts, and two additionally recruited experts from the energy domain with no expertise in visualization. Most of the participants were male, none of them reported color blindness. The age of the participants reached from 22 years to 38 years with a median of 28 years. All of the participants had an academic degree.

4.2 Results of the Task-Completion Tests

Figure 4 shows the results of the task-completion tests. Tasks one, three and four were completed correctly by most of the participants while tasks two, five and six had higher error rates. The tasks with the low error rates were completed by using the Input Interface, the Optimized Plan View and the Impacts View. For the tasks with the higher error rates the users had to make use of an additional Overview and the Compare View. The Overview consisted of a scatterplot visualizing all computed plans with respect to their overall energy produced and overall costs. In the Overview the user had to select a set of plans for comparison. The usability test shows that on average 90% of the first four tasks were completed correctly but only 65% of the last two tasks. This concludes that the users were able to use the Input Interface and calculate optimized energy plans. Thus requirement \mathbf{R}_1 is fulfilled. The Optimized Plan View supports the analysis of energy plans and the Impacts View displays the environmental impacts.

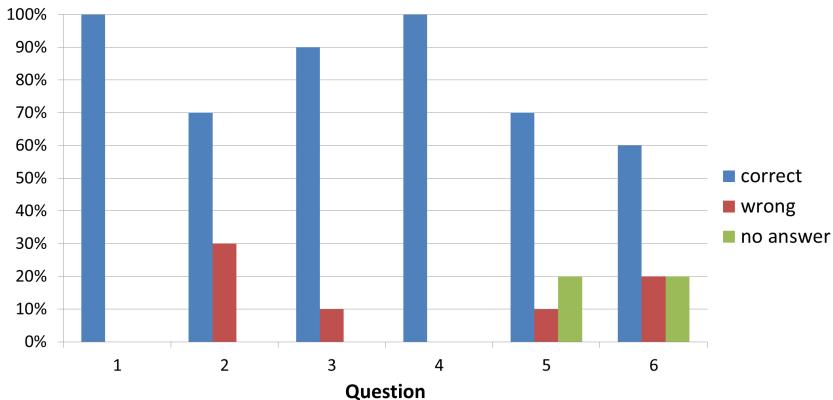


Fig. 4. Results of the task-completion tests

The tasks 1-4 related to these views were completed correctly by most of the users. Thus, requirements R_2 and R_4 are met. The results of the last two tasks indicated that the exploration of multiple energy plans and the following comparison is not as easy as the previous tasks. To solve the task the Overview and the Compare View had to be used. The error rates of the last two tasks showed that the visual-interactive design of these views did not support the user as well as the other views. Thus the requirements R_3 and R_4 are met but the solutions have to be improved.

4.3 Results of the Questionnaire

The questionnaire was designed to test the usability of the system with respect to the visual encodings and interactive capabilities. For each view the users were asked regarding its intuitiveness and usefulness. For all views, except for the Overview, above 80% of the users stated that they are easy to understand. Above 70% of the users argued that the views are useful, the Compare View even gained 100% of the users approval. The Overview was only intuitively understood by 50% of the users. Its usefulness was only affirmed by half of the users. Hence, we assume that the lower task-completion rates of tasks 5 and 6 are caused by the inappropriate design of the Overview, not the Compare View. As a result the Overview was excluded from the application after the test. Calculated energy plans are now directly added to the Compare View.

5 Discussion and Future Work

In this section, we summarize lessons learned during the case study and the collaboration with policy makers and domain experts. Additionally, we have a look at possible improvements and functional extensions for our approach.

We share Few's view with regard to his choice of techniques [8]. All views designed with bar charts were easy understood by all user groups. That way, our system is applicable by all relevant stakeholders. Still, we learned that different stakeholders need different visual designs with regard to the amount of information presented. While domain experts were satisfied with several visualizations in one view, the policy makers only want selected output data to be presented. Further variables should only be shown, if they exceed critical values (e.g. a scatterplot matrix depicting relations between variables was considered too complex). This is caused by the fact, that policy makers do not have time to learn complex applications, while domain experts need higher functionalities and would adopt to more informative visualizations.

For future work, we will provide an user-adaptive visual interface for policy makers with a focussed functionality. The user will be supported by customized views, depicting only selected information. Still, the policy maker view and the domain expert view have to build a coherent basis for communication, which has to be considered during the design process. Moreover, we will prove the generalizability of the system and apply it to different use cases and non-linear optimization problems. As another issue, sensitivity analysis may be included in the interface, since politicians are often interested in adjusting optimal solutions. The complexity introduced to the system may be reduced by user-guidance techniques.

6 Conclusion

In this approach, we combined the capabilities of information visualization and optimization in order to support the policy making process with methods from SEA. Firstly, we addressed the demand to ensure that SEA is more effective in policy making, by presenting the information clearly to the decision makers, as stated in [3]. Secondly, we addressed the demand to provide access to an optimization model for non-IT-experts. Therefore, we developed a visualization-tool that connects to an optimization module [5], using approved concepts of information visualization [8] to reduce the complexity of the data created by the optimization. As a result, modeling experts could spot errors in the model and energy experts were able to validate the model. Especially the policy makers, who had to rely on the knowledge of experts, were able to understand the process behind creating an energy plan, and thus could make better founded decisions.

In conclusion, the demand for finding solutions for complex decision making processes will expand in the future, and thus will the need for clear information presentation. Information visualization can help domain experts to communicate their results as well as allow decision makers to use complex optimization models.

Acknowledgements. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 288147. <http://www.epolicy-project.eu/>

References

1. Ware, C.: *Information Visualization: Perception for Design*. Morgan Kaufmann series in interactive technologies. Elsevier Science (2004)
2. Fischer, T.B.: The theory and practice of strategic environmental assessment: towards a more systematic approach. *Earthscan* (2007)
3. Therivel, R.: *Strategic Environmental Assessment in Action*. Taylor & Francis (2012)
4. Dalal-Clayton, D., Sadler, B.: *Strategic Environmental Assessment: A Sourcebook and Reference Guide to International Experience*. Earthscan LLC (2005)
5. Gavanelli, M., Riguzzi, F., Milano, M., Cagnoli, P.: Constraint and optimization techniques for supporting policy making. *Computational Intelligent Data Analysis for Sustainable Development* (2012)
6. You, F., Tao, L., Graziano, D.J., Snyder, S.W.: Optimal design of sustainable cellulosic biofuel supply chains: Multiobjective optimization coupled with life cycle assessment and input - output analysis. *AIChE Journal* 58 (2012)
7. Lim, S.R., Suh, S., Kim, J.H., Park, H.S.: Urban water infrastructure optimization to reduce environmental impacts and costs. *Journal of Environmental Management* (2010)
8. Few, S.: *Now You See it: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press (2009)
9. Kohlhammer, J., Nazemi, K., Ruppert, T., Burkhardt, D.: Toward visualization in policy modeling. *IEEE Computer Graphics and Applications* 32 (2012)
10. Booshehri, M., Möller, T., Peterman, R.M., Munzner, T.: Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum* 31 (2012)
11. Heaps, C.: Long-range Energy Alternatives Planning (LEAP) system. Stockholm Environment Institute. Somerville, MA, USA (2012), <http://www.energycommunity.org>
12. Markovic, D., Cvetkovic, D., Masic, B.: Survey of software tools for energy efficiency in a community. *Renewable and Sustainable Energy Reviews* 15 (2011)
13. Matkovic, K., Gracanin, D., Jelovic, M., Ammer, A., Lez, A., Hauser, H.: Interactive visual analysis of multiple simulation runs using the simulation model view: Understanding and tuning of an electronic unit injector. Published in *IEEE Transactions on Visualization and Computer Graphics* 16 (2010)
14. Sadler, B.: Strategic Environmental Assessment at the Policy Level: Recent Progress, Current Status and Future Prospects. Ministry of the Environment, Czech Republic (2005)
15. Jones, C.: *Visualization and Optimization*. Operations Research/Computer Science Interfaces Series. Kluwer Acad. Publ. (1996)
16. Guillen-Gosalbez, G., Grossmann, I.: A global optimization strategy for the environmentally conscious design of chemical supply chains under uncertainty in the damage assessment model. *Computers & Chemical Engineering* 34 (2010)
17. Agarwal, R.K.: Assessment and optimization of an airplane's environmental impact", aircraft engineering and aerospace. *Technology* 85 (2013)
18. Ruppert, T., Bernard, J., Kohlhammer, J.: Bridging knowledge gaps in policy analysis with information visualization. In: Conf. on Electronic Government (to be published, 2013)
19. Harrower, M., Brewer, C.A.: Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal* 40 (2003)

Mesh Generation from Layered Depth Images Using Isosurface Raycasting

Steffen Frey, Filip Sadlo, and Thomas Ertl

Visualization Research Center, University of Stuttgart

Abstract. This paper presents an approach for the fast generation of meshes from Layered Depth Images (LDI), a representation that is independent of the underlying data structure and widely used in image-based rendering. LDIs can be quickly generated from high-quality, yet computationally expensive isosurface raycasters that are available for a wide range of different types of data. We propose a fast technique to extract meshes from one or several LDIs which can then be rendered for fast, yet high-quality analysis with comparatively low hardware requirements. To further improve quality, we also investigate mesh geometry merging and adaptive refinement, both for triangle and quad meshes. Quality and performance are evaluated using simulation data and analytic functions.

1 Introduction

Raycasting is available for a wide range of higher-order volumetric data representations, including classical polynomials [1] or radial basis functions [2] from smoothed-particle hydrodynamics, which are not defined on grids. Cell-based fields featuring piecewise polynomial representation resulting from higher-order finite element or discontinuous Galerkin simulations can also be visualized directly using raycasting [3]. Rendering this data interactively on a desktop computer can be impracticable due to storage and computational costs. Layered Depth Images (LDI) [4] of isosurfaces can be used as a replacement for the actual data to drastically lower hardware requirements, e.g., for preview rendering. An LDI is a view-dependent representation that stores several depth values per pixel and can easily be generated with slight modifications of existing raycasting code. However, common LDI rendering methods (warping or splatting) suffer from quality or performance issues in a number of scenarios, and many analysis operations (e.g., distance measurement) are not applicable. Our integrated technique allows for quick generation of LDIs and subsequently for fast extraction of a mesh from one or more LDIs to serve as a high-quality stub for interactive rendering (Sec. 3). These meshes can further be enhanced by surface-based refinement (Sec. 4) with advantages over traditional volume-based techniques (e.g., using octrees), and we also present a technique for the geometric merging of meshes from several LDIs (Sec. 5). In contrast, extracting meshes directly from the original data using common mesh-based isosurface extraction tools (like Marching Cubes [5], dual contouring [6], or others [7]) depends on the structure

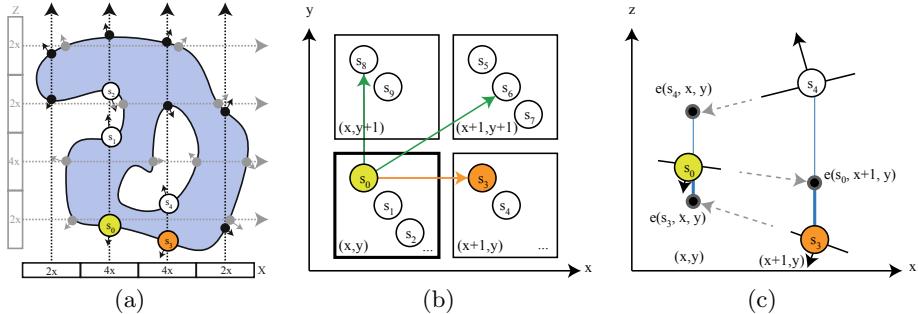


Fig. 1. (a) Generation of LDIs from different directions. (b) For each LDI independently, sample s_0 at (x, y) potentially forms a quad with samples from $(x+1, y)$, $(x, y+1)$, and $(x+1, y+1)$. (c) Normal-based depth prediction for s_0 to find its best match (in this case s_3), normals by black arrows.

of the data, and often cannot be applied directly. Further, the resampling of volumes either leads to inaccurate results or, as in the case of direct rendering, is very expensive in terms of computation time and memory. LDIs allow for the decoupling of mesh generation and the actual data representation. They provide a view-dependent region-of-interest selection that is represented in high quality. The compact representation of LDIs, their fast generation, as well as the quick extraction and rendering of meshes therefrom make our approach useful in many scenarios, e.g., for local or remote preview or in situ visualization.

2 Related Work

For isosurface extraction from higher-order data, quad mesh generation techniques [8], contouring [9], and approximate isocontouring [10] have been proposed. For uniform grids based on trilinear interpolation, classical Marching Cubes (MC) [5] and variants are most popular (e.g., dual MC [11] among others [12, 13]). MC adaptations were further introduced for tetrahedral meshes [14, 15], also supporting adaptive reconstruction [16, 17]. However, in contrast to our technique, they refine the volume and not the surface directly, and preventing cracks requires additional effort. Other approaches use Voronoi diagrams [18], advancing front techniques [19], and meshing from point clouds [20]. For combining disjoint meshes, as done with our approach (Sec. 5), several approaches have been proposed, including sewing [21], volumetric methods [22], zipping overlapping meshes [23], laser range images from different views [24], and polygon triangulation [25]. Various techniques have been presented for rendering cell-based higher-order fields [26], including a raytracer for cut-surfaces [27] and point-based visualization [28]. LDIs represent one camera view with multiple pixels along each line of sight [4], their size growing linearly with the observed depth complexity of the scene. In volume rendering, layer-based representations (like LDIs) have been used to defer lighting or transfer function changes (e.g., [29–31]).

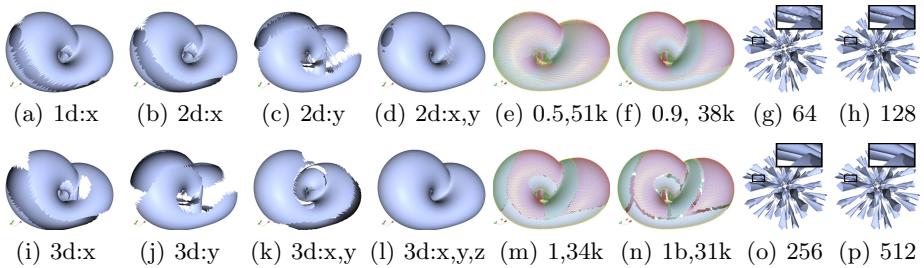


Fig. 2. (a)–(d),(i)–(l) Preview meshes of the KleinBottle data with a resolution of 128×128 per direction, showing the differences in trimming for the same $r = 0.5$ but a varying set of directions (subcaption depicts the total number of directions as well as the ones that are rendered). (e),(f),(m),(n) Varying r with meshes from x , y and z -direction colored in red, yellow and green respectively (subcaption depicts r (plus additional boundary trimming) and the number of triangles). (g),(h),(o),(p) Barth data set with $r = 0.5$ for x , y , and z -directions and an increasing number of samples per direction, resulting in higher quality with a higher number of triangles: $64^2 : 24k$, $128^2 : 136k$, $256^2 : 625k$, $512^2 : 2646k$.

3 Mesh Generation and Trimming

For LDI generation, we restrict ourselves to raycasting with parallel projection in the following for the sake of simplicity (perspective projection works accordingly with slight adjustments). Depth and gradient information are stored not only for the first, but for all hits occurring along a ray (Fig. 1(a)). This is the only required, typically straight-forward, modification to existing raycasting codes.

To form a quad, every sample (s_0 , located at (x, y)) selects one sample (its best match) each from the right $(x + 1, y)$, top $(x, y + 1)$ and the top right $(x + 1, y + 1)$ image position (Fig. 1(b)). The best match is determined by depth predictions based on normals. In detail, s_0 estimates the depth value $e(s_0, x + a, y + b)$ at the image position of the neighbor set ($a, b \in \{0, 1\}$, $(x + 1, y)$ in Fig. 1(c)). Likewise, all candidate samples s (s_3 and s_4 in Fig. 1(c)) estimate a depth value $e(s, x, y)$ at image position (x, y) . The best match for s_0 is then the sample s from the respective neighbor set with the smallest sum of distances $|e(s, x, y) - d(s_0)| + |e(d(s_0), a, b) - d(s)|$ where $d(\cdot)$ returns a sample's depth.

While using a single mesh can deliver a good approximation for slightly varying camera positions, meshes from multiple directions substantially improve the result for larger surface variations (Fig. 2). To determine the number of meshes (views) to use, independent from the actual data, we employ a quality measure based on the largest possible angle between the normal of a surface patch and the mesh direction, resulting in the following: 1 view: 90° , 2: 90° , 3: 58° , 4: 55° , 6: 49° , 6: 41° , 7: 39° , and 8: 37° . While the computation cost increases linearly with further directions, the quality of surface coverage does not. Accordingly, although our approach works with an arbitrary number of directions, we restrict ourselves to three as a reasonable trade-off in the following.

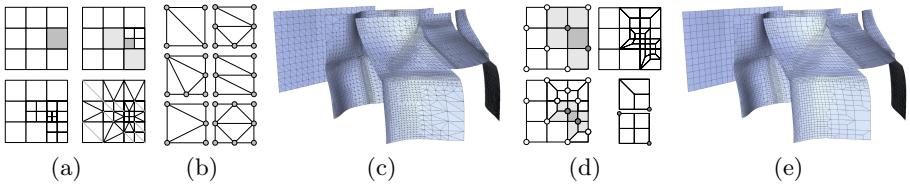


Fig. 3. Mesh refinement for triangles and quads demonstrated with the higher-order Shock Channel data [3] ($t = 3.0$, isovalue 4.6) showing the density of the flow around a box obstacle (c), (e). *Triangles:* (a) The subdivision of a cell (dark gray) forces further subdivisions (light gray) according to the 1-level difference rule. (b) Triangles are generated using six triangle templates (up to rotation and inversion). *Quads:* (d) Subdivision of a cell (dark gray) marks two respective edge nodes (dark circles) and thus cells (light gray). New edge nodes are introduced and previously marked edge nodes are removed. Another cell is chosen for subdivision (dark gray), respective edge nodes are marked and transition cells selected (light gray). Finally, quads are created using two templates.

Meshes from multiple directions cover some surface areas multiple times. Trimming them according to their quality of coverage—defined by means of the view direction and the surface normal (Sec. 3)—both avoids issues when the meshes are overlaid for rendering and creates distinguished boundaries for merging (Sec. 5). For every quadrilateral primitive $q \in Q_d$ from view direction $d \in D$, we compute the normal vector n of each of its four vertices $v \in q$ as the cross product with its two neighboring vertices: $n_v = (v - v_{prev}) \times (v - v_{next})$. Whether a vertex v is valid or not is determined by testing for $|n_v \cdot d_q| / \max(|n_v \cdot d|, \forall d \in D) > r$, with d_q being the view direction from which q was generated. The user-adjustable trimming parameter r specifies the extent of surface reduction with respect to its normal and view direction as well as all other view directions. The larger r , the more vertices are classified as invalid and the more primitives are eventually discarded (Fig. 2). The choice of r is application-dependent as discussed below. Finally, only quads remain that feature no invalid vertex. Overlaying meshes from different directions for rendering simply requires enabled depth testing. Optimally, r should thus be chosen such that there are neither low quality primitives occluding fine details, nor holes in the geometry.

4 Mesh Refinement

Templates are employed for refining the obtained meshes from Sec. 3 to better represent areas of high curvature without generating hanging vertices (i). This requires new samples for the LDI (ii). Refinement not only happens within meshes, but also at boundaries for growing the mesh toward silhouettes (iii).

(i) Refinement Templates. Templates differ for triangle and quad output. For triangles, a classical quadtree approach is used (Fig. 3) with the maximum subdivision level difference between neighboring cells being restricted to one for good quality triangles. After cells are marked for subdivision, an additional iterative *1-level difference* pass is employed that marks cells that have to be subdivided additionally to assure the constraint before generating triangles.

For quad mesh output the 1-level difference pass is substituted with the 2-refinement quad templates (Fig. 3(d)) [32] [33]. Two templates featuring “reference nodes” (dark circles in Fig. 3(d)) have to match so-called *edge nodes* inside the mesh. During subdivision, an edge node is created each time an edge is subdivided. For example, classical quadtree subdivision produces 5 new nodes, one in the center and one on each of the edges of the original cell. Ebeida et al. [33] propose to subdivide every cell of the initial mesh for producing an initial set of edge nodes. Instead, we identify edge nodes for the initial (unrefined) mesh from the pixel coordinates (x, y) of a vertex by testing if $x + y \bmod 2 = 0$, thus producing a “chessboard pattern” of initial edge nodes. First, edge nodes are identified that belong to cells marked for subdivision (*active edge nodes*, gray circles). Subsequently, all cells sharing such an active edge node are identified (*transition cells* marked by light gray quads) and templates are applied to all identified cells. Finally, previously active edge nodes are removed from the edge node set. A quad is marked for refinement, if any of its vertices is invalid, or the smallest dot product of a vertex normal with all neighboring vertex normals is below a certain value (0.99 proved successful in our experiments).

(ii) LDI Extension Sampling. Requests for additional LDI samples using the raycaster contain the direction and the new image position. All requests of one refinement pass are collected and processed at one go. Quad generation then works analogous to the initial mesh creation process (Sec. 3). If no new quad can be generated with the samples, an “invalid” vertex is used instead whose depth and normal are generated by interpolation from surrounding vertices. Invalid vertices are used for refining toward boundaries and silhouettes (iii). Primitives containing invalid vertices at the end of the refinement phase are removed.

(iii) Growing toward Boundaries and Silhouettes. The initial quads might not suffice as features can be missed that are smaller than the initial sampling distance. Thus boundaries and sharp tips of the isosurface might not be represented appropriately (Fig. 4). To refine toward these (Fig. 4(a)), a new top level quad is added if one valid vertex exists on an open edge of an existing top level quad (quad before subdivision) (Figs. 4(b)–(d)). Such a vertex represents a hanging node that is resolved by quad subdivision (Figs. 4(f)–(g)). Adding a top level quad instead of a “small” (already refined) quad allows efficient recognition of quads growing from different boundaries to a sharing edge, thus preventing mesh overlaps. Furthermore, the refinement templates with their associated 1-level-difference criterion or marked vertices can be handled more consistently.

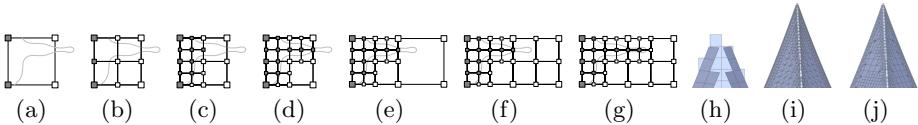


Fig. 4. Refinement at silhouettes. (a) Initial quad and true silhouette (gray curve). Cells containing true (gray) and invalid (white) samples are refined. (b) After two levels of refinement (c), the edge growing rule marks the right upper quad because it contains a valid sample on its left edge. (d) To resolve the hanging node at the right edge of the original quad, a new quad to its right is generated (e) and subdivided (f)–(g) to the respective level. Exemplary refinement of (h) toward edges (and corners) with triangles (i) and quads (j).

5 Combining Meshes

First, we trim overlapping parts (Sec. 3) using $r = 1$ (i.e., keeping only vertices whose normal best matches its original view direction) and then remove the “boundary” layer of cells featuring an edge with no neighbor (e.g., Fig. 2(n)). The resulting parts are combined by inserting so-called bridges (i) that connect close meshes. The remaining holes in the connected meshes are filled with triangles (ii) which can be quality-improved (iii), and finally converted to quads (iv).

(i) Bridge Generation. Bridges are quads with one edge e_a being connected to mesh a , one edge e_b connected to another mesh b , and two connecting open edges e_{ab} and e_{ba} . Bounding boxes are used to determine the distances of all mesh patch pairs a and b in order to identify the bridge to be inserted with the smallest edge lengths $|e_{ab}|$ and $|e_{ba}|$. The bridge is kept and thus the meshes are merged if neither $|e_{ab}|$ nor $|e_{ba}|$ exceed $l = |e_a| + |e_b|$. In our experiments, we stopped the search early when $|e_{ab}| + |e_{ba}| < l/2$ for faster results.

(ii) Hole Filling. Bridges reduce mesh combination to a hole filling problem. We employ an ear-cutting algorithm due to its simplicity and low computational complexity: iteratively the shortest possible edge is introduced that forms a triangle with two existing open edges until there are no open edges left. However, due to the intricacy of 3D meshing, these approaches are “heuristics” that typically cannot be proven to provide the correct result [25]. Nevertheless, refinement and trimming typically provide good-natured hole problems that are easy to fill.

(iii) Triangle Mesh Enhancement. First, all quads that are directly adjacent to hole-filling triangles are split into triangles (resulting in the sewing region Fig. 5(a)(i) (light)). Next, edge flips based on edge length are performed. 3-to-1 triangle merges then resolve configurations of three adjacent almost coplanar triangles, detected by a vanishing determinant of the spanned tetrahedron.

(iv) Quad Mesh Generation. The triangles introduced during hole filling and mesh enhancement (Fig. 5(a)(i)) can be converted to quads. The Catmull-Clark algorithm would generate a high primitive count by introducing three quads per triangle. Instead, we iteratively combine the pair of triangles that leads to the

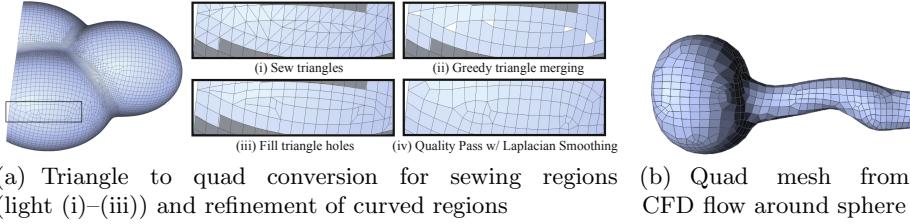


Fig. 5. Merged and refined quadrilateral isosurface meshes from three LDIs

Table 1. Timing results in seconds for different data sets and steps of our approach (if executed) on a single core of an Intel Core i7 with 3.4 GHz

Fig.	Mesh.	Ref.	Bridge	Sew	Qual.	Tot.
Klein Bottle [34] (128^2)						
2	0.04	—	—	—	—	0.04
Barth [34] ($64^2, 128^2, 256^2, 512^2$)						
2(g)	0.02	—	—	—	—	0.02
2(h)	0.11	—	—	—	—	0.11
2(o)	0.43	—	—	—	—	0.43
2(p)	1.84	—	—	—	—	1.84
Marschner-Lobb [35] ($64^2, 512^2$)						
6(g)	0.08	—	0.11	0.04	0.23	0.46
6(h)	0.08	0.73	0.62	0.1	0.79	2.32
6(i)	7.19	—	0.82	3.38	22.07	33.46

Fig.	Mesh.	Ref.	Bridge	Sew	Qual.	Tot.
Coulomb [34] 64^2						
5(a)	< 0.01	< 0.01	0.01	0.01	1.84	1.86
Sphere [3] 64^2						
5(b)	< 0.01	—	< 0.01	0.01	0.40	0.40
Shock Channel [3] 64^2						
3	< 0.01	0.01	—	—	—	0.01
Slices $16^2, 24^2, 32^2$ $x^2 + y^2 + z^2 + \sin(5x + 15y + 6z) - 1$						
6(j)	< 0.01	—	< 0.01	< 0.01	< 0.01	< 0.01
6(k)	< 0.01	—	0.01	0.01	< 0.01	0.02
6(l)	0.02	—	0.02	0.01	0.04	0.09

best quad according to a simple quality metric (ratio of minimum to maximum edge length) (Fig. 5(a)(ii)). Typically a small number of unmerged triangles remains. Then, we find the triangle pair with the shortest connecting path using Dijkstra’s algorithm. This path is edge-connected and contains the triangle pair and the quads in between. Subsequently, we go from one triangle to the other, splitting traversed edges by introducing edge vertices (Fig. 5(a)(iii)). Passing a quad straight splits the quad in two, while passing adjacent edges splits the quad in three (using the quad refinement template from Fig. 3(d)). We repeat from identifying the shortest connecting path until all triangles are split into quads and finally merge quads sharing two edges, improve the vertex valence via quad edge flips toward four and apply Laplacian smoothing (Fig. 5(a)(iv)).

6 Results

For evaluation, we use the higher-order unstructured grid raycaster by Üffinger et al. [3] using CUDA, and the implicit surface raycaster by Knoll et al. [34] implemented in Cg. All data sets as well as timings for the presented results throughout the paper can be found in Tab. 1. Timings do not include raycasting times to generate the underlying LDI. Results were obtained using three viewing directions along the x , y , and z -axis unless otherwise noted. Fig. 2(e), (f), and (m) show that the larger r , the larger the mutually covered regions, reducing the risk of holes, but also potentially leading to invalid coverings. In our experience, $r = 0.5$ provides a good trade-off overall. The timings (Tab. 1) also

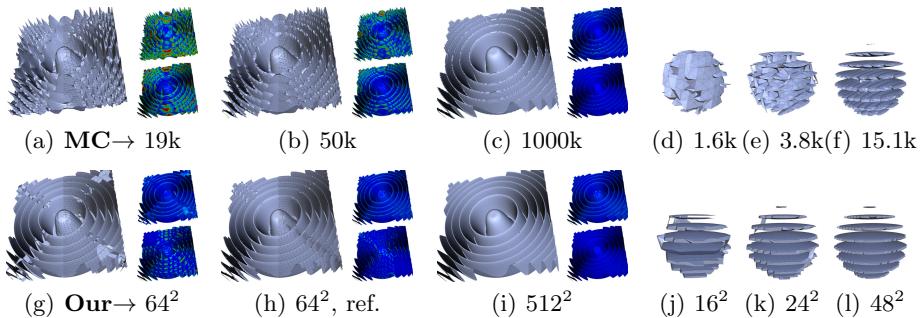


Fig. 6. Comparison of our approach (bottom row with LDI resolution, “ref.” denotes refinement) to MC (top row with triangle count) using the Marschner-Lobb signal and the Slices data set. Vertical image pairs have approximately the same triangle count. For Marschner-Lobb, additionally forward and backward geometrical distances are depicted (top and bottom right) using a rainbow color map from 0 to 0.005 and 0.03, respectively (domain extent is 0.64 per direction).

suggest that r could be interactively adjusted to best fit the data set and the requirements of the user. Fig. 2(n) shows $r = 1$ with the additional boundary trimming prior to sewing. Preview meshes generated from different LDI resolutions (Fig. 2(g),(h),(o), and (p)) provide more details for higher resolutions with a better coverage of thin features and strong curvature with an approximately linear relation between primitive numbers and generation time. Even the generation of high-resolution meshes at interactive rates would be possible, considering the large improvement potential through parallelization. Only one LDI can already suffice depending on the nature of the isosurface (Fig. 3). Fig. 3 also demonstrates refinement to the surface structure for triangles and quads. Although the original mesh of a test function (Fig. 4) is very coarse (Fig. 4(h)), mesh growing still allows to nicely adapt to edges and corners, both for triangles and quads (Fig. 4(i) and (j)).

In Fig. 6, we demonstrate the accuracy of our approach using geometric distances in comparison to meshes from MC [36] with an approximately equal triangle count. We compare all vertices of the candidate mesh to a reference (MC with 5M triangles) (forward) and all vertices of the reference to the candidate (backward). In contrast to MC with a similar triangle count, our meshes deliver close to perfect results with forward comparison, and are also consistently better for backward comparison (particularly at signal peaks in comparison to the jagged coverage of MC). Much better results than MC (Fig. 6(a)) are also achieved for very low resolutions, despite some artifacts due to insufficient sampling (Fig. 6(g), incorrect bridges, and hole filling as both steps rely on sampling distance for finding correct matches). Our approach preserves the basic structure of data well, even if topologically incorrect bridges between the slices are introduced (Fig. 6(d)–(f), (j)–(l)). Refinement leads to smaller gaps between mesh patches that belong together topologically and thus reduces artifacts

(Fig. 6(h)). However, although significantly decreasing, small holes in the peaks of the signal in boundary regions even persist with fine sampling (Fig. 6(i)) as connections across the peak are shorter than along the peak. This could be fixed by advanced bridging and hole filling heuristics considering normal variation in addition to distance. Overall, from low to high resolution, our approach was able to generate a more detailed approximation for similar triangle counts.

7 Conclusion

We presented and evaluated a novel approach for view-dependent mesh extraction from LDIs independent of the underlying data structure. We demonstrated its usefulness in the context of existing raycasting-based techniques, providing fast generation of adaptively refined triangle and quad meshes, both for the purpose of preview rendering and further analysis. As a consequence, however, no topological guarantees can be made in contrast to commonly used isosurface extraction techniques (e.g., [5] [6]). For future work, we plan to work on a data-dependent LDI view selection and to conduct a more in-depth evaluation of our technique in comparison to other meshing techniques beyond classic MC, e.g., by using topology verification techniques [37].

References

1. Knoll, A.M., Wald, I., Hansen, C.D.: Coherent multiresolution isosurface ray tracing. *Vis. Comput.* 25, 209–225 (2009)
2. Gamito, M.N., Maddock, S.C.: Ray casting implicit fractal surfaces with reduced affine arithmetic. *Vis. Comput.* 23, 155–165 (2007)
3. Üffinger, M., Frey, S., Ertl, T.: Interactive high-quality visualization of higher-order finite elements. *Comput. Graph. Forum* 29, 337–346 (2010)
4. Shade, J., Gortler, S., He, L.W., Szeliski, R.: Layered depth images. In: SIGGRAPH, pp. 231–242 (1998)
5. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3D surface construction algorithm. *Comput. Graph.* 21, 163–169 (1987)
6. Schaefer, S., Warren, J.: Dual marching cubes: primal contouring of dual grids. In: *Comput. Graph. Forum*, pp. 70–76 (2004)
7. Fryazinov, O., Pasko, A., Comninou, P.: Fast reliable interrogation of procedurally defined implicit surfaces using extended revised affine arithmetic. *Comput. Graph.* 34, 708–718 (2010)
8. Remacle, J.F., Henrotte, F., Baudouin, T., Geuzaine, C., Béchet, E., Mouton, T., Marchandise, E.: A frontal delaunay quad mesh generator using the l^∞ norm. In: 20th Meshing Roundtable, pp. 455–472 (2012)
9. Wiley, D.F., Childs, H.R., Gregorski, B.F., Hamann, B., Joy, K.I.: Contouring curved quadratic elements. In: VisSym, pp. 167–176 (2003)
10. Pagot, C.A., Vollrath, J., Sadlo, F., Weiskopf, D., Ertl, T., Comba, J.: Interactive isocontouring of high-order surfaces. In: Scientific Visualization: Interactions, Features, Metaphors, vol. 2, pp. 276–291 (2011)
11. Nielson, G.M.: Dual marching cubes. In: IEEE Vis., pp. 489–496 (2004)

12. Dietrich, C., Scheidegger, C., Schreiner, J., Comba, J., Nedel, L., Silva, C.: Edge transformations for improving mesh quality of marching cubes. *Trans. Visual. Comput. Graphics* 15, 150–159 (2009)
13. Bommes, D., Lévy, B., Pietroni, N., Puppo, E., Silva, C., Tarini, M., Zorin, D.: Quad meshing. In: Eurographics, pp. 159–182 (2012)
14. Zhou, Y., Chen, B., Kaufman, A.: Multiresolution tetrahedral framework for visualizing regular volume data. In: IEEE Vis., pp. 135–142 (1997)
15. Anderson, J., Bennett, J., Joy, K.: Marching diamonds for unstructured meshes. In: IEEE Vis. 2005, pp. 423–429 (2005)
16. Grosso, R., Ertl, T.: Progressive iso-surface extraction from hierarchical 3d meshes. *Comput. Graph. Forum* 17, 125–135 (1998)
17. Westermann, R., Kobbelt, L., Ertl, T.: Real-time exploration of regular volume data by adaptive reconstruction of iso-surfaces. *Vis. Comput.* 15, 100–111 (1999)
18. Dey, T., Levine, J.: Delaunay meshing of isosurfaces. In: Shape Modeling and Applications, pp. 241–250 (2007)
19. Schreiner, J., Scheidegger, C.E., Silva, C.T.: High-quality extraction of isosurfaces from regular and irregular grids. *Trans. Visual. Comput. Graphics* 12, 1205–1212 (2006)
20. Scheidegger, C.E., Fleishman, S., Silva, C.T.: Triangulating point set surfaces with bounded error. In: EG Symposium on Geometry Processing, pp. 63–72 (2005)
21. Kobbelt, L.P., Botsch, M.: An interactive approach to point cloud triangulation. In: Eurographics (2000)
22. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH, pp. 303–312 (1996)
23. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: SIGGRAPH, pp. 311–318 (1994)
24. Rocchini, C., Cignoni, P., Ganovelli, F., Montani, C., Pingi, P., Scopigno, R.: The marching intersections algorithm for merging range images. *Vis. Comput.* 20, 149–164 (2004)
25. Held, M.: Fist: Fast industrial-strength triangulation of polygons. Technical report, Algorithmica (2000)
26. Sadlo, F., Üffinger, M., Pagot, C., Osmari, D., Comba, J., Ertl, T., Munz, C.D., Weiskopf, D.: Visualization of cell-based higher-order fields. *Computing in Science and Engineering* 13, 84–91 (2011)
27. Nelson, B., Kirby, R.M., Haimes, R.: GPU-Based Interactive Cut-Surface Extraction From High-Order Finite Element Fields. *Trans. Visual. Comput. Graphics* 17, 1803–1811 (2011)
28. Rosenthal, P., Linsen, L.: Direct isosurface extraction from scattered volume data. In: EuroVis, pp. 99–106 (2006)
29. Ropinski, T., Prassni, J., Steinicke, F., Hinrichs, K.: Stroke-based transfer function design. In: SPBG, pp. 41–48 (2008)
30. LaMar, E., Pascucci, V.: A multi-layered image cache for scientific visualization. In: PVG, pp. 61–68 (2003)
31. Tikhonova, A., Correa, C., Ma, K.L.: Explorable images for visualizing volume data. In: PacificVis, pp. 177–184 (2010)
32. Schneiders, R.: Refining quadrilateral and hexahedral element meshes. In: 5th International Conference on Grid Generation in Computational Field Simulations, pp. 679–688 (1996)
33. Ebeida, M.S., Patney, A., Owens, J.D., Mestreau, E.: Isotropic conforming refinement of quadrilateral and hexahedral meshes using two-refinement templates. *International Journal for Numerical Methods in Engineering* 88, 974–985 (2011)

34. Knoll, A., Hijazi, Y., Kensler, A., Schott, M., Hansen, C.D., Hagen, H.: Fast ray tracing of arbitrary implicit surfaces with interval and affine arithmetic. *Comput. Graph. Forum* 28, 26–40 (2009)
35. Marschner, S.R., Lobb, R.J.: An evaluation of reconstruction filters for volume rendering. In: *IEEE Vis.*, pp. 100–107 (1994)
36. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: Measuring error on simplified surfaces. *Comput. Graph. Forum* 17, 167–174 (1998)
37. Etiene, T., Nonato, L.G., Scheidegger, C., Tierny, J., Peters, T.J., Pascucci, V., Kirby, R.M., Silva, C.T.: Topology verification for isosurface extraction. *Trans. Visual. Comput. Graphics* 18, 952–965 (2012)

FractVis: Visualizing Microseismic Events

Ahmed E. Mostafa¹, Sheelagh Carpendale¹, Emilio Vital Brazil¹, David Eaton², Ehud Sharlin¹, and Mario Costa Sousa¹

¹ Department of Computer Science,

² Department of Geoscience,

University of Calgary

Abstract. We present our efforts of applying information visualization techniques to the domain of microseismic monitoring. Microseismic monitoring is a crucial process for a number of tasks related to oil and gas reservoir development, e.g., optimizing hydraulic fracturing operations and heavy-oil stimulation. Microseismic data has many challenging features including high dimensionality and uncertainty. We present a brief introduction to the domain of microseismic monitoring, and derive a set of tasks and data abstractions that can establish common ground between microseismic monitoring domain experts and visualization researchers. We then present FractVis, a prototype for visual analysis of microseismic data, describing the ongoing process of iteratively refining FractVis through close collaboration and consultation with domain experts. FractVis is designed to offer microseismic monitoring experts with visual analytic tools that allow investigation of the 3D spatial distribution of microseismic events, time-varying analysis and interactive exploration of high-dimensional parameter spaces, extensively complementing the existing tools in their disposal.

1 Introduction

The increasing global demand for energy motivates the oil/gas industry to invest in tools that can help domain experts make better-informed decisions. Recently microseismic monitoring has emerged as one of the most important processes to support such decisions. However, making informed decisions about improving reservoir modeling based upon microseismic data is a challenge for expert analysts. These difficulties arise due to the inherent features of the microseismic data: intrinsic complexity, high-dimensionality, and a high degree of uncertainty. Currently these difficulties are intensified by a lack of visual analytic tools to support interactive visual interpretation of the dataset. To address these difficulties, domain experts are demanding efficient interactive visualization tools that can help them as they explore their data.

Microseismic data is comprised of events, each representing an extremely small earthquake [1]. These events are the result of fractures created and/or activated to allow oil and gas trapped in rock pores to flow more easily. The fracture information is captured by sensors (e.g. geophones) and structured as continuous raw ground-motion records. Following, the raw data is pre-processed resulting in an event catalog containing tabular information with many attributes per event. The data inherits high abstraction and uncertainty from the measurements and the preprocessing [2, 3]. Once gathered

and processed, microseismic data is generally analyzed by several domain experts such as geophysicists, geologists and reservoir engineers, each representing a different skill set, and often having different interests. The analysis consists of several tasks; expert interpreters need to know the locations of the events in relation to the wells in the reservoir, to be able to filter out noisy events, and perform correlations between various attributes within the large, high-dimensional microseismic dataset. Some important high-level tasks performed by the experts include: understanding hydraulic fracture geometry, estimating the stimulated reservoir volume (SRV), and optimizing long-term field development [1]. These tasks could benefit dramatically from an interactive visualization tool that converts the microseismic data into efficient and effective visual representations. Such a tool should be designed to better reflect and express the available information, the level of uncertainty and other pertinent data details from different stages of oil/gas exploration and production.

The primary contribution of our work is the characterization of the microseismic domain challenge, outlining the potential benefit of applying information/scientific visualization techniques to this problem domain, and sharing our insight based on the design and evaluation of our current implementation FractVis. We describe the data exploration tasks involved in microseismic monitoring, and the common domain abstractions in order to highlight and share our insights of the domain challenges and needs. From these we derive our prototype design requirements, encoding choices and interaction techniques. The secondary contribution of our work is the design, development and preliminary evaluation of FractVis; an interactive visualization prototype that enable exploration and analysis of microseismic events. FractVis is being developed and refined iteratively with feedback and consultations from domain experts. FractVis combines and extends existing and novel visualization techniques to help experts to explore their data and make informed decisions. We conclude with our reflections and lessons we have learned during the design of FractVis.

2 Related Work

Many visual analytic systems and visualization techniques applied to reservoir geoscience and engineering have been developed through the recent years [4–7]. Although these tools assist people in their decision making process, there is still lack of visual analytic systems of geophysical data in the microseismic domain. The majority of the work in the domain of microseismic engineering and geosciences has been in the area of developing mathematical methods for microseismic monitoring [1, 2]. Limited research has been done in the area of microseismic visualization and many of the microseismic scientific papers use commercial tools that lack the support of visual interpretation and analysis of microseismic data. In this section, we summarize some of the key related works that have inspired our implementation.

A scatterplot matrix [8] can visually represent multidimensional data by creating a matrix of N^2 scatterplots arranged in N rows and N columns. However, the resolution of each scatterplot in the scatterplot matrix is limited when the data contains high dimensionality. Elmqvist et al. proposed a starplot-like system titled DataMeadow [9], which is a visual canvas designed to support analysis of large-scale multivariate data with flexible visual queries.

One important component of our visualization tool involves the use of parallel coordinates (PC) [10]. This is a well-known technique for visualizing highly dimensional data that represents every dimension as vertical axis parallel to other dimensions on a 2D plane. Heinrich and Weiskopf [11] presents a survey of the current state of the art of visualization techniques for parallel coordinates. While we know that PC suffers from data cluttering, we employed some strategies to alleviate this problem including brushing [12] and axis ordering [13]. PC have also been applied in many different visual analytic systems. Steed et al. presented a system for analyzing weather data using an enhanced PC's implementation [14]. Other visualization techniques have been combined with PC to provide better visualizations tools. For example, Yuan [15] presented a system that scatters the data points within PC. Martin et al. [12] discussed high dimensional brushing for exploring multivariate data with focus on PC. These brushing methods have been integrated in XmdvTool [16] which is a system that combines many multivariate visualization methods. Also, evaluation of PC [17] have shown that the people who performed the tasks with PC found them more effective than other methods.

Roberts [18] provided a discussion of the state of art on using coordinated multiple views, and discussed many systems that support this technique. Similarly to other systems (e.g. Bowman et al. [19]) that provide coordination of different representations of the data, we also make use of multiple coordinated views. Wang-Baldonado et al. [20] provided a set of guidelines for using multiple views in information visualization while Andrienko et al. [21] provided a critical examination of multiple coordinated views.

3 Microseismic Characterization

Microseismic monitoring offers unique information visualization challenges and potential. In this section we briefly characterize the microseismic monitoring domain to motivate our own design, and in hope that this characterization would allow future information visualization efforts to address the various domain challenges. We describe the typical structure of microseismic monitoring datasets, and highlight its important attributes. We present the data abstractions experts are using when approaching the datasets and the high-level tasks they are pursuing, along with the processes and the challenges they are facing. The raw data we present was gathered during continuous meetings, contextual inquiries [22] sessions, and semi-structured interviews with domain experts.

Microseismic Background. Hydraulic fractures are created by injecting water or other specially developed fluids such as cross-linked gels into the rock formation. The injection is performed under high pressure through a chamber in the well causing the formation to crack or fracture, thus generating micro-earthquakes (also called microseismic events). A multi-stage hydraulic fracturing is created by multi-chamber, illustrated by spheres with different colors in Fig. 1. These multi-stage hydraulic fracturing techniques are designed to expose a larger amount of drainage area to the wellbore as compared to a single-stage fracture. Receiver systems (i.e. geophones) are placed in locations near the (fracturing) process to detect the energy generated by the events, and then providing geometric information.

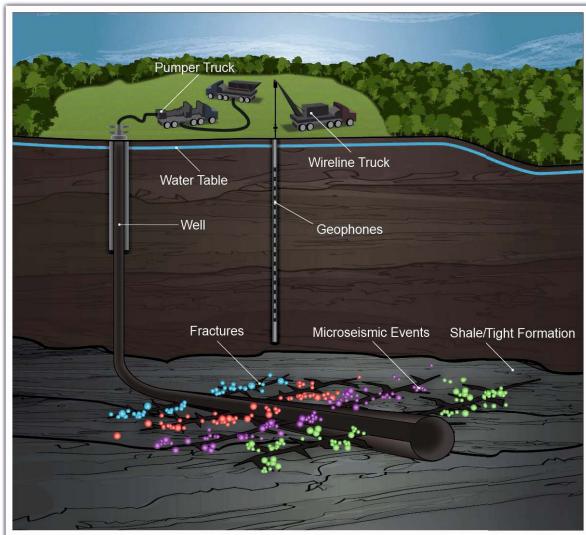


Fig. 1. Hydraulic fracture schematic overview showing multi-stage fracturing (four different colors for the spheres of each microseismic events stage) along with a single well [23].

Microseismic events' locations are calculated using a velocity model [2]. The velocity model describes the propagation of elastic waves (P and S) from the fracture to the detection system. Specifically, this model incorporates the acoustic wave-speed and thickness of rock layers between the source and the receiver locations. Combined with limited acquisition geometry, the uncertainty is inherent in this model and within the calculated locations of the microseismic events. In addition recorded microseismic events typically have noise associated with them, and this may come from many sources including even a truck moving on the surface. Thus the microseismic data events, in addition to their ambiguity, also contain noise and inaccuracies that make them highly uncertain.

Data Description. Microseismic dataset is composed of many layers, but in our work we focus on three because they fit our exploratory goals. The first layer is the microseismic "Events Catalog" which describes each event along with its attributes. The second layer is the "Monitoring and Treatment" wells information. A third component comprises the engineering data and pumping curves. All of these data layers usually exist within a single dataset.

The microseismic data employed in our design of FractVis was a highly multidimensional time varying point cloud dataset. According to experts, some of the most important microseismic attributes are **Time**, **Location**, **Distance**, **Magnitude**, **Noise-Level**, and **Energy**. They also expressed that some of these attributes are independent while others are dependent on each other. For example, the attributes *distance* and *magnitude* are independent and are usually used as standard test for initially checking the validity of the data. In contrast, the attribute *noise level* is dependent on the *signal-to-noise ratio* attribute.

The engineering data layer represents the different characteristic of the fracture growth and the events population with time. For example, pumping curves provide correlation between time and pressure in the injection process. By examining these real-time plots, experts can confirm that microseismic events start to be generated when the pressure reaches its peak with the fluid injection causing the rock to break or fracture. Visualizing the engineering data layer curves and linking them with 3D visualization of the events is important for better understanding of the fracture geometry. The current version of FractVis does not address this second layer of the dataset and we are planning to integrate it in our future prototypes.

Task Analysis and Challenges. Microseismic experts perform different tasks while exploring the data. First, domain experts mentioned that estimating the stimulated reservoir volume (SRV) is one of the most common tasks in microseismic engineering. The goal of this task is to generate a bounding volume which defines subsets of the data events as initial estimations of the production volume. The locations of the events are important in this calculation. However, these locations are estimated due to the inherit uncertainty of the measurements. Experts consider the inclusion of uncertainty in SRV calculation an important future challenge [3]. Various methods are applied to the events prior to calculating the SRV in an attempt to filter out the unimportant events and analyze the data attributes [24]. The ability to filter the data and make decisions regarding the events is greatly affected by the insight and understanding of the dataset; and the expert's ability to extract relations among its attributes. Additionally, experts are analyzing dataset outliers manually. They believe that this manual component can benefit greatly from applying interactive or semi-automatic interactive visualization data correlation techniques. Secondly, since the microseismic data is a time-varying point-cloud, there is a room for supporting time-based visualization and analysis. Microseismic experts consider the *time* attribute as one of the most important independent variables. They expressed that it is common to analyze the correlation between the *time* and many other attributes. Thirdly, analyzing fracture growth over time (i.e. measurements of fracture azimuth, width, etc.) could be spatially visualized to obtain an understanding of fracture geometry and the fractures' interactions, understanding that can be crucial when analyzing the dataset. Finally, domain experts expressed that the ability to see the data from different perspectives at the same time is important. For instance, the synchronized: visualization of the 3D events, visualization of the attributes, and the visualization of the engineering curves would be useful if represented intuitively.

Attempting to analyze the microseismic monitoring dataset involves several challenges. First, although some of the data attributes have dependency, the dimensionality of the independent data attributes is still quite high. Potential techniques for reducing this high dimensionality will certainly aid in the analysis of the data. Second, the data inherently contains uncertainty due to the inaccurate measurements and the noise associated with them. Noise in the data comes from many sources and can not be completely removed. In fact, many techniques have been attempted recently in order to reduce the noise, but the processed data still contains noise which can be quantified for each event [3]. Finally, the microseismic data is highly abstract. The data could have different interpretations and it can often be difficult to validate which of them is the most accurate one. Experts explained that some of the attributes may have different meanings

in different contexts, and that applying domain insight is still a crucial part of the process. Overall, the domain experts we consulted thought that visual-analytic tools would be very effective in helping them interactively and effectively explore the dataset. For instance, domain experts said that they sometimes do not fully understand the relationship between many of these attributes and were hoping to be able to intuitively spatially correlate various data attributes in order to learn more about the potential effect of each of them.

4 Design Rationale

We adopted an iterative design approach, we built our first prototype, and we modified our system iteratively based on the requirements and the feedback of our domain collaborator. We decided to focus on supporting the simple tasks of "data filtering" and "attributes correlation". We analyzed the high-level tasks of "data filtering" and "attributes correlation" then we identified the following concrete tasks: *Find Anomalies, Associate, Correlate, Identify, Filter, and Categorize*; by following the taxonomy of [25]. As a result, we designed our prototype to support these tasks.

We chose to represent every microseismic event as sphere centered at its 3D spatial location with radius proportional to any of the attribute values. The color of any sphere event and its correspondent PC line is defined by correlating a color map with some attribute. Among the color maps that FractVis supported, we also supported a rainbow (jet), which may not be recommended for usage in visualization systems [26], but domain experts are familiar with this color map. Our domain collaborators acknowledged our choices of mapping the radius/color of each event sphere relative to some attributes. They considered this mapping to be natural to them, powerful for showing much information at once, and comparable to many existing commercial tools.

Why Parallel Coordinates? First, the technique of PC supports exploration of data trends and attributes correlation without affecting the scale and the dimensionality of the data, which is not the case for the other projective and non-projective techniques. Second, PC is a widely used technique and supports extensibility. Indeed, we extended the PC by integrating dynamic magic lenses and embedding them with it. Furthermore, experts can dynamically recolor the content of the PC according to some attribute to examine attributes correlation without the need to reorder the axes of the PC. Third, the study performed by Siirtola and Räihä [17] revealed that who performed their tasks with PC found it more effective than those who used other methods. Finally, we think that if we extended our visualization and provided interactively embedded visuals (e.g. scatter plot) within our PC, then it would be easier for the experts to familiarize themselves with it and learn interacting with it quickly, interaction which would empower the experts with rich visuals without the need to show additional visualization windows.

5 FractVis

Our implementation follows the multiple coordinated views approach [18]. We considered this approach because, we think, it is important to have different representations

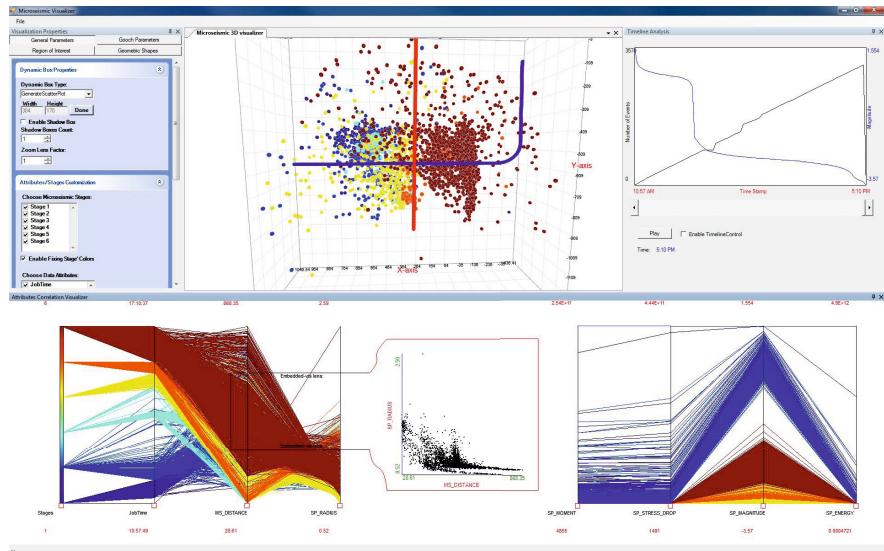


Fig. 2. System overview showing the synchronization of the PC view (bottom) with the other data visualization components: (top center) 3D microseismic events' point cloud, (top right) the time-based visualization and (top left) the GUI view for controlling the visualization parameters.

of the data, at the same time, for achieving simultaneous data analysis. Our system, FractVis, supports two primary coordinated views (Figure 2). The main 3D view enables visual exploration of the microseismic events in the reservoir space with well integration. The second view supports flexible interaction and correlation through an improved PC visualization. Each view presents the data in a different way, allowing experts to link and relate the meanings gained from one view with the others.

The technique of PC [10] can be used to visually explore the main trends and/or relations of a multidimensional data. The standard PC consists of n -parallel lines typically vertical and equally spaced, where ' n ' is the number of dimensions (attributes) of the data. Each data sample is represented as a polyline intersecting each attribute at the corresponding relative value. In fact, we extended the PC by introducing and integrating two novel extensions. The first extension describes the integration of magic lenses over the PC to enable intuitive interactions such as data filtering and scaling. In the second extension we present our idea of visual correlation through the use of visual legends.

We extended the implementation of PC through the concept of dynamic boxes (similar to magic lenses [27]) blended over the PC plot. Once a dynamic box is created, all the visible attributes' axes that intersect this dynamic box will be considered for achieving the corresponding effect. We support two types of dynamic boxes where each of them is being represented using different color and shape to utilize the cognitive power of the users and facilitate interactions. The first dynamic box causes data filtering (Figure 3 top). Such a filter box will constrain the events to only those who fall within its limits (range), similar to data brushing [12]. Additionally, our visualization shows the

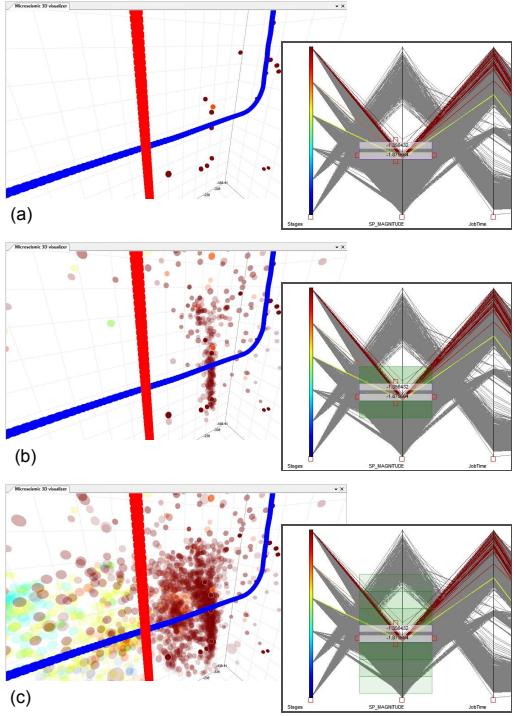


Fig. 3. The effect of filtering over the PC with and without keeping the context: (a) show the normal filtering with a single filter box, (b) show the effect of two shadow boxes and how they result in a small partial context, and (c) show the effect of activating six shadow boxes to increase the partial context.

filtered out events as transparent 3D spheres and/or gray polylines within the PC to make it easier to identify them. The user can create many filter boxes to achieve complex filtering. This idea is similar to iterative brushing [9] where composite filters are created in order to focus on a refined subset of the data. The second type is a dynamic box which causes embedding custom visuals within the PC visualization. For instance, the user could embed scatter plot within the PC similar to the work of Holten and van Wijk [28]. However, in our implementation, we support such integration interactively. Figure 2 (bottom) shows a dynamic box that caused a scatter plot visual to be generated (and embedded) within the PC.

Shadow boxes (Figure 3) are other novel visual elements that can be attached with filter boxes to enable: (1) range/cluster navigation; by gradually fading all the events before and after the range of the current active filter box, and (2) partial contextualization. This feature is inspired by the work of Doleisch and Hauser [29], where the authors used smooth brushing to reflect the smooth nature of features in their flow simulation data. The number of shadow boxes as well as their properties can be controlled through the GUI of FractVis. For example, in Figure 3 (b and c), the effect of shadow boxes is shown. We can see that although we are strictly filtering the data events (using our filter box), the (synchronized) 3D view shows other (transparent) events representing the partial context.

We introduce a new forms of interactive-based correlation through the concept of "visual legends". The basic idea is about placing visual maps (i.e. color map) over any attribute's axis to update the data representation relative to this attribute. This idea is similar to "gradient color brush" introduced by Matkovic et al. [30] but we extended this idea by allowing it to represent different visual variables such as color and size. In our implementation, we support two visual legends (maps) in order to perform color correlation and/or size correlation. First, a color map can be placed over any attribute to (associate and) enforce (re)coloring all the PC's polylines, as well as the 3D spheres, according to the distribution of the values of the selected attribute. Second, a size map can be placed over any attribute to (re)size all spheres of the 3D events accordingly. This could help in identifying the spatial geometric location of events relative to the well. Furthermore, it can be also useful for analyzing the 3D location of the possible events outliers and confirm if they are outliers or not. Our implementation also supports the feature of axes reordering to analyze the relation between any non-sequenced attributes, but we believe that our dynamic legend-based correlation (for instance using color) can be useful for quickly identifying such relations without the need to reorder the attributes.

6 Discussion and Lessons Learned

Given the relatively recent emergence of microseismic monitoring methods, the number of domain experts is limited. Clearly, having access to a limited number of domain experts may not be suitable for conducting detailed formal evaluation, but it does suggest other benefits. The repeating sessions with the same experts allowed for continuous and coherent feedback and refinement of the prototype. Having repeated access to the same experts allowed us to confirm that the system features meet their expectations. We conducted informal evaluation by demoing our visualization prototype to the domain experts and also to visualization researchers. The goal was to gather their reaction about our prototype.

Most of our participants provided positive feedback about many of the system's features. One of the highly experienced domain experts discovered a very interesting issue with the data calculation using our visualization. He analyzed the relation: *Magnitude* vs. *Distance*, and he specifically expressed: "*When I look at this, I can see there is a problem with the data ... because it is not physically feasible ... So this just highlights some problem with the data*". Another feedback that shows some limitations and weaknesses in our tool has been provided by some of the participants as well. One domain expert expressed her opinion about our feature of having embedded visuals within the PC as confusing. She specifically expressed: "*I like it popped up in the middle, but what it did is just disconnected the way I am looking into the data so I have to go back*". We also received many suggestions for improving our tool. For instance, one domain expert suggested that integrating additional types of data (i.e. engineering curves) would be important.

During some of the assessment sessions with the domain experts, they commented about having different visualization and interaction possibilities in our system. Some of them considered that to be confusing and they just preferred simple visualization, while others considered it to be a form of flexibility. Regarding the PC, one expert expressed:

"The parallel coordinates is very unique, and you've just showed me that it can be more powerful... when I become a good user with it, it will be tremendously useful". On the other hand, when we asked an expert participant about the idea of having multiple dynamic (filter) boxes, and whether it is easier or not, she specifically said: *"That would be something that I have to use for some time to know if it would be easier or not, but for now I think the concept is useful. I think it can be a very good idea"*. These comments suggest that our prototype may be a good start for microseismic visual-analysis, though a detailed and formal evaluation is needed to fully confirm that feedback and guide future development.

Generally, throughout the process we felt that domain experts are resisting considering and learning new tools and new ways of thinking about analyzing their datasets. While we understood this reaction, it was one of the main challenges that we were facing. Indeed, it inspired us to think about simplifying our design in order to provide experts with simpler tool that will provide new insight while still feeling familiar. One such experience had to do with introducing PC to them as new visualization tool. Our experts were not familiar with PC, and they seem to resist understanding or using it in our early sessions with them. Following this initial resistance we provided the experts with additional visualizations which were more familiar to them, such as scatterplot, integrated with the PC visualization. Our approach was that embedding the new visualization side-by-side with familiar ones would allow users to explore it while retaining a known baseline context, allowing them to learn the new technique. The feedback that we received (from most of the participants) confirmed that our approach was useful and helpful. Overall, we wanted to empower the PC visualization by adding the flexibility to see additional (embedded) visuals which would lead to enhancing the data analysis experience.

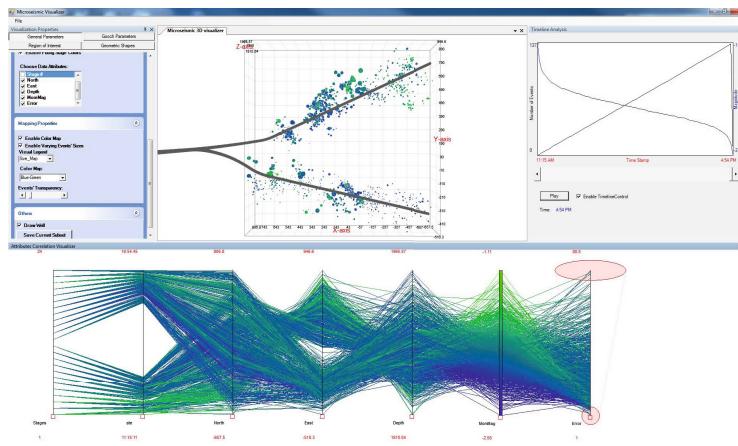


Fig. 4. Visualizing another microseismic dataset using FractVis. The 3D visualization shows that the events from well A (top) are systematically higher than those from well B (bottom).

7 Conclusion and Future Work

In this work, we detail a characterization of the microseismic domain including data abstraction and description. Based on that, we also explain a set of design requirements and visual representation choices specific to the development of microseismic visualization. We developed a prototype, FractVis, for visual exploration of microseismic data. FractVis is composed of a set of coordinated visualizations that resulted by combining and extending different techniques through an iterative collaborative process with the domain experts. Our implementation is flexible and can adapt to any new microseismic data file. Indeed, we visualized another microseismic dataset using our system (Figure 4) and initial insight has been found.

Since it is an ongoing project, there are many improvements to follow. As future work, we are considering the suggestions provided from the feedback regarding improving the prototype and adding additional important features. Furthermore, we plan to conduct a formal detailed user study in the near future. We are planning to conduct ethnographic sessions with the microseismic domain experts to refine our understanding of their processes and practices.

Acknowledgments. We thank the anonymous reviewers for the constructive comments and feedback. We also thank ESG Solutions for Figure 1. We extend our thanks to the Microseismic Industry Consortium (Geoscience, U. of Calgary) and ConocoPhillips for the data set. This research was supported by the NSERC/AITF/Foundation CMG IRC in Scalable Reservoir Visualization and by the AITF/NSERC/SMART IRC in Interactive Technologies.

References

1. Norm Warpinski, P.: Microseismic monitoring: Inside and out. *Journal of Petroleum Tech.* 61, 80–85 (2009)
2. Daku, B., Salt, J., Sha, L.: An algorithm for locating microseismic events. In: CCECE 2004, vol. 4, pp. 2311–2314 (2004)
3. Ulrich, Z.: Calculating stimulated reservoir volume (srV) with consideration of uncertainties in microseismic-event locations. In: CURC 2011. SPE International (2011)
4. Höllt, T., Beyer, J., Gschwantner, F., Muigg, P., Doleisch, H., Heinemann, G., Hadwiger, M.: Interactive seismic interpretation with piecewise global energy minimization. In: PacificVis 2011, pp. 59–66 (2011)
5. Patel, D., Bruckner, S., Viola, I., Groller, E.: Seismic volume visualization for horizon extraction. In: PacificVis 2010, pp. 73–80 (2010)
6. Dopkin, D., James, H.: Trends in visualization for e&p operations. *First Break* 24 (2006)
7. Rusby, R.I.: The future of visualization: Vision 2020. *WorldOil* 229 (2008)
8. Elmqvist, N., Dragicevic, P., Fekete, J.D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *TVCG* 14, 1148–1539 (2008)
9. Elmqvist, N., Stasko, J., Tsigas, P.: Datameadow: A visual canvas for analysis of large-scale multivariate data. In: VAST 2007, pp. 187–194 (2007)
10. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: VIS 1990, pp. 361–378. IEEE (1990)

11. Heinrich, J., Weiskopf, D.: State of the art of parallel coordinates. In: Eurographics Association (ed.) STAR Proceedings of Eurographics 2013, pp. 95–116 (2013)
12. Martin, A.R., Ward, M.O.: High dimensional brushing for interactive exploration of multivariate data. In: VIS 1995, pp. 271–278. IEEE (1995)
13. Peng, W., Ward, M.O., Rundensteiner, E.A.: Clutter reduction in multi-dimensional data visualization using dimension reordering. In: INFOVIS 2004, pp. 89–96. IEEE (2004)
14. Steed, C., Swan, J., Jankun-Kelly, T., Fitzpatrick, P.: Guided analysis of hurricane trends using statistical processes integrated with interactive parallel coordinates. In: VAST 2009, pp. 19–26 (2009)
15. Yuan, X., Guo, P., Xiao, H., Zhou, H., Qu, H.: Scattering points in parallel coordinates. TVCG 15, 1001–1008 (2009)
16. Ward, M.: Xmdvtool: integrating multiple methods for visualizing multivariate data. In: Visualization 1994, pp. 326–333 (1994)
17. Siirtola, H., Räihä, K.J.: Discussion: Interacting with parallel coordinates. Interact. Comp. 18, 1278–1309 (2006)
18. Roberts, J.: State of the art: Coordinated multiple views in exploratory visualization. In: CMV 2007, pp. 61–71 (2007)
19. Bowman, I., Joshi, S., Van Horn, J.: Query-based coordinated multiple views with feature similarity space for visual analysis of mri repositories. In: VAST 2011, pp. 267–268 (2011)
20. Wang Baldonado, M.Q., Woodruff, A., Kuchinsky, A.: Guidelines for using multiple views in information visualization. In: AVI 2000, New York, pp. 110–119 (2000)
21. Andrienko, G., Andrienko, N.: Coordinated multiple views: a critical view. In: CMV 2007, pp. 72–74 (2007)
22. Holtzblatt, K., Jones, S.: Contextual inquiry: a participatory technique for system design, pp. 177–210. Lawrence Erlbaum Associates, Hillsdale (1993)
23. ESG Solutions hydraulic fracture mapping,
<https://www.esgsolutions.com/english/view.asp?x=741>
(accessed: March 31, 2012)
24. Amorim, R., Boroumand, N., Vital Brazil, E., Hajizadeh, Y., Eaton, D., Costa Sousa, M.: Interactive sketch-based estimation of stimulated volume in unconventional reservoirs using microseismic data. In: Proceedings of 13th European Conference on the Mathematics of Oil Recovery, ECMOR XIII (2012)
25. Amar, R., Eagan, J., Stasko, J.: Low-level components of analytic activity in information visualization. In: INFOVIS 2005, pp. 111–117. IEEE (2005)
26. Borland, D., Taylor, R.: Rainbow color map (still) considered harmful. IEEE Comp. Graph. and Appl. 27, 14–17 (2007)
27. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and magic lenses: the see-through interface. In: SIGGRAPH 1993, pp. 73–80 (1993)
28. Holten, D., Van Wijk, J.J.: Evaluation of cluster identification performance for different pcp variants. Computer Graphics Forum 29, 793–802 (2010)
29. Doleisch, H., Hauser, H.: Smooth brushing for focus+context visualization of simulation data in 3d. Journal of WSCG, 147–154 (2001)
30. Matkovic, K., Jelovic, M., Juric, J., Konyha, Z., Gracanin, D.: Interactive visual analysis and exploration of injection systems simulations. In: VIS 2005, pp. 391–398 (2005)

Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm

Gwenael Bothorel^{1,2}, Mathieu Serrurier², and Christophe Hurter^{2,3}

¹ DSNA/DTI, avenue du Docteur Maurice Grynfogel, 31100 Toulouse, France
<http://www.developpement-durable.gouv.fr/-navigation-aerienne-.html>

² IRIT, 118, route de Narbonne, 31062 Toulouse Cedex 9, France
<http://www.irit.fr>

³ ENAC, 7 avenue Edouard Belin, BP 54005, 31055 Toulouse Cedex 4, France
<http://www.enac.fr>

Abstract. Frequent itemset mining is one of the major data mining issues. Once generated by algorithms, the itemsets can be automatically processed, for instance to extract association rules. They can also be explored with visual tools, in order to analyze the emerging patterns. Graphical itemsets representation is a convenient way to obtain an overview of the global interaction structure. However, when the complexity of the database increases, the network may become unreadable. In this paper, we propose to display itemsets on concentric circles, each one being organized to lower the intricacy of the graph through an optimization process. Thanks to a graph bundling algorithm, we finally obtain a compact representation of a large set of itemsets that is easier to exploit. Colors accumulation and interaction operators facilitate the exploration of the new bundle graph and to illustrate how much an itemset is supported by the data.

Keywords: Data Mining, frequent itemsets, graph visualization, bundling, optimization.

1 Introduction

Frequent itemsets mining aims at finding links between data, which may not be easily detected. Among the varied fields of Data Mining, it is one of the most studied, because it is a key elements in mining pattern. This is why along two decades, it has been the subject of many studies and publications. For instance, in 1994, Agrawal and Srikant have presented Apriori algorithm which leans on frequent itemsets calculations [1]. Later, other algorithms and methods have been studied, like CAP [2] or DHP [3]. The frequent itemsets can be processed in a next step of automatic calculation, for instance in order to find associations rules. Representing and exploring data mining results is a challenging issue since the number of itemsets or rules can be very large when the complexity of the database increases. The association rules are displayed in systems like AViz, which is an interactive visualization system for discovering numerical association rules from

large data sets [4], or ARVis which shows the rules and associated measures values in a 3D information landscape [5]. But before visualizing association rules, it is also interesting to visualize the frequent itemsets. FIsViz proposes such an approach [6]. It displays the frequent itemsets in a 2D space, by linking the items thanks to connecting edges. Such a tool gives a global graph of the dataset. It illustrates that it is necessary to use a graph when the purpose is the representation of itemsets. Indeed we have to show at the same time the data and the links between the different elements. Moreover, an element is generally involved in many itemsets that can have different sizes. So a graph layout is much adapted to show frequent itemsets. Thus, trying to find frequent itemsets patterns with such a presentation, is trying to find patterns in graph mining. This issue has been recently studied in [7]. It shows that graph mining has become an active and important theme in data mining. One problem is the complexity of the representation, considering the tremendous quantity of connections between the nodes.

As the number of edges can be very large in a graph representation of data, in recent years, graph bundling methods have gained increased attention. They stem from studies on confluent drawing by reducing non-planar graphs to planar ones [8]. The purpose was to allow groups of edges to be merged together and drawn as 'tracks'. Bundling starts with a set of nodes positions, given as input data or computed by a layout algorithm. Edges being close in terms of graph structure, position, data attributes or combinations, are drawn as tightly bundled curves. This trades clutter for overdraw and produces images which are easier to understand and/or better emphasize the graph structure. Blending or shading can be used to add information or emphasize structure [9–11]. Bundling algorithms exist for both compound (hierarchy-and-association) [12] and general graphs [9, 13, 14]. However attractive, many bundling algorithms for general graphs are relatively complex and have high computational costs. A recent study has proposed a faster method based on density maps using kernel density estimations. It relies on graphic cards acceleration techniques [15].

In this paper, we propose to apply a bundling method to a new graph representation of itemsets that takes advantage of their properties (Itemsets are described in Section 2). Itemsets are disposed on nested circles, each one corresponding to the number of items in the itemsets. Then an algorithm reorganizes the itemsets in order to have relevant proximities of the nodes. Finally the bundling algorithm is applied. As the proximity of the connections is a key factor indicating that they correspond to frequent itemsets, the bundling simplifies the clutter that becomes more readable (Section 3). Moreover, Infoviz techniques are used to enhance the visualization, particularly by using color and transparency accumulation, and interaction operators help the user to explore the bundle layout (Section 4).

2 Frequent Itemsets

Data mining consists in extracting knowledge from a vast volume of data. One purpose is to find relevant patterns that are underlined by this data. A database

is a set of vectors $\langle a_{i1}, a_{i2}, \dots, a_{im} \rangle$, also known as tuples, over an attributes space $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$. The attributes are qualitative (string, category, enumeration) or quantitative items which can be discretized. We denote an item as a value of an attribute (e.g. a_{i2}). An itemset or k -itemset is a subset of values in the attributes space. k is the number of items concerned by the itemset: $k \in \{1, 2, \dots, m\}$.

The database used to illustrate the paper is the Mushroom data set from the UC Irvine Machine Learning Repository¹. It includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms. The database consists of 8416 tuples of 23 discrete attributes about the edibility, the cap shape, the odor, the ring, the habitat, etc. For instance the cap shape attribute may have curve, flat, bell values, and the odor may have almond, anise, none values. An example of 2-itemsets is $\{\text{capshape} = \text{flat}, \text{odor} = \text{anise}\}$.

Itemsets are mainly characterized by the support which indicates the frequency they appear in the database (i.e. k -itemset with a frequency equals to 0.3 appears in 30% of the tuples in the database). A frequent itemset is an itemset which has a support greater than a threshold. Obviously, the support of a k -itemset is greater or equal than the support of a $(k+1)$ -itemset that contains this k -itemset. This $(k+1)$ -itemset is then named a *superset* of the k -itemset. So, as the value of k is increased, the support is decreased. As explained in [1], the k -frequent itemsets are iteratively built thanks to the combinations of two $(k-1)$ -frequent itemsets that differ only on a single item. For instance, the 2-itemset $\{a, b\}$ is the combination of the 1-itemsets $\{a\}$ and $\{b\}$. The 4-itemset $\{a, b, c, d\}$ is the combination of the 3-itemsets $\{a, b, c\}$ and $\{a, b, d\}$, but it is also the combination of the 3-itemsets $\{a, b, d\}$ and $\{b, c, d\}$, or $\{a, b, d\}$ and $\{a, c, d\}$, etc. This type of approach ensures that all the k -frequent itemsets will be extracted given a support threshold and a maximal value of k .

One major challenge of data mining is the exploration and the analysis of the whole set of frequent itemsets which is usually very large.

3 Circular Graph Layout for Frequent Itemsets

Usually, frequent itemsets presentations are linear layouts as shown in Figure 1. This type of presentation becomes quickly unreadable as the number of nodes and frequent itemsets is increased, and the layout is horizontally stretched to show the whole graph. Moreover, to keep the unicity of each node in order not to clutter up the graph, several frequent itemsets are usually connected to the same nodes. Taking into account the construction of the itemsets, we propose a circular presentation shown in Figure 2. We build our representation in three steps:

1. Structure of the graph
2. Optimization of the itemsets positions
3. Graph bundling

¹ <http://archive.ics.uci.edu/ml>

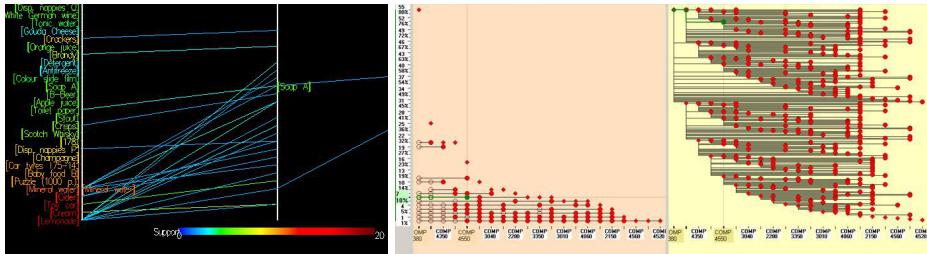


Fig. 1. Examples of frequent itemsets linear layout. From [16] (left) and WiFiIsViz [17] (right)

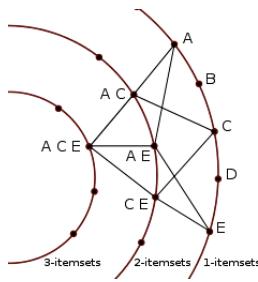


Fig. 2. Circular graph presentation of frequent itemsets: construction principle. $\{A, C, E\}$ is built from $\{A, C\}$ and $\{A, E\}$, which are built from $\{A\}$ and $\{C\}$ and from $\{A\}$ and $\{E\}$. $\{A, C, E\}$ is also built from $\{A, E\}$ and $\{C, E\}$.

Structure of the Graph. The graph is built with concentric circles, separated from the same distance, each one corresponding to the cardinal of the frequent itemsets. The 1-itemsets are on the external circle, the 2-itemsets on the next smaller circle, and so on until the smallest circle corresponding to the frequent itemsets with the highest cardinal. On this circular graph layout, a node is a frequent itemset and the segments are the links between the frequent itemsets of two consecutive circles. An itemset is the combination of two previous itemsets. Note that it can stem from several pairs of previous itemsets (itemset $\{A, C, E\}$ in Figure 2 for instance). In this case, all the combinations are represented. The distance between the nodes of a same circle is calculated in order to have an homogeneous repartition. Figure 4 shows examples of circular graphs.

Graph layouts with many nodes and edges have always to face the problem of readability due to the cluttering. A way to improve a circular graph layout, with one circle, has been studied in [18].

Optimization. In our study, to enhance the readability of the graph layout, we reorganize the itemsets on the circles, by minimizing the sum s of the segments length. In order to reduce s , we use a simple hill climbing algorithm which finds a local minimum. Two random itemsets are swapped on a random circle. If s is

decreased then this permutation is kept, otherwise it is cancelled. This operation is done a large number of times in order to find a minimum value of s .

Graph Bundling. Graph bundling algorithms aims at simplifying a graph by merging close edges in order to obtain main tracks. We have chosen Kernel Density Estimation-based Edge Bundling [15] for its simplicity and speed. It is simple because it requires only an input graph with nodes positions. It is efficient because GPU processing allows to make it parallelizable and much faster than comparable methods. Considering that each edge links two itemsets of two consecutive circles, the begin and end points must not be moved. Thus the bundling algorithm must be applied to the partial graphs between two consecutive circles. So, with N circles, it must be applied $N - 1$ times. The final layout is the combination of the partial bundlings (See Figure 3).

Our representation has many advantages:

- With the optimization, there is a coherence between the itemsets proprieties and their positions. Indeed if the itemsets are linked then they are more likely to be close. Otherwise, they are further.
- Considering the way to build an itemset from previous ones, the combination complexity should always grow as the value of k increases, and become unreadable on a small circle. However, the constraints of the support limit this effect. Indeed the number of itemsets generally increases with the first values of k , and then decreases quickly. In Figure 4(a), there are 11 1-itemsets, then the number of itemsets increases and finally decreases to 8 5-itemsets.
- The bundle layout is a way to simplify a complex graph. The optimization ordering the itemsets, the bundling groups edges that have common itemsets upstream or downstream. Thus the graph is more clear and easier to exploit compared to the original graph. The graph layouts of Figure 4(b) are completely unreadable, as the bundle graph gives information about the itemsets. Indeed with the latter it is easier to discriminate the edges leading to itemsets or leaving them. The whole process of generating the visualization takes less than two minutes.
- The bundle representation is a way to detect many itemsets proprieties. Indeed it is easier to notice if an itemset has many connections or not, to have an idea of the relations between itemsets. Thus areas where there are many connections correspond to areas where there is much information. This is illustrated in Figures 4 and 5.

4 Itemset Visualization Enhancement

Having built the itemsets circles, given a support threshold, we propose to enhance the visualization by three ways:

1. Alpha assignment
2. Color accumulation
3. Itemsets selection

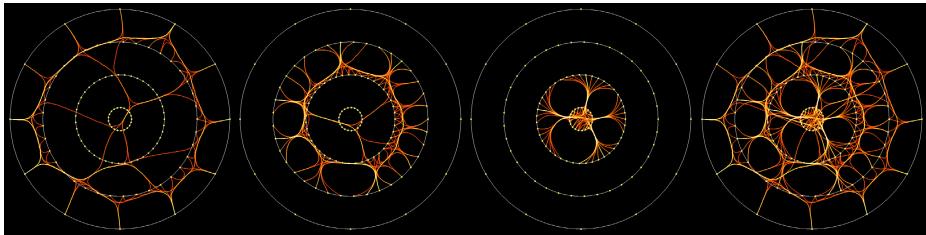


Fig. 3. The final bundling is the result of successive partial bundlings between two consecutive circles. Example with four circles.

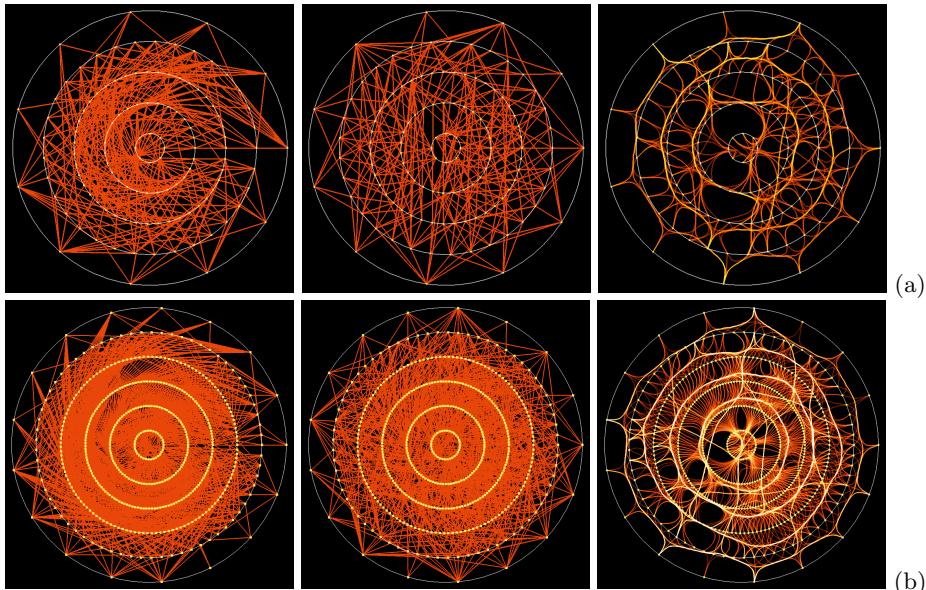


Fig. 4. (a) From left to right: original graph showing frequent itemsets (support=0.5 - number of segments=440 - segments length sum=112). Then the graph is optimized (segments length sum=72). Finally it is bundled. (b) The same sequence with a support equals to 0.45 and 6 circles. There are 1776 segments and the segments length sums are respectively 443 and 303.

Alpha Assignment. In order to emphasize the most relevant itemsets (i.e. the ones with greater support), we propose to use the support as the alpha transparency value. As the support concerns only the itemsets, that is to say the begin and end points of each edge, the value of the transparency gradients on the edges corresponds to the interpolation of the supports between these two points. It allows to have a continuous gradient of transparency, and that facilitates the layout overview.

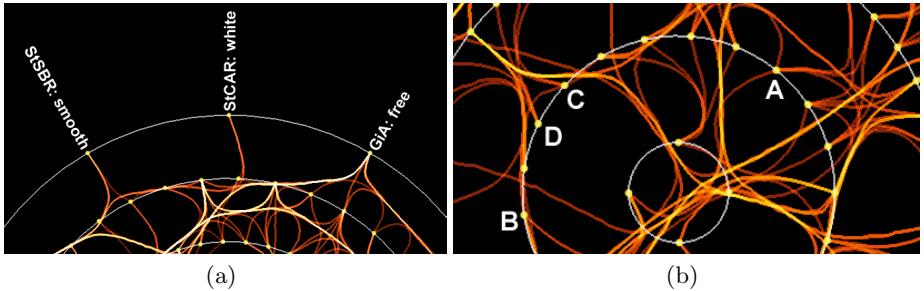


Fig. 5. (a) The accumulation of the color and the transparency is stronger on the right 1-itemset than on the others. It corresponds to a higher support. (b) Points C and D are maximal itemsets, whereas A and B are not.

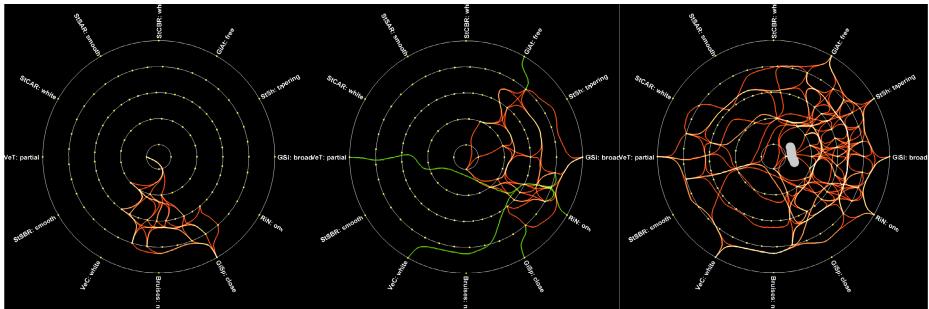


Fig. 6. Different types of selections. Left: as a 1-itemset is selected, the propagation shows which itemsets stem from this itemset. Middle: the selection of a 1-itemset is disseminated to the supersets, and from these 2-itemsets to the 1-itemsets. It shows which attributes share the same itemsets than a chosen attribute. Right: brush selection of edges. It shows which itemsets are affected by this selection.

Color Accumulation. Another way to get information about the itemsets is to use accumulation. This technique is a consequence of edge bundling, because the bundling distorts the original segments to get bent overlaid edges. So, the more segments are involved in a bundle edge, the more accumulation there is, and it enhances the values of the color and the transparency. As a result the value of the support has a direct effect on the visibility and the color of the edges. Thus, a itemset with a high support appears with a high color level and a high opacity. This feature makes it easier to enhance the most itemsets that are involved in the database, and to reduce the others.

Figure 5(a) illustrates these points. The RGBA value of the color is (1.00, 0.31, 0.21, 0.66). The right itemset corresponds to the attribute *Gill attachment* = *free*. It appears 8200 times in the database (support=97.4%), and has 11 supersets. As these values are large, the resulting accumulation of green and blue reaches 1.0, and it is the same for alpha. So the resulting color of the edges leaving this 1-itemset is (1.0, 1.0, 1.0, 1.0), which corresponds to white color

without transparency. Moreover, the edge leaving leftward is white, whereas the edge leaving rightward is yellow. It indicates the area of the second circle where the 2-itemsets are more concerned by this 1-itemset, and where they are less concerned. The left itemset of Figure 5(a) corresponds to the attribute *Stalk surface below ring = smooth*. It appears 5076 times in the database (support=60.3%), and has 4 supersets. The visual aspect of the edges leaving this itemset shows that its support is lower and that it has fewer supersets than the right itemset. Indeed, the green component is quite low and the blue value is lower. So the resulting edge is orange, with some transparency. In Figure 5(b) we can easily remark that C and D have no supersets. This kind of remarkable itemset is known as maximal itemset. Such an itemset is especially valuable in data mining process. On the contrary, we can immediately deduce from the visualization that A and B are not maximal itemsets since they have at least one superset.

Itemsets Selection. In order to focus on one or more itemset we propose a selection tool. Selecting an itemset shows only its backward and forward connections, thanks to a propagation effect, while the other connections are hidden. Thus selecting the itemset *A* on Figure 2 highlights its connections to *AC* and *AE*, and from *AC* and *AE* to *ACE*. Selecting *AC* highlights its connections to *A*, *C* and *ACE*, and selecting *ACE* highlights its connections to *AC*, *AE*, *CE*, and then to *A*, *C* and *E*. Figure 6 shows an examples of this selection. On the left picture, the selection of the itemset *Gill spacing=close* (support = 81.1%) shows that it has five supersets on level 2. The propagation to the next levels gives seven, four and finally one itemset. Note that, as the algorithm (see Section 3) gathers the itemsets in the same areas, they are not spread on the graph.

It is also interesting to know for instance which 1-itemsets share the same 2-itemsets than the 1-itemset that is selected. A backward feature proposes to highlight differently these 1-itemsets, with a simple green color coding. So, by selecting a 1-itemset, there is a propagation to the supersets, and then to the other 1-itemsets that are previous itemsets of these supersets. As a consequence, when a 1-itemset is selected, it is easy to detect the other 1-itemsets that are linked to it. Thus it is easy to represent which attributes are related with a selected attribute. It is also possible to do it for any circle. The middle picture of Figure 6 illustrates this concept. We can see that the attributes that share the same supersets than *Gill size=broad* are *Gill attachment=free*, *Veil color=white* and *Veil type=partial*.

A second type of selection is the edges selection. By brushing an edge, it selects every edges that are grouped with it thanks to the bundling. This selection highlights all the connections forward and backward linked to the selected edges. It is a way to have a quick overview of the itemsets that are linked thanks to these edges. The right picture of Figure 6 shows an example of such a selection.

Finally, by considering the color properties of the edges, the proposed visualization gives a fast and relevant view of large set of itemsets. In addition to the advantages pointed out in the previous section, we enhance the visualization with the following properties:

- Identification of the relevant itemset. The itemsets with higher support are emphasized if the alpha value is associated with it. Moreover, the evolution of the transparency of an edge, from a larger circle to the lower one, gives information about the evolution of the support when the number of items increases.
- Relevant areas of the circular graph. The accumulation of color, together with the optimization of the position, automatically emphasizes the areas where there is relevant information. It corresponds to areas where itemsets have a high support and are heavily linked.
- Itemsets hierarchy identification. With selection, it is possible to focus on an itemset or a group of itemsets, by enhancing the other itemsets that are linked.

5 Conclusion

In this paper, we propose a new visualization of frequent itemsets based on a multi-circular graph which competes the state of the art visualizations in this domain. The position of the itemsets is optimized in order to improve the quality of the visualization while respecting and emphasizing the properties of the itemsets. Then a Kernel Density Estimation-based Edge Bundling is applied. The result is a more simple graph of the frequent itemsets that shows the main streams in the layout, even when the number of itemsets is high. It enhances the most involved itemsets and their links. In other words it shows the most supported attributes in the database and how they are combined. We have proposed selection operators that can be used to focus on itemsets and on the way they take place in the graph. Thanks to color coding and accumulation, the importance of each itemset is highlighted or reduced. We have illustrated the effectiveness of our approach on the mushroom database from UCI Learning Repository.

In a future work, we plan to improve the optimization by using more efficient algorithms and allowing the itemsets to be placed more freely on the circles without keeping a constant distance between them. Moreover a view of the association rules stemming from enhanced or selected itemsets should be useful. Finally an evaluation will assess our visualization to verify that it is a good approach.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) VLDB 1994, Proc. of 20th Int. Conf. on Very Large Data Bases, Chile, pp. 487–499. Morgan Kaufmann (1994)
2. Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. SIGMOD Rec. 27, 13–24 (1998)
3. Park, J.S., Chen, M.S., Yu, P.S.: Using a hash-based method with transaction trimming for mining association rules. IEEE Trans. on Knowl. and Data Eng. 9, 813–825 (1997)

4. Han, J., Cercone, N.: Aviz: A visualization system for discovering numeric association rules. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 269–280. Springer, Heidelberg (2000)
5. Blanchard, J., Guillet, F., Briand, H.: A user-driven and quality-oriented visualization for mining association rules. In: Proc. of the Third IEEE Int. Conf. on Data Mining, ICDM 2003, pp. 493–496. IEEE Computer Society, Washington (2003)
6. Leung, C.K.-S., Irani, P.P., Carmichael, C.L.: FIsViz: A frequent itemset visualizer. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 644–652. Springer, Heidelberg (2008)
7. Singh, V., Garg, D.: Survey of finding frequent patterns in graph mining: Algorithms and techniques. *Int. J. of Soft Computing and Engineering* 1, 19–23 (2011)
8. Dickerson, M., Eppstein, D., Goodrich, M.T., Meng, J.Y.: Confluent drawings: Visualizing non-planar diagrams in a planar way. In: Liotta, G. (ed.) GD 2003. LNCS, vol. 2912, pp. 1–12. Springer, Heidelberg (2004)
9. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. *Comput. Graph. Forum* 28, 983–990 (2009)
10. Lambert, A., Bourqui, R., Auber, D.: Winding roads: Routing edges into bundles. *Comput. Graph. Forum* 29, 853–862 (2010)
11. Telea, A., Ersoy, O.: Image-based edge bundles: simplified visualization of large graphs. In: Proc. of the 12th Eurographics/IEEE - VGTC Conference on Visualization, EuroVis 2010, Aire-la-Ville, Switzerland, pp. 843–852. Eurographics Association (2010)
12. Holten, D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics* 12, 741–748 (2006)
13. Cui, W., Zhou, H., Qu, H., Wong, P.C., Li, X.: Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 1277–1284 (2008)
14. Gansner, E.R., Hu, Y., North, S., Scheidegger, C.: Multilevel agglomerative edge bundling for visualizing large graphs. In: Proc. of the 2011 IEEE Pacific Visualization Symposium, PacificVis 2011, USA, pp. 187–194. IEEE Computer Society (2011)
15. Hurter, C., Ersoy, O., Telea, A.: Graph bundling by kernel density estimation. *Comp. Graph. Forum* 31, 865–874 (2012)
16. Yang, L.: Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In: Kumar, V., Gavrilova, M.L., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003, Part I. LNCS, vol. 2667, pp. 21–30. Springer, Heidelberg (2003)
17. Leung, C.K.S., Irani, P.P., Carmichael, C.L.: Wifisviz: Effective visualization of frequent itemsets. In: Proc. of the 2008 Eighth IEEE Int. Conf. on Data Mining, ICDM 2008, USA, pp. 875–880. IEEE Computer Society (2008)
18. Gansner, E.R., Koren, Y.: Improved circular layouts. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 386–398. Springer, Heidelberg (2007)

Automatically Extracting Hairstyles from 2D Images

Chuan-Kai Yang and Chia-Ning Kuo

National Taiwan University of Science and Technology, Taipei, 106, Taiwan

Abstract. Automatic hair extraction from a given 2D image has been a challenging problem for a long time, especially when complex backgrounds and a wide variety of hairstyles are involved. This paper has made its contribution in the following three aspects. First, it proposes a novel framework that successfully combines the techniques of *face detection*, *outlier-aware initial stroke placement* and *matting* to extract the desired hairstyle from an input image. Second, it introduces an *alpha space* to facilitate the choice of matting parameters. Third, it defines a new comparison metric that is well suited for the *alpha matte* comparison. Our results show that, compared with the manually drawn *trimaps* for hair extraction, the proposed automatic algorithm can achieve about 86.2% extraction accuracy.

1 Introduction

Extracting the hairstyle from an input image has always been a challenging problem due to numerous intervening factors such as lighting, shadow, skin colors, backgrounds, clothing, and so on, especially when many of these factors can present large variety or complexity. For example, it is possible that both the environment background and the subject's clothing share similar color or textures to the hair, and thus a precise separation of the hairstyle from the rest may be very difficult.

In this paper, we have proposed a novel framework that combines *face detection*, *outlier-aware initial stroke placement*, and *matting* to make automatic hairstyle extraction possible. First, we make use of the existing *active shape model* technique for identifying the facial contour and facial features from the input image. Second, by making use of the detected positions of the facial features, our system could place an initial stroke so that most of the it would lie within the target hairstyle. A *dynamic programming* approach is applied to further adjust the position of the stroke to increase the degree of alignment of the stroke with the desired style. Finally, given the previously generated stroke, a well known *closed-form matting* is applied to derive a segmentation of the image so that the hairstyle is roughly separated from the background. Such a segmentation is used to generate a *trimap* so that *closed-form matting* is applied again to more accurately extract the hairstyle as the foreground.

In addition to proposing a new framework to automatically extracting desired hairstyle from a given image, we have also made two more contributions as

follows. First, we propose a more systematic way to remove undesired eyebrows and neck during the matting process. Second, we propose a *similarity metric* that could efficiently and intuitively compare two *alpha matting* results, and we believe such a mechanism can also be generalized to other comparisons as well. Based on numerous experiments, our system can achieve 86.2% hair extraction accuracy.

The rest of the paper is organized as follows. Section 2 briefs existing works which are related to this work. Section 3 details our core algorithm on extracting the desired hairstyle from an input image. Section 4 evaluates our results, while Section 5 concludes this work.

2 Related Work

There are at least several techniques involved in this study. The first one is *face detection*. Viola et al. [1] adopted a *machine learning* approach, together with the well known technique of *AdaBoost*. The final output is a rectangular window to mark the detected face. *Active shape model*, or *ASM* for short, proposed by Cootes et al. [2] is also a method based on machine learning; however, it represents a detected object through a collection of feature points instead of a rectangle. Cootes et al. [3] later on proposed another approach called *active appearance model*, or *AAM* for short, which is based on machine learning, and uses training data to obtain a statistical appearance model.

The second one is *outlier detection* [4], which is to find *outliers* from given data. Some of the approaches could be further classified into *distance-based*, *clustering-based* and *spatial-based* [5]. In particular, the definition of a distance-based outlier is as follows: if β percentage of the data are at least r distance away from o , then o is an outlier.

The third related technique is *matting*. The problem of a matting is to separate the foreground from the background of a given image. In general, such a problem can be formulated as the following:

$$I_p = \alpha_p F_p + (1 - \alpha_p) B_p \quad (1)$$

where p is a pixel (coordinate), I is the input image, F is the foreground in I , B is the background in I , and α_p is the *opacity* of the foreground, respectively. Normally users need to provide *hints* through the aforementioned form of a *trimap* to distinguish the *foreground*, *background* and *unknown* parts of an image. Basically matting algorithms can be classified into three categories: *sampling-based*, *propagation-based* and *combined*. *Sampling-based* approaches derive/interpolate the α values for the unknown pixels based on the color distributions of the known foreground and background pixels, such as Chuang et al.'s *Baysian matting* [6]. Through the definition of *affinity* between adjacent pixels, *propagation-based* approaches estimate the α values of neighboring pixels based on the known ones, and normally they convert matting problems into corresponding *linear systems* which are solved by matrix operations. *Poisson matting* [7], proposed by Sun et al., is an example of such. *Closed-form matting* [8] adopted a similar approach.

Some combined approaches also exist, such as Wang et al.'s robust matting [9]. Evaluating the quality of a matting result is an important issue. According to a popular website [10], four metrics are proposed to measure the matting quality, including *sum of absolute difference* or *SAD* for short, *mean square error*, or *MSE* for short, *gradient* and *connectivity*, where the latter two are to compare the difference between two alpha maps in terms of gradient and connectivity, respectively. In this paper, we resort to *closed-form* matting for the following two reasons. First, our inputs are relatively sparse, and so far the best implementation of *propagation-based* matting approaches is *closed-form* matting. Second, the source code of *close-form* matting is publicly available.

The final related technique is *automatic hair segmentation*. And examples can be seen from the works proposed by Rousset et al. [11] and Lipowezky et al. [12].

3 Automatic Hairstyle Extraction

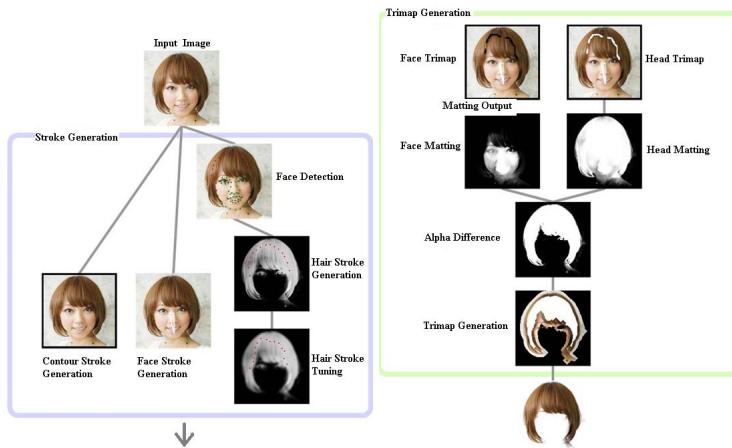


Fig. 1. System overview

3.1 System Overview

Our current system assumes that the face and hair of the input photo are strictly within the image without touching the boundary of the input photo. And the *hairstyle extraction* consists of the following three steps. First, we have to define the *foreground* and *background* through the help of *ASM* so that automatic *strokes* can be generated. Second, we apply *matting* techniques to identify the initial *image segmentations*. Along the hair segment, we extend its boundary into a *narrow band* to be used as a *trimap* for being the input for the next stage. Finally, given the trimap the final *matting* process is applied to extract

the desired hairstyle from the input image. Figure 1 shows a system overview of our approach and the details regarding each step is to be given in the ensuing sub-sections.

3.2 Stroke Generation

Stroke generation includes three parts, the *face stroke*, the *hair strokes*, and the *boundary stroke*, where the first two are related to face detection, while the last one requires only the dimension information of the input image. We now describe the generation of these strokes in detail.

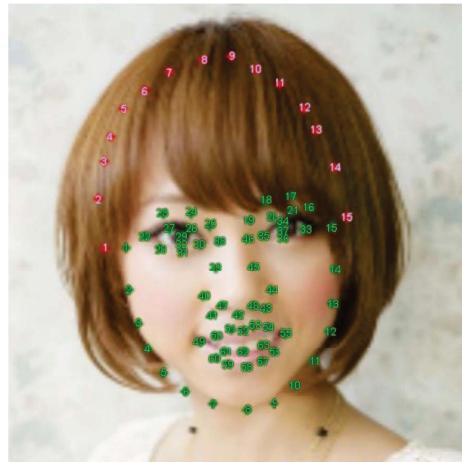


Fig. 2. ASM for face detection, where the green dots are detected by ASM, and the red dots, generated based on the green dots, are for placing the hair strokes

First, from the face detected by ASM, 63 feature points are defined, as shown in green in Figure 2. Let us denote these feature points as ASM_1 to ASM_{63} hereafter. To place the face stroke, the trick is to avoid touching the neck, eyes, and hair portion of a photo. Currently we set the *face stroke* to be of a triangle shape, consisting of ASM_{38} , ASM_{46} and ASM_{58} .

Second, we could set up the initial positions of the points in the hair, denoted as H_1 to H_{15} and shown in red in this figure. More specifically, according to the well known fact that human eyes often locate near the vertical mid point between the top of the head and bottom of the chin, we therefore treat the line connecting both eyes (ASM_{32} and ASM_{37}) as a horizontal mirroring axis, to flip up the facial contour points from ASM_1 to ASM_{15} to become the points from $Mirror_1$ to $Mirror_{15}$. By setting the mid point of ASM_1 and ASM_{15} to be the central position C for a face, that is, $C = \frac{ASM_1 + ASM_{15}}{2}$, then the initial point for H_i is:

$$H_i = [1 + 0.2(\frac{8-i}{7})^2](\text{Mirror}_i - C) + C \quad (2)$$

And then we connect two consecutive H_i and H_{i+1} to form hair stroke L_i , as shown in Figure 3(a). Later on we show how to tune the positions of these initial points so that they could be good candidates for representing the hair. We then connect these points to form a set of *piecewise linear* line segments, and from which we select a *reliable* subset of these line segments for being the *hair strokes*.

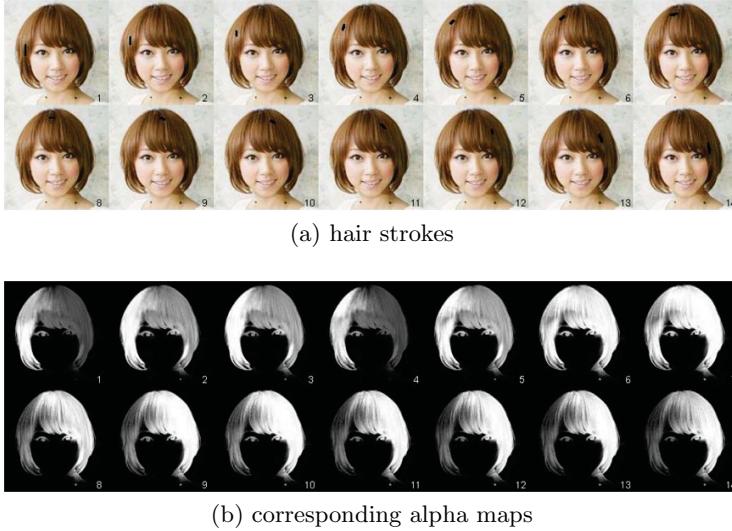


Fig. 3. Hair stroke generation

Finally, the boundary stroke is composed of the rectangular boundary of the input image, and based on the assumption that the input photo is fully contained within the image, the boundary stroke would not intersect with the hair portion.

3.3 Hair Stroke Tuning

It is easy to see that the previous setting of hair strokes may not be accurate as the hair configuration could vary from person to person. The goal for this hair stroke tuning is to avoid hair strokes from intersecting with the skin or background area of the input photo. For a better tuning, we first calculate the *similarity* among L_i s, and then based on the *proximity matrix* to remove *outliers* of the initial hair strokes, followed by adjustment on the positions of the remaining hair strokes.

By treating hair stroke L_i as the foreground stroke, and the boundary stroke as the background stroke, we could obtain the resulting *alpha map* A_i , and the

similarity among L_i s is defined upon the similarity among A_i s. Figure 3(b) shows the corresponding alpha map for each L_i .

As the pixel value in an alpha map is a floating point, we define the similarity between two alpha maps A and B as:

$$\text{similarity}(A, B) = \frac{\sum_p \min(A_p, B_p)}{\sum_p \max(A_p, B_p)} \quad (3)$$

where it is evident that the more similar these two image to each other, the higher this value. For outlier detection, we adopt a *distance-based* approach, and the parameter setting is, $r = 0.7$ and $n = 6$. And in case more than half (7 hair strokes) are determined to be outliers, we only remove the 7 hair strokes that are the most far from others.

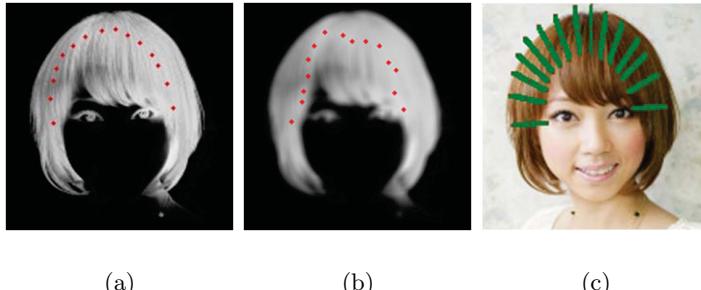


Fig. 4. Stoke tuning

We now describe how to tune the remaining hair strokes. Assume that there are q hair strokes left, and S be the average map of these q alpha maps, as shown in Figure 4(a). To avoid the interference of potential noise, we further apply a 5×5 *uniform blur filter* on S . As a higher value in S represents a higher probability to be hair, we try to adjust the positions of hair point H_i so that the resulting hair strokes can pass through the area with higher values, and the result is shown in Figure 4(b). Note that the hair points H_i cannot move too drastically so that the hair points themselves or the connecting hair strokes L_i may fall out of the hair area. For this reason, let $M_i = H_i - C$, then $S = C + 0.7M_i$, denotes the lower (closer to the center) bound and $T = C + 1.3M_i$, the upper bound for the hair point movement, respectively. Let HM_i to be the set of pixels on \overline{ST} , and further let $HM_{i,j}$ be a point j in this set, where j ranges from 1 to $|HM_i|$, and it represents the j -th point in this set. And $LM_{i,j,k}$ denotes the line connecting $HM_{i,j}$ and $HM_{i-1,k}$. Figure 4(c) shows 15 green lines, from HM_1 to HM_{15} , where each line has three red points, denoting the corresponding H_i , S and T mentioned previously.

To guide the movement of H_i , we define the *objective function* to be: finding the hair strokes in S that maximize average alpha value beneath each hair stroke. More specifically, we use a vector J to represent the choices we collectively make

on each green line, and further assume that on HM_i we choose J_i , then the objective function can be formulated as the following:

$$J^* = \arg \max_J \left(\sum_{i=1}^{14} cost(LM_{i,J_{i+1},J_i}) \right) \quad (4)$$

Here $cost(L)$ is defined as:

$$cost(L) = \frac{1}{|L|} \sum_{p \in L} S_p \quad (5)$$

where S_p is a pixel p in the average alpha map S .

Let $L_{i,j}$ be the maximum cost that a hair stroke ends at $HM_{i,j}$, then such an optimization problem can be solved by a *dynamic programming* method by defining the following recursive relationship:

$$L_{i,j} = \begin{cases} 0 & \text{if } n = 0 \\ \max_k (cost(LM_{i,j,k}) + L_{i-1,k}) & \text{if } n > 0 \end{cases} \quad (6)$$

3.4 Trimap Generation

To derive the desired hair portion through matting, intuitively we could treat the boundary stroke and face stroke as the background strokes and the hair strokes as the foreground strokes, as shown in Figure 5(a). However, the shadow on the neck, as well as the eye balls could potentially be mistaken as part of the hair, as shown in Figure 5(b). To deal with this problem, we separately perform *head matting* and *face matting* and subtract the alpha map of the latter one from that of the former one to derive the desired hair matting result.

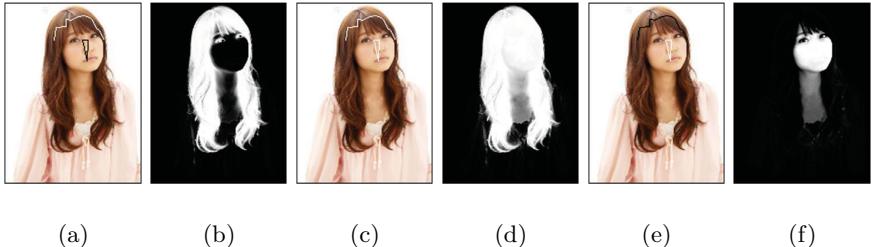


Fig. 5. Trimap generation

The head matting is performed by setting the hair strokes and face stroke as the foreground strokes, and the boundary stroke as the background stroke, as shown in Figure 5(c), and the resulting alpha map, called *head alpha* hereafter, is shown in Figure 5(d). As for the face matting, it treats the face stroke as the foreground stroke, and hair strokes together with the boundary stroke as

the background strokes, as shown in Figure 5(e), and the resulting alpha map, called *face alpha* hereafter, is shown in Figure 5(f). We observe that head alpha and face alpha overlap significantly, and their difference roughly corresponds to the *hair alpha* that we want to obtain. Let (m, n) correspond to $m \times \text{head}_{\text{alpha}} - n \times \text{face}_{\text{alpha}}$, then after numerous experiments and observations, we found that the combination of (2, 4) in the alpha space normally gives us the best result, that is, we could obtain a clean and clear alpha that excludes the neck. However, the resulting *alpha difference* could still include undesired eye regions, further processing is required to get rid of the eye regions. As shown in white in Figure 6(a), and the pixels with value smaller than 0.5 as the background stroke, as shown in black in Figure 6(b), then the *final trimap*, and as shown in Figure 6(c), can be generated. The *final alpha map* and the *final extracted hairstyle*, are shown in Figure 6(d) and (e), respectively.

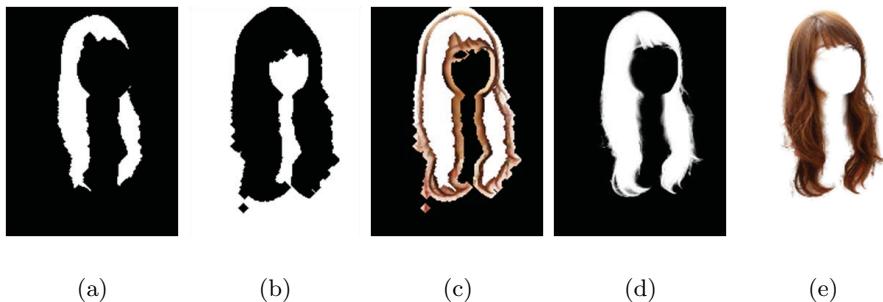


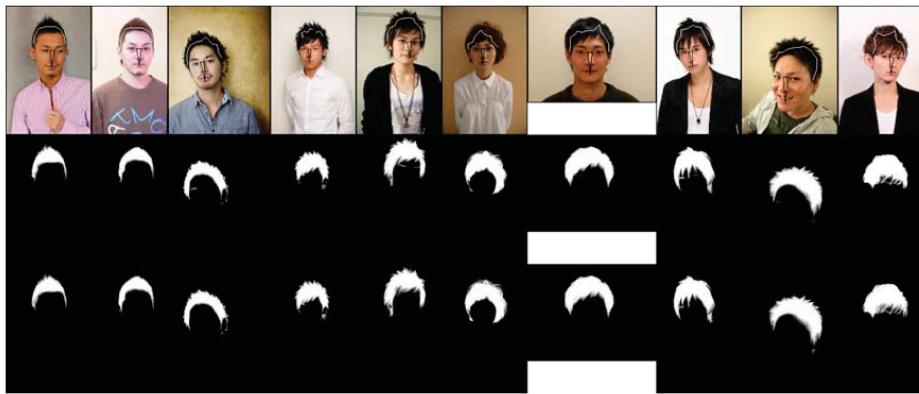
Fig. 6. Matting

4 Results

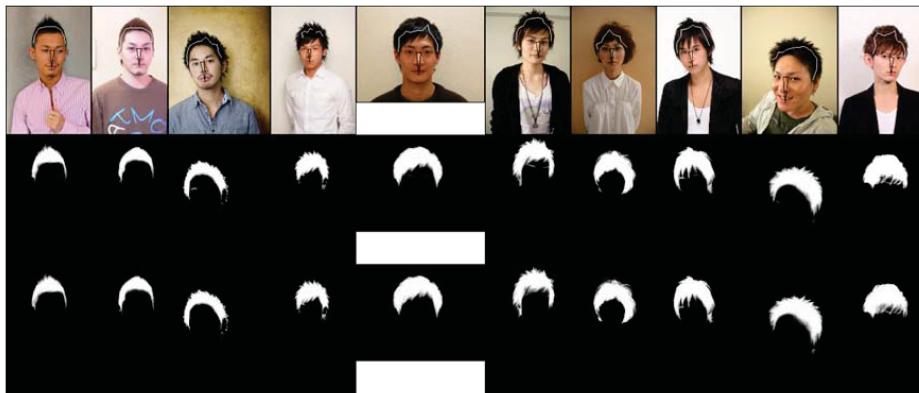
To test our proposed automatic hair extraction algorithm, we have randomly collected 70 photos from a Japanese hairstyle website <http://www.beauty-box.jp>, manually drawn their trimaps, executed the closed-form matting, and used the resulting alpha maps as the ground truth to evaluate the results. We adopt the evaluation metrics MSE and SAD, together with our proposed *alpha similarity* metric, as shown in Equation 3. Note that in terms of MSE and SAD metrics, they are the smaller the better, our proposed alpha similarity, on the other hand, is the larger the better. And all the images are uniformly scaled to the size of $1XX \times 200$ or $200 \times 1XX$.

Figure 7 shows the 10 best results based on MSE, SAD, and our proposed alpha similarity. The first row shows the hair trimaps, the second row the final alpha maps, while the third row the ground truth alpha maps as mentioned previously. Note that the leftmost column shows the best among these 10, the rightmost one the worst.

It can be observed that when the hair strokes intersect with the background or the face area, the results become evidently worse. In addition, using MSE and



The best 10 photos based on MSE



The best 10 photos based on SAD

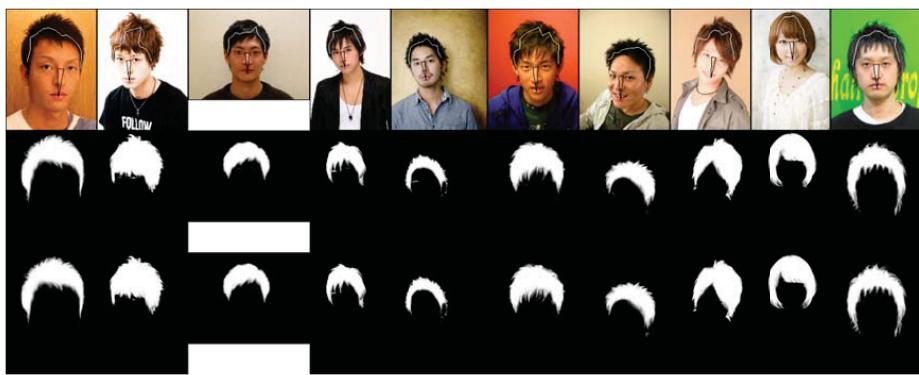


Fig. 7. Metric Comparisons

SAD as the similarity metric tend to favor the cases where the foreground area is smaller, but our proposed metric does not suffer as much, and thus is better. Using such a metric to evaluate the resulting alpha maps, our proposed method can achieve about 86.2% accuracy on an average.

5 Conclusions

We have proposed an automatic hair style extraction system. The contribution of this paper is three fold. First, given a photo, our algorithm can *automatically generate a desired trimap*, and in particular, by removing the unwanted portion of neck and eyes from the matting process. Second, we propose an *alpha similarity* metric that can better match human vision system in terms of evaluating the quality of an derived alpha map. Finally, we introduce the concept of an *alpha space* that allow us to systematically probe for better parameters used to produce the best matting result. In general, our hair style extraction can achieve 86.2% accuracy on an average.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision And Pattern Recognition 2001 (2001)
2. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Computer Vision and Image Understanding 61, 38–59 (1995)
3. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001)
4. Ben-Gal, I.: Outlier Detection. In: Data Mining and Knowledge Discovery Handbook. Springer (2005)
5. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. The VLDB Journal - The International Journal on Very Large Data Bases 8 (2000)
6. Chuang, Y., Curless, B., Salesin, D., Szeliski, R.: A bayesian approach to digital matting. In: Computer Vision and Pattern Recognition 2001 (2001)
7. Sun, J., Jia, J., Tang, C., Shum, H.: Poisson matting. In: Siggraph 2004 (2004)
8. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. In: Computer Vision and Pattern Recognition 2006 (2006)
9. Wang, J., Cohen, M.: Optimized color sampling for robust matting. In: Computer Vision and Pattern Recognition 2007 (2007)
10. Rhemann, C., Rother, C., Wang, J., Gelautz, M., Kohli, P., Rott, P.: A perceptually motivated online benchmark for image matting. In: Computer Vision and Pattern Recognition 2009 (2009)
11. Rousset, C., Coulon, P.Y.: Frequential and color analysis for hair mask segmentation. In: International Conference on Image Processing 2008 (2008)
12. Lipowezky, U., Mamo, O., Cohen, A.: Using integrated color and texture features for automatic hair detection. In: Convention of Electrical and Electronics Engineers in Israel 2008 (2008)

Evaluation of Image Forgery Detection Using Multi-scale Weber Local Descriptors

Sahar Q. Saleh¹, Muhammad Hussain¹, Ghulam Muhammad¹, and George Bebis²

¹ College of Computer and Information Sciences, King Saud University, Riyadh 11543,
Saudi Arabia Department of Computer Science and Engineering

² University of Nevada at Reno, USA
{mhussain, ghulam}@ksu.edu.sa, bebis@cse.unr.edu}

Abstract. In this paper, a detailed evaluation of multi-scale Weber local descriptors (WLD) based image forgery detection method is presented. Multi-scale WLD extracts the features from chrominance components of an image, which usually encode the tampering information that escapes the human eyes. The WLD incorporates differential excitation and gradient orientation of a center pixel around a neighborhood. In the multi-scale WLD, three different neighborhoods are chosen. A support vector machine is used for classification purpose. The experiments are conducted on three image databases, namely, CASIA v1.0, CASIA v2.0, and Columbia color. The experimental results show that the accuracy rate of the proposed method are 94.19% for CASIA v1.0, 96.61% for CASIA v2.0, and 94.17% for Columbia dataset. These accuracies are significantly higher than those obtained by some state-of-the-art methods.

Keywords: image forgery detection, Weber local descriptors, image splicing, copy-move forgery.

1 Introduction

Nowadays, we are living in an age, where digital imaging has grown and developed to become the widespread technology. With the increasing applications of digital imaging, different types of software are introduced for image processing. Such software can do an alteration in digital image by changing blocks of an image or combining two images with no showing the effect of the modification in the forged image. The most commonly used forgery is copy-move forgery (CMF), where a region within an image is copied and moved to another region in the same image in order to conceal an important object from the original image. The copied block may be changed by any kind of pre-processing such as rotation, scaling, additive noise, etc. to suit the copied area with the whole image. In another type of forgery, one part of an image is copied and pasted to another image. This type of forgery is called image splicing.

Many techniques have been developed for authenticity checking of the digital images. These techniques can be divided into intrusive (active) and non-intrusive (blind or passive) [1]. In active techniques, particular data is embedded in the digital

images for supporting multimedia digital authentication and rights safety. If the image contents are modified, the embedded data is also changed. The image authenticity is verified by checking whether the true signature corresponds to the signature that is retrieved from the suspicious test image. These techniques are restricted because of the inability of many digital cameras to embed the signature. Due to the restrictions of active techniques, the researchers tend to develop non-intrusive techniques for validating the authenticity of digital images. These techniques examine images with no embedded data such as signatures or watermarks, and result whether these images are authentic or tampered.

An improved DCT (discrete cosine transform)-based technique was proposed in [2] to discover CMF in digital images. The image is subdivided into blocks, and the DCT is computed. The DCT coefficients are lexicographically sorted, and compared with different blocks. The proposed technique is robust against JPEG compression, additive white Gaussian noise, or blurring distortion. Cao et al [3] proposed an improved DCT-based method to locate the duplicated regions in a given image. The method uses the circle block for representing the DCT coefficient's array.

Noise pattern based image forgery detection method was proposed in [4]. Noise pattern is obtained by subtracting the denoised image from the input image. Then, histograms of noise from different segments of the image are compared to find the distortion caused by image forgery. Peng et al [5] also used sensor pattern noise to detect image forgery. Instead of using the histogram, they use four statistical measures, namely, variance, entropy, signal-to-noise ratio, and average energy gradient, from the noise pattern. He et al in [6] proposed a method relied on approximate run length (ARL) to detect CMF. Firstly, the edge-gradient array of a given image is calculated, and then the ARL is computed along the edge-gradient orientation. Zhao *et al* used chrominance spaces with RLRN (run-length run-number) for CMF detection [7]. The input color image is transformed into the YCbCr color mode. Then RLRN is used to extract the features from the de-correlation of the chrominance channels. Support vector machine (SVM) was used for classification purpose. This method gave better performance with JPEG image format than the TIFF image format. Shi *et al* proposed statistical features based on 1D and 2D moments, and transition probability features based on Markov chain in DCT domain for image splicing detection [20]. In CASIA v2.0 database [16], the method achieves 84.86% accuracy. Later, He *et al* improved the method by combining transition probability features in DCT and DWT domains [21]. For classification, they used SVM - recursive feature elimination (RFE). Their method obtains 89.76% accuracy on the CASIA v2.0 database.

Undecimated wavelet transforms (UWT) based image forgery detection was proposed in [8]. Approximation and detailed coefficients of the UWT from overlapping blocks of an image are used to find the similarity between the blocks. The method is robust against JPEG compression and a certain degree of rotation and scaling. Scale invariant feature transform (SIFT) based forgery detection methods are proposed in [9], [10], [11]. They are quite robust against rotation and scaling post-processing. Two good surveys can be found in [1], [12].

In this paper, we give a detailed evaluation of a method based on a multi-scale Weber's law descriptor [13] and SVM [14] for detecting image forgery. The forgery can be either copy-move or spliced. The proposed method is evaluated on three publicly available image databases designed for forgery detection.

The rest of the paper is organized as follows. Section 2 presents the image forgery detection method, Section 3 gives experimental results with discussion, and finally, Section 4 draws some conclusion.

2 Forgery Detection Method

Fig. 1 shows a block diagram of the proposed image forgery detection method. In the first step, input color image is converted into the YCbCr color space. Image forgers generally do image tampering in RGB color-space and attempt to wrap manipulated traces. YCbCr color space stores the color in terms of its luminance and chrominance. The human eyes are less sensitive to chrominance than luminance; however, even a tampered image looks natural, some tampered traces are left in the chrominance channels [7]. In the second step, the chrominance component (either Cr or Cb) is used to extract image features in the form of Weber local descriptors (WLD) [13]. Multi-scale WLD is introduced where the histograms from different operators of variation (P, R) are concatenated and used to represent the image features; P is the count of the neighbors, and R is the spatial-scale for the operator. In the last step, SVM based classifier is used to classify the input image as authentic or forged.

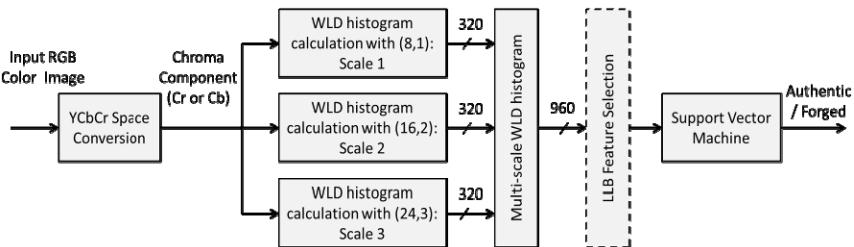


Fig. 1. Block diagram of the proposed image forgery detection method

WLD is a robust local descriptor, which is based on the fact that human sensitivity of a sample relies on the change of the original stimulus intensity [13]. WLD descriptor is described below for feature extraction purpose. WLD based on Weber's law has two components: differential excitation (D) and orientation (Φ).

Ernst Weber viewed that the ratio of the increase threshold to the intensity of the background is a constant. It is formulated as

$$\frac{\Delta x}{x} = C \quad (1)$$

Where x is the initial stimulus-intensity, Δx is the increase in threshold (noticeable distinction), and C is a constant.

f_{00}	f_{01}	f_{10}	f_{11}
+1 +1 +1	0 0 0		
+1 -8 +1	0 +1 0	-1	
+1 +1 +1	0 0 0	+1	-1

Fig. 2. Filters used in WLD calculation

A differential excitation (D) is used to change the intensity of each pixel in an image. The $D(p_c)$ for a pixel p_c is calculated as follows:

Step1: Compute the difference between the pixel p_c and its neighbours via the filter (f_{00}) in Fig. 2.

$$k_s^{00} = \sum_{i=0}^{N-1} (\Delta p_i) = \sum_{i=0}^{N-1} (p_i - p_c) \quad (2)$$

where p_i is the i th neighbour of pixel p_c and N is the number of neighbours.

Step2: Calculate the proportion of the differences to the current pixel intensity by the outputs of the filter (f_{00}) and (f_{01}) in Fig. 2.

$$I = \frac{k_s^{00}}{k_s^{01}} = \sum_{i=0}^{N-1} \left(\frac{p_i - p_c}{p_c} \right) \quad (3)$$

The differential excitation $D(p_c)$ of the current pixel p_c is

$$D(p_c) = \arctan \left[\sum_{i=0}^{N-1} \left(\frac{p_i - p_c}{p_c} \right) \right] \quad (4)$$

WLD orientation component is the gradient orientation, $\Phi(p_c)$, and it is calculated as follows:

$$\Phi(p_c) = \arctan \left(\frac{k_s^{11}}{k_s^{10}} \right) \quad (5)$$

where, k_s^{11} and k_s^{10} are the outputs of the filters f_{11} and f_{10} .

Later, Φ is mapped to Φ' and is quantized into T dominant directions. After calculating differential excitation and gradient orientation, WLD histogram is formed. In WLD histogram, there are three parameters that affect on optimizing the results: the number of dominant orientations (T), the number of differential excitation segments (M), and the number of bins in sub histogram segments $H_{m,t}(S)$.

In the proposed multi-scale WLD, differential excitation and gradient orientation are calculated in three different neighborhoods, which are (8,1), (16,2), and (24,3), where the first component inside the parenthesis corresponds to the number of neighboring pixels and the second component is the radius of the neighbors from the center pixel (scale). The histograms from these three neighborhoods are fused to produce the multi-scale WLD histogram.

Local learning based (LLB) feature selection technique is applied to the whole feature vector to reduce the dimension [15]. The main design of this technique is to

decompose a randomly complicated non-linear problem into a group of locally linear problems by using local learning, and the feature relevance is learned globally in the maximum margin framework.

3 Experiments

The proposed method is evaluated in three publicly available databases that are designed for image forgery detection. The three databases are CASIA TIDE v1.0 and v2.0 [16], and Columbia authentic and spliced color image database [17].

Different scales with various numbers of neighborhoods in the WLD are used. We name the scaling from C1 to C7, where C1 means (8, 1), C2 corresponds to (16, 2), C3 refers to (24, 3), C4 is a combination of (8, 1) and (16, 2), C5 is a combination of (8, 1) and (24, 3), C6 is a combination of (16, 2) and (24, 3), and finally, C7 is the combination of all the scales (8, 1), (16, 2) and (24, 3). For (T, M, S) parameters of WLD, various combinations are tried and finally fixed to (4, 4, 20) that gives the optimal result.

Performance of the proposed method with SVM classification by employing RBF kernel and polynomial kernel has been evaluated using a 10-fold cross validation. The polynomial kernel performs better than the RBF kernel in our experiments, so we report results only with polynomial kernel. Grid search method is used to find the optimal parameters of SVM. LIBSVM is utilized for SVM implementation [18]. The performance of the method is given in terms of accuracy (averaged over ten iterations).

3.1 Experiments with CASIA v1.0

CASIA v1.0 dataset has 800 authentic images and 921 forged images of which 459 are copy-move forged and the remaining are spliced. Scaling and rotation have been applied on some of the forged images. All the images have the size of 384×256 pixels, and they are in JPEG format.

Fig. 3 shows the effect of different WLD scales in Cr channel. For individual scale (C1, C2, C3), C3 performs the best. In the case of multi-scale, C7, which is the combination of all the three scales, has the highest detection accuracy. With Cr channel, C7 achieves 92.62% accuracy, while with Cb channel, it obtains 88.66% (not shown). Therefore, it is experimentally proved that the multi-scale WLD performs better than the single scale WLD in the case of image forgery detection. Each single scale WLD produces 320 features (bins in the histogram), so C7 has a total of (320×3=) 960 features. All the subsequent results are with C7.

In the next experiment, Cr and Cb histograms are combined to see the accuracy. We call this combination as feature level fusion (FLF) and the feature vector is of dimension 1920 (=960×2). After feature selection, this dimension is reduced to 770. In the experiments, three cases are considered: testing with spliced images, testing

with copy-move images, and testing with the whole dataset. Fig. 4. shows the detection accuracies (%). FLF performs better than individual chrominance channel, and it achieves 94.19% accuracy for the full CASIA v1.0 dataset. False positive rate and false negative rate is 6.3% and 3.7%, respectively. Splicing detection accuracy (94.52%) is higher than copy-move forgery detection accuracy (92.08%). Fig. 5. shows the ROC curve of the proposed method on the full dataset. For comparison purpose, we implemented the method described in [19] and evaluated it on the full CASIA v1.0 dataset using Cr channel. The method [19] obtains 78.53% accuracy, which is much less than 92.62% achieved by the proposed method using Cr.

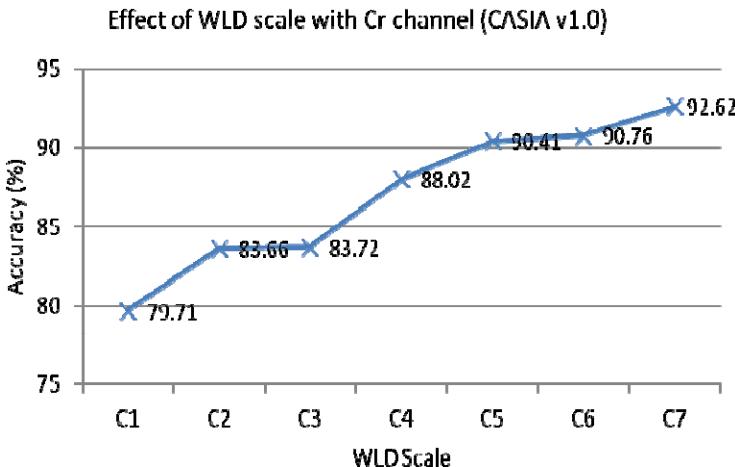
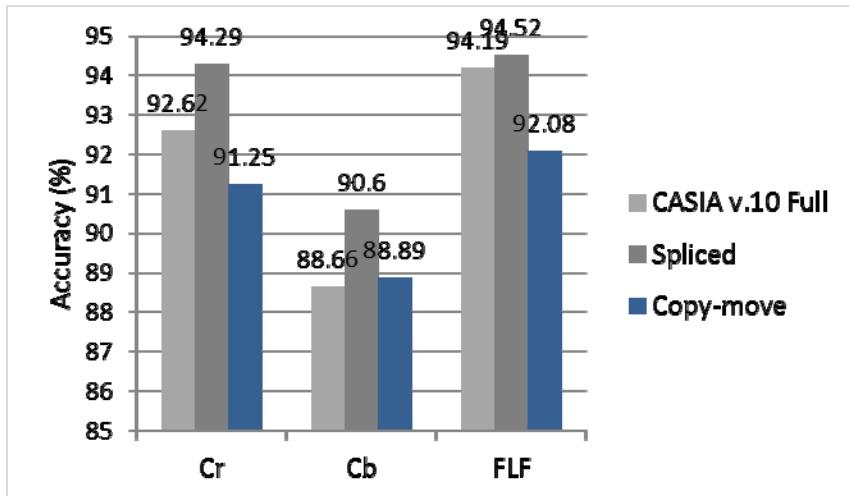
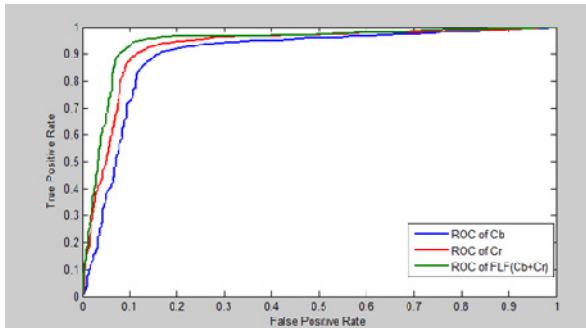


Fig. 3. Accuracy of splicing detection using chrominance channel, Cr, in different scales of WLD

3.2 Experiments with CASIA v2.0

CASIA v2.0 database consists of 7491 authentic and 5123 forged images of JPEG, BMP, and TIFF format, where image sizes vary from 240×160 to 900×600 pixels. Scaling and rotation have been applied on some of the forged images. The experiments are performed with the full dataset. Table 1 shows the result of the proposed method in terms of accuracy (%) with standard deviation (sd) and area under curve (AUC). The best performance (accuracy = 96.61%) is achieved with the Cb channel. It is noted that no feature selection is applied in the experiments in Table 1. The performance of the proposed method is far superior to that of the other state of the art methods [20], [21] evaluated in full CASIA v2.0. Table 2 shows a comparison of accuracies between the methods.

**Fig. 4.** Accuracy of the proposed method in CASIA v1.0**Fig. 5.** ROC curves of the proposed method in CASIA v1.0

3.3 Experiments with Columbia Color

Columbia color image database consists of 183 authentic and 180 spliced images of TIFF format. The image size is 1152×768. Table 3 shows the results with feature selection. The proposed method with FLF achieves 94.17% accuracy, which is better than the previously reported best result by the method in [22]. The number of features in FLF after feature selection is 316.

Table 1. Performance of the proposed method in CASIA v2.0

Channel	Acc(%) \pm sd	AUC \pm sd
Cr	96.38 \pm 0.36	0.966 \pm 0.0049
Cb	96.61 \pm 0.49	0.969 \pm 0.0038
FLF	96.28 \pm 0.675	0.96 \pm 0.0076

Table 2. Accuracies of the three methods in CASIA v2.0

Proposed Method	Method [20]	Method [21]
96.61%	84.86%	89.76%

Table 3. Performance of the proposed method in Columbia

Channel	Acc(%) \pm sd	AUC \pm sd	Acc (%) of [22]
Cr	92.5 \pm 4.73	0.93 \pm 0.05	93.14
Cb	92.78 \pm 3.51	0.93 \pm 0.05	
FLF	94.17 \pm 3.57	0.93 \pm 0.05	

4 Conclusion

A detailed evaluation of a multi-scale WLD based image forgery detection method is presented. WLD features are extracted from the chrominance channels of a color image. SVM is used for classification purpose. The best results achieved by the forgery detection method are 94.19% with CASIA v1.0, 96.61% with CASIA v2.0, and 94.17% with Columbia color image databases. These accuracies are better than some of the previously reported results in these databases. The performances of Cb and Cr channels are comparable, while their fusion gives the best result except in the case of CASIA v2.0. A future work will be to localize the forgery in a tampered image.

Acknowledgement. This work is supported by the National Plan for Science and Technology, King Saud University, Riyadh, Saudi Arabia under project number 10-INF1140-02.

References

1. Mahdian, B., Saic, S.: A bibliography on blind methods for identifying image forgery. *Signal Processing: Image Communication* 25(6), 389–399 (2010)
2. Huang, Y., Lu, W., Sun, W., Long, D.: Improved DCT-based detection of copy-move forgery in images. *Forensic Science International* 206(1), 178–184 (2011)
3. Cao, Y., Gao, T., Fan, L., Yang, Q.: A robust detection algorithm for copy-move forgery in digital images. *Forensic Science International* 214(1), 33–43 (2012)
4. Muhammad, N., Hussain, M., Muhamad, G., Bebis, G.: A non-intrusive method for copy-move forgery detection. In: Bebis, G., et al. (eds.) ISVC 2011, Part II. LNCS, vol. 6939, pp. 516–525. Springer, Heidelberg (2011)
5. Peng, F., Nie, Y.-Y., Long, M.: A complete passive blind image copy-move forensics scheme based on compound statistics features. *Forensic Science International* 212(1), e21–e25 (2011)

6. He, Z., Sun, W., Lu, W., Lu, H.: Digital image splicing detection based on approximate run length. *Pattern Recognition Letters* 32(12), 1591–1597 (2011)
7. Zhao, X., Li, J., Li, S., Wang, S.: Detecting digital image splicing in chroma spaces. In: Kim, H.-J., Shi, Y.Q., Barni, M. (eds.) IWDW 2010. LNCS, vol. 6526, pp. 12–22. Springer, Heidelberg (2011)
8. Muhammad, G., Hussain, M., Bebis, G.: Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digital Investigation* (2012)
9. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: A SIFT-Based Forensic Method for Copy-Move Attack Detection and Transformation Recovery. *IEEE Transactions on Information Forensics and Security* 6(3), 1099–1110 (2011)
10. Huang, H., Guo, W., Zhang, Y.: Detection of copy-move forgery in digital images using SIFT algorithm. In: Pacific-Asia Workshop on Computational Intelligence and Industrial Application, PACIIA 2008, pp. 272–276. IEEE (2008)
11. Ling, H., Zou, F., Yan, W.-Q., Ma, Q., Cheng, H.: Efficient image copy detection using multi-scale fingerprints (2011)
12. Farid, H.: Image forgery detection. *IEEE Signal Processing Magazine* 26(2), 16–25 (2009)
13. Chen, J., Shan, S., He, C., Zhao, G., Pietikainen, M., Chen, X., Gao, W.: WLD: A robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9), 1705–1720 (2010)
14. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press (2000)
15. Sun, Y., Todorovic, S., Goodison, S.: Local-learning-based feature selection for high-dimensional data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9), 1610–1626 (2010)
16. CASIA image tampering detection evaluation database (CASIA TIDE) v1.0 and v2.0, <http://forensics.idealtest.org>
17. Ng, T.-T., Chang, S.-F., Sun, Q.: A data set of authentic and spliced image blocks. Columbia University, ADVENT Technical Report, 203-2004 (2004)
18. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3), 27 (2011)
19. Wang, W., Dong, J., Tan, T.: Image tampering detection based on stationary distribution of markov chain. In: 2010 17th IEEE International Conference on Image Processing, ICIP, pp. 2101–2104. IEEE (2010)
20. Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. In: Proceedings of the 9th Workshop on Multimedia & Security 2007, pp. 51–62. ACM (2007)
21. He, Z., Lu, W., Sun, W., Huang, J.: Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recognition* (2012)
22. Zhao, X., Li, S., Wang, S., Li, J., Yang, K.: Optimal chroma-like channel design for passive color image splicing detection. *EURASIP Journal on Advances in Signal Processing* 2012(1), 1–11 (2012)

Energy-Transfer Features for Pedestrian Detection

Radovan Fusek, Eduard Sojka, Karel Mozdřeň, and Milan Šurkala

Technical University of Ostrava, FEECS, Department of Computer Science,

17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

{radovan.fusek,eduard.sojka,karel.mozdren,milan.surkala}@vsb.cz

Abstract. In this paper, we propose an interesting and novel method for computing the image features that are useful for object detection. The method is interesting and novel in the terms of the feature vector dimensionality and object information capturing. In the proposed method, the areas of objects (that contain the important information useful for recognition) are described by the distribution of energy. The energy is transferred through the energy sources that are placed into the image and the distribution of energy is encoded into a vector of features. The vector is then used as an input for the SVM classifier. Using this approach, the objects of interest can be successfully described with a relatively small set of numbers if compared with the state-of-the-art descriptors that are based on the histograms of oriented gradients. We show the robustness of the features in the task of pedestrian detection.

1 Introduction

The objects of interest can be described using a lot of image information (e.g. shape, texture, colour). In the area of feature based detectors, the image features are carriers of this information. The goal is to design such features that are able to successfully describe the different objects of interest with a relatively small set of numbers. In this area and especially in the area of human detection, the features that are based on the Histograms of Oriented Gradients (HOG) [1] are dominant in the recent years. In HOG, the information about the distribution of gradient magnitudes and directions is used for the object description. The histograms of gradients are computed for each position of the sliding window that is divided into the blocks that consist of small connected cells. A feature vector is composed from these histograms and the vector is then used as an input for the trainable classifiers (e.g support vector machine). The detection methods based on these descriptors have been successfully presented in many papers (see Section 2).

Nevertheless, the classical HOG descriptors suffer from the high dimensionality of feature vector and sometimes it is useful to apply the methods for reducing the feature space (e.g. principal component analysis). The high dimensionality of feature vector negatively affects the speed of detection and training phases

and the large training set must be used for perfect object detection in the variable surroundings (streets, buildings, airports, etc.). In additional to that, the features that are based on the edge information (e.g. length, magnitude, orientation, localization) are also sensitive to the noise due to the fact that noise have a negative effect to the image quality (quality of edges). The noise must be suppressed, but the images can lose the important information about the object edges after the filtering. These shortcomings create the motivation for developing the novel approach for computing the image features that can be successful with a relatively small dimensionality of feature vector and with the filtering step that is directly included in the extraction of proposed features.

Basically, the proposed method is inspired by HOG but instead of the distribution of gradient magnitudes and directions the method captures the object information using the distribution of energy. In essence, the main idea of the proposed features is that the areas of objects (that consist of information about the shape) can be effectively described by the energy distribution. We will consider the transfer of energy as the transfer of heat in this paper. In our approach, the sliding window is also divided into the regions. The sources of temperature are defined inside each region. After the temperature transfer, the distributions of temperature inside the regions are encoded to the feature vector. Finally, the vector of features is used as an input for the SVM classifier. Since the temperature is transferred within the object areas, using this approach, we are able to describe the object areas with the positive filtration abilities that are obtained from the diffusion equation.

The next parts of the paper are organized as follows. The related works are described in Section 2. The process of extraction of the proposed features is described in Section 3. Finally, the results are shown in Section 4.

2 Related Works

In the area of feature based detectors, the methods that are based on the Histograms of Oriented Gradients [1] have been successfully presented in the recent years. Pedestrian detection method using infrared images and histograms of oriented gradients combined with the SVM classifier was presented in [2]. Near real-time human detection system using the cascade-of-rejectors with the histograms of oriented gradients was proposed in [3]. In [4], the authors applied the principal component analysis to the HOG feature vector to obtain the PCA-HOG vector. This vector contains the subset of HOG features and such vector is used as an input for the SVM classifier. Their method was used for pedestrian detection with the satisfactory results. The method for vehicle detection in low-altitude airborne videos using boosting HOG features was presented in [5]. In [6], the authors proposed Augmented Histograms of Oriented Gradients (AHOG) feature for human detection from a non-static camera. Their approach extended the classical HOG features by adding the human shape properties. The authors reported that the method achieved a good performance at many views of targets. The feature set that contains the combination of Histograms of Oriented Gradients and Local Binary Pattern (HOG-LBP) for human detection was

presented in [7]. Pyramid of Histogram of Orientation Gradients (PHOG) was proposed in [8]. This method uses the combination of the image pyramid representation and the histograms of orientation gradients. The very popular method for object detection was presented by Viola and Jones in [9]. Their method uses the integral image, rectangular features, and AdaBoost algorithm. The method was used for moving-human detection in [10].

3 Proposed Features

The main idea behind the proposed features is that the appearance of objects can be efficiently described by the function of energy distribution. Especially, the information about the object areas that are useful for recognition are precisely described by this distribution in the presented method. The usefulness of energy distribution can be described in the following way.

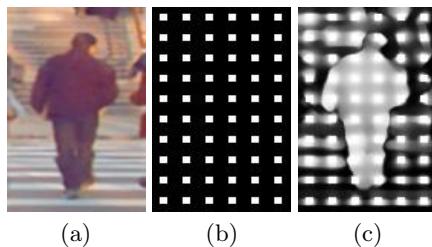


Fig. 1. The real-life image (a). The regular grid of sources (b). The visualization of distribution of temperature from these sources (c). The value of temperature is depicted by the level of brightness.

Consider the simple theoretical image containing one object of constant brightness with the extremely thin edges; theoretically, the edges can be infinitely thin. In the case that the object appearance should be described by analyzing the function of intensity gradients and directions. The sample values of such a function will be difficult to obtain; theoretically, this is not possible in the case of the infinitely thin edges. Conversely, the information about the area of this object can be described without any difficulties. Suppose that the temperature source is placed into the previously mentioned object with extremely thin edges, and suppose that the transfer of temperature can be solved by making use of physical laws inside the image; the thermal conductivity properties are determined by the gradient of brightness (high gradients indicate the low conductivity and vice versa). After the temperature transfer that is carried out during a certain chosen time, the area of this object will contain a certain distribution of energy. The function values of this distribution are approximately constant inside the object area. The information about the object area (that also contains the information about the shape of object) can then be simply obtained by sampling, and it can be used for the recognition.

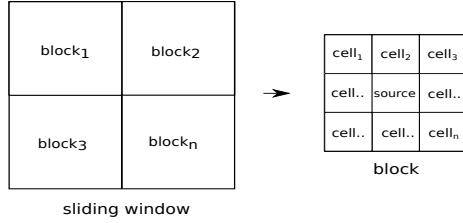


Fig. 2. The blocks and cells inside the sliding window

It is clear that in the real images (Fig. 1(a)), the situation is more complicated, but the previous assumption can be extended also for these images. In the real images, the objects of interest consist of many areas and one temperature source will not be enough to cover all areas. Accordingly, suppose the locations of the sources in the form of a regular grid inside the image (Fig. 1(b)). The temperature transfer starts at the time $t = 0$ from all sources at once. The temperature of sources equals 1 for all $t > 0$. After the temperature transfer process inside the image (which is stopped at a suitable chosen time), the temperature distribution will reflect the areas of objects and also the shape of objects (Fig. 1(c)). After this process, the sample values from the distribution function can be used for recognition.

In the process of extracting the proposed features, the image inside the sliding window is divided into the regular blocks (Fig. 2). We use the gravity centers of these blocks as the places in which we put the temperature sources. For the purpose of obtaining the distribution of temperature, the blocks are divided into the small connected cells (Fig. 2).

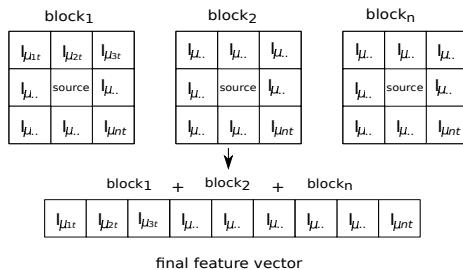


Fig. 3. The feature vector that is composed of the mean temperatures of cells

Let $I(x, y, t)$ be a value of temperature at a time t and at a position (x, y) . Inside each cell, the mean temperature $I\mu_{it}$ of the i -th cell at a time t can be calculated. The final feature vector is composed of these mean values (Fig. 3). We note that the temperature transfer is computed in the whole image inside the sliding window and temperature transferred from one source can influence every cell inside the sliding window; the blocks and cells are formed only for distribution measurement.

For the practical realization of the method, it is important to mention that the thermal field over one position of sliding window can be solved by making use of the following equation [11]

$$\frac{\partial I(x, y, t)}{\partial t} = \operatorname{div}(c \nabla I), \quad (1)$$

where I represents the temperature at a position (x, y) and at a time t , div is a divergence operator, ∇I is the temperature gradient and c stands for thermal conductivity. For the source points and arbitrary time $t \in [0, \infty)$, we set $I(x_s, y_s, t) = 1$, where (x_s, y_s) are the coordinates of source points (i.e. we hold the temperature constant during the whole process of transfer, which is in contrast with the usual diffusion approaches). In all remaining points, we take into account the initial condition $I(x, y, 0) = 0$. We solve the equation iteratively. The conductivity in Eq. 1 is determined by

$$c = g(\|E\|), \quad (2)$$

where E is an edge estimate. We define the edge estimate E as the gradient of original image $E = \nabla B$, where B is the brightness function. The function $g(\cdot)$ has the form of [11]

$$g(\|\nabla B\|) = \frac{1}{1 + \left(\frac{\|\nabla B\|}{K}\right)^2}, \quad (3)$$

where K is a constant representing the sensitivity to the edges [11]. Once the temperature field over the input image (inside the sliding window) is obtained (at a chosen time t), the mean cell temperature $I\mu_{it}$ can be obtained by making use of the formula

$$I\mu_{it} = \frac{\iint_M I(x, y, t) dx dy}{|M|}, \quad (4)$$

where M stands for the cell area, and $|M|$ is its size.

In the next step, the SVM classifier is trained over the proposed descriptors. Let us consider a training data set (x_i, y_i) where x is the vector of proposed descriptors from training samples and y is the class label (+1 for pedestrian, -1 for non-pedestrian). The linear SVM determine hyperplane $w \cdot x + b$ where w is a weight vector, x is the vector of features and b is a constant. The goal is to find the optimal decision function that maximizes the distance between the nearest point x_i and the hyperplane. In the case when it is difficult separate examples in a linear manner, the non-linear SVM can be used. The non-linear SVM maps the original space in a high-dimensional space using a kernel function that separate training samples. The optimal hyperplane for non-linear SVM is obtained by the function $f(x)$:

$$f(x) = \sum_{i=0}^N y_i \alpha_i k(x, x_i) + b, \quad (5)$$

where N represents the number of training patterns, y_i is a class indicator (+1 for pedestrian, -1 for non-pedestrian) for each training pattern x_i , α_i and b are learned weights and $k(., .)$ is a kernel function. In our case, we use Gaussian radial basis function kernel:

$$k(x, y) = e^{\frac{|x-y|^2}{2\sigma^2}}. \quad (6)$$

4 Experiments

We collected 2500 positive samples and 10000 negative samples for the training phase. For the positive set, we combined the pedestrian images from the CBCL Pedestrian Database [12] with the images from the Daimler benchmark [13]. For the negative images, the examples were randomly sampled from the INRIA Person Dataset [1]. For the proposed method, each sample was resized to the size of 91×151 pixels. The visualization of the proposed features of positive samples is shown in the Fig. 5 (the parameters will be discussed later). The size of sliding window was set to the size of training samples. In the detection phase, we created the different resolutions of input image in which the sliding window was moving across these images.

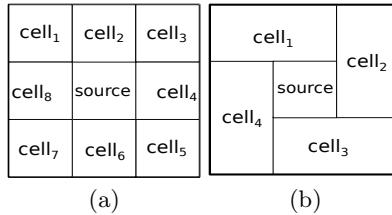


Fig. 4. The design of every block in the *Energy*₄₈₀ configuration (a). The design of every block in the *Energy*₂₄₀ configuration (b).

We experimented with the parameters of proposed features and we suggested the two optimal following configurations: *Energy*₄₈₀, *Energy*₂₄₀. The *Energy*₄₈₀ configuration was designed with the size of block = 15×15 pixels; the size of cells = 5×5 pixels (the number of cells inside each block = 8); the size of sources = 5×5 pixels; the time for the temperature transfer = 100 (the number of iterations). This configuration consists of 480 descriptors for one position of sliding window. The design of each block of the *Energy*₄₈₀ configuration is shown in Fig. 4(a). The *Energy*₂₄₀ configuration was designed with the size of block = 15×15 pixels; the size of cells = 10×5 and 5×10 pixels (the number of cells inside each block = 4); the size of sources = 5×5 pixels; the time for the temperature transfer = 100 (the number of iterations). This configuration consists of 240 descriptors for one position of sliding window. The design of each block of the *Energy*₂₄₀ configuration is shown in Fig. 4(b).



Fig. 5. The visualization of proposed descriptors of pedestrian images. The value of temperature is depicted by the level of brightness. The features are designed with the following parameters: the size of blocks 15×15 ; the time $t = 150$ (the number of iterations for the transfer of temperature), the size of temperature sources $= 5 \times 5$.

For the comparison, we designed the two configurations of classical HOG features HOG_{336} , HOG_{3780} . We use the classical version of HOG descriptors without the extensions that were mentioned in Section 3 (e.g. PCA, AdaBoost, LBP) due to the fact that the proposed features are also presented without these extensions. We note that some of these extensions can be also used in the proposed features in the future works. The training sets for the HOG features and proposed features were identical (2500 positive and 10000 negative samples). For the HOG descriptors, each training sample was resized to the size of 64×128 pixels. The HOG_{336} configuration was designed with the similar number of descriptors like in the proposed configurations. The parameters were as follows; the size of block $= 32 \times 32$ pixels; the size of cell $= 16 \times 16$ pixels, the horizontal step size $= 16$ pixels; the number of bins $= 4$. This configuration consists of 336 HOG descriptors. The HOG_{3780} configuration was designed with the typical parameters of HOG descriptors; the size of block $= 16 \times 16$ pixels; the size of cell $= 8 \times 8$ pixels; the horizontal step size $= 8$ pixels; the number of bins $= 9$. This configuration consists of 3780 HOG descriptors. For the testing, we collected 55 images from the [1]. The detection results are shown in Table 1.

The worst detection results were acquired with the HOG_{336} configuration with the 336 HOG descriptors. The high numbers of false positive detections were visible in this configuration compared with the proposed method. This negative effect was caused by the small dimensionality of feature vector of the HOG configuration. Many significant object details cannot be precisely described with the size of blocks and cells of the HOG_{336} configuration. The 336 HOG descriptors were not able to successfully distinguish between the positive and negative samples and the detector based on this configuration detected human in the images in which the people were not visible; for example, the HOG_{336} configuration detected the traffic signs like pedestrians (Fig. 6).

Table 1. The detection performance of proposed features and the features that are based on HOG

	Precision	Sensitivity	F1 score
<i>Energy</i> 240	85.29%	77.33%	81.12%
<i>Energy</i> 480	87.14%	84.72%	86.56%
<i>HOG</i> ₃₃₆	51.72%	85.71%	64.86%
<i>HOG</i> ₃₇₈₀	86.97%	81.97%	84.03%

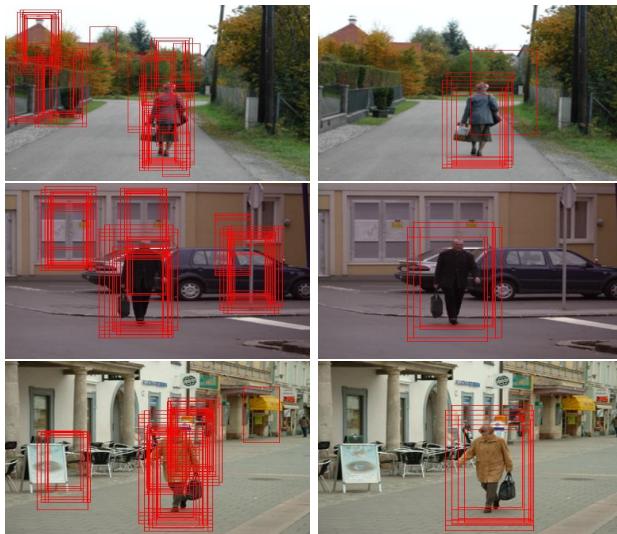


Fig. 6. The example of detection. The left images show the detection results of HOG_{336} configuration. The right images show the results of $Energy_{240}$ configuration. The detection results of approaches are shown without the post-processing (the detection results are not merged).

On the other hand, the $Energy_{240}$ configuration of the proposed method (with the relatively small set of descriptors = 240) was able to successfully describe the appearance of the objects of interest. The $Energy_{240}$ configuration of the proposed method achieved the better results (F1 score 81.12%) than the HOG_{336} configuration (F1 score 64.86%). The $Energy_{240}$ configuration detected the objects of interest with the very promising detection rates. We tried to increase these rates by creating the second configuration with more descriptors. As we show in the results, the second proposed configuration ($Energy_{480}$) achieved the best detection rate (F1 score 86.56%). The detector that was based on this configuration successfully detected most of the pedestrians with the 480 descriptors. Compared with the HOG_{3780} configuration (F1 score 84.03%), the proposed features achieved the similar results, however the proposed method gives 7× less descriptors than the classical HOG_{3780} configuration.

Finally, the *Energy₄₈₀* configuration shows that the pedestrians can be efficiently encoded with the reasonable dimensionality of feature vector without need for the methods for reducing the feature space. The detection results of the *Energy₄₈₀* configuration are shown in Fig. 7.

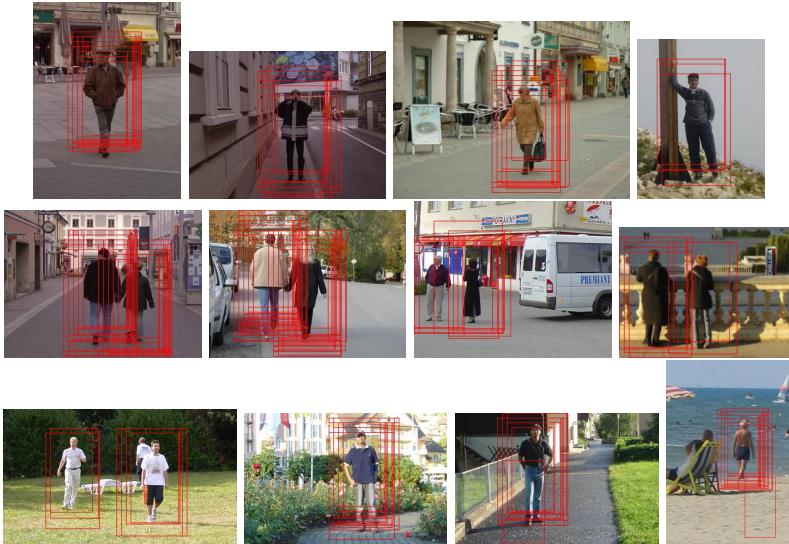


Fig. 7. The detection results of the *Energy₄₈₀* configuration without the post-processing (the detection results are not merged)

5 Conclusion

In this paper, we proposed the efficient method for the computation of image features. The proposed features are based on the energy distribution. Using this distribution, the appearance of objects can be effectively described in the sense of dimensionality of the feature vector. The detection results that were achieved with this dimensionality are very promising for the future works in which we will focus on the detection of other objects of interest (faces, cars) and we will also focus on the time complexity of computation of the proposed features.

Acknowledgments. This work was supported by the SGS in VSB Technical University of Ostrava, Czech Republic, under the grant No. SP2013/185.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893 (2005)

2. Suard, F., Rakotomamonjy, A., Bensrhair, A., Broggi, A.: Pedestrian detection using infrared images and histograms of oriented gradients. In: 2006 IEEE Intelligent Vehicles Symposium, pp. 206–212 (2006)
3. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1491–1498 (2006)
4. Kobayashi, T., Hidaka, A., Kurita, T.: Selection of histograms of oriented gradients features for pedestrian detection. In: Ishikawa, M., Doya, K., Miyamoto, H., Yamakawa, T. (eds.) ICONIP 2007, Part II. LNCS, vol. 4985, pp. 598–607. Springer, Heidelberg (2008)
5. Cao, X., Wu, C., Yan, P., Li, X.: Linear svm classification using boosting hog features for vehicle detection in low-altitude airborne videos. In: 2011 18th IEEE International Conference on Image Processing, ICIP, pp. 2421–2424 (2011)
6. Chuang, C.H., Huang, S.S., Fu, L.C., Hsiao, P.Y.: Monocular multi-human detection using augmented histograms of oriented gradients. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4 (2008)
7. Wang, X., Han, T., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 32–39 (2009)
8. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007, pp. 401–408. ACM, New York (2007)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I-511–I-518 (2001)
10. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 734–741 (2003)
11. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 629–639 (1990)
12. Center for Biological and Computational Learning: MIT CBCL Pedestrian Database #1 (2013),
<http://cbcl.mit.edu/software-datasets/PedestrianData.html>
13. Enzweiler, M., Gavrila, D.: Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 2179–2195 (2009)

Basic Shape Classification Using Spatially Normalised Fourier Shape Signature

Chin Yeow Wong, Stephen Ching-Feng Lin,
Guannan Jiang, and Ngai Ming Kwok

School of Mechanical and Manufacturing Engineering,
The University of New South Wales,
Sydney, NSW, 2052, Australia

{chin.wong,stephen.lin,guannan.jiang,nmkwok}@unsw.edu.au

Abstract. Fourier Descriptors (FD) generation depends heavily on the input shape signature and is a core component in traditional Content Based Image Retrieval (CBIR) systems. This paper presents a novel basic shape classifier developed using Complex Coordinates (CC) FD. A spatial domain normalisation of the FD is achieved by overlaying a Fourier Synthesised Boundary (FSB) against its original Boundary Points (BP). This process creates Intersection Points (IP). A new shape signature is formed using a ratio representing the number of IP over the number of BP. The shape signature coined as Spatially Normalised Fourier Shape Signature (SNFSS), varies from 0 to 1 with increasing number of FD used, and exhibit key trends for the detection of basic shapes like circle and regular polygons. The trends are proven experimentally to be invariant to scale and rotation, as well as being robust to noise.

1 Introduction

Content Based Image Retrieval has become an active and challenging research area because the exponential increase in storage space and ease of creating images has led to larger image database. Retrieving a queried image from such a dataset would require improvements in accuracy and computational speed. Signature are extracted from query images and compared against a library of signature before the best match image is retrieved. Figure 1 shows an example of the query-retrieval process using an equilateral triangle.

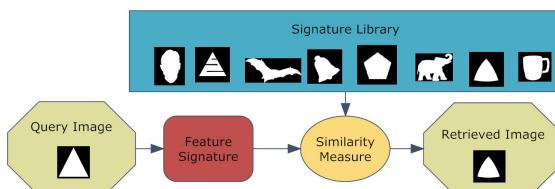


Fig. 1. Block diagram representation of processes in a CBIR system

Objects resembling basic shapes like circles and regular polygons appear frequently in image databases [1]. Being able to extract these shapes accurately and quickly is useful for applications like road sign recognition [2, 3] or nano particle sorting [4, 5]. Probabilistic or iterative approaches can achieve good accuracy but are computationally expensive because mathematical models with varying modeling and environmental constraints must be derived and adhered to. In contrast, classifying images based on visual contents, such as color data, texture cues, and shape information, are simpler to implement when available, but lacks robustness. Among the three visual content classes, shape information is a natural and stable classifier for images because cognitive studies show that users are more prone to index image datasets based on its shape information [6].

There are generally two types of shape descriptors in literature, namely, contour-based or region-based [1, 7]. Region-based techniques take into account all pixels within a shape region and as such, have the advantage of being able to describe shapes with disjointed boundaries. It can also capture the interior of a shape but at the expense of more computational resources and a reduction in emphasis on contour features, which are vital for human perception of shapes. Contour-based techniques, on the other hand, are simpler to acquire and are descriptively sufficient for many applications. As such, it is a popular choice for larger datasets. Contour-based methods have two sub categories [8]; Structural approaches divide a contour at key points into segments while conventional approaches treat the contour as a single entity.

Fourier Descriptors (FD) by Kahn et al. [9], are considered attractive shape descriptors because it is based on a sound theoretical foundation, have the advantages of being computationally efficient to generate, as well as, have unique invariance properties. FD are generally obtained by applying Fourier Transform (FT) on one, two [10, 11], or even three dimensional [12] functions derived from images and video. A variety of applications like copyright watermarking [13], object and character recognition [14–16], shape interpolation [9], reconstruction [17], retrieval [7, 12] and classification [3, 5, 11, 18–20], rely on the generation of FD.

The FD are proven in literature to help improve retrieval rates for both the contour-based and region-based class of shape descriptors. The introduction of generic FD on polar raster sampled images [10], followed by Wavelet FD [7] on Centroid Distance (CD) signature, led to breakthrough achievements in retrieval rates for region-based class, exceeding the performance of moment, angular radial transform and grid descriptors. For the case of contour-based methods, Farthest Point FD [19] outperforms commonly used methods such as Curvature Scale Space Descriptors (CSSD) and AutoRegressive (AR) methods [8]. The CSSD does not capture global features and matching is too complex, while AR suffers from poor initialisation and does not carry physical meaning. Pseudo-Fourier methods such as the Statistical FD [20] and Invariant Curvature FD [11] are gaining popularity because of the adaptable implementation of FD.

The difference in FD generation between contour-based and region-based techniques rests in its dimensionality with the former being of order one and latter being of order two. A common technique to achieve starting point, translation, rotation and scale invariance is to use magnitudes of the FT, and ignore the phase information [7, 18]. This is proven to be insufficient to describe a shape [21]. The introduction of Dynamics Time Warping distance as a similarity metric instead of traditional Euclidean Distance [11] allowed phase information to be exploited for matching [22] and has been improved since.

In our work, we performed an indirect normalisation of the magnitude and phase information to produce a new basic shape detector capable of identifying circles and regular polygons. The basis of our research is closely related to structural modeling of contours. While segments have been used as a structural feature to describe shapes [23, 24], the key points used to split segments are largely ignored, as they are usually generated manually. Instead of following a set of predefined rules, FD are chosen to create key points adaptively on a shape to be analysed.

This paper is presented in the following order: In Section 2, the proposed modification to traditional FD generation and manipulation is introduced along with SNFSS. This is followed by Section 3, where we describe three experiments devised to highlight the features of SNFSS for shape detection. The experimental results and associated discussions are given in Section 4. Finally, in Section 5, we conclude the paper by providing a future direction for our research.

2 Proposed Modification

Traditionally, preprocessing is applied to produce BP from images. Fourier Descriptors are then generated using FT. For shape classification, a normalisation procedure is required before similarity measures are evaluated. In our work, we replace the last two blocks with Inverse Fourier Transform (IFT) and superposition (as shown in Figure 2) to form a Spatially Normalised Fourier Shape Signature (SNFSS) for our circle and regular polygon detector.

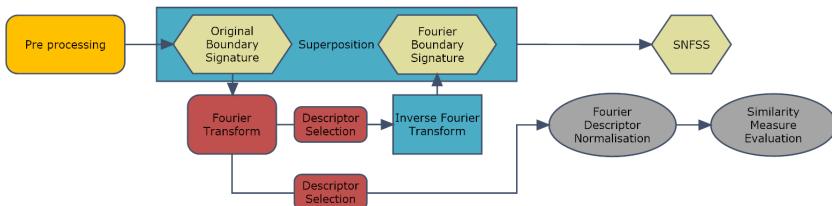


Fig. 2. As a modification, traditional blocks (ellipses) are replaced by the rectangular blocks to form the Spatially Normalised Fourier Shape Signature

2.1 Pre-processing

Pre-processing involves performing Canny edge detection [25] on an image. A boundary tracing algorithm [26] that operates in a clockwise fashion is then used to attain the x and y coordinates of each object in that image. These boundaries must be closed to ensure the signal continuity criteria is met. A sequence of boundary coordinates z , can be represented by an array

$$z(n) = [x(n), y(n)] \text{ for } n = 0, 1, 2, \dots, N - 1, \quad (1)$$

where N represents the total number of BP detected. To ensure that the periodicity criteria is met, equal arc-length (constant velocity) sampling is required. This can be achieved by selecting n in such a way that the sampling rate, $t = n_i - n_{i-1}$ is constant. The Nyquist-Shannon sampling theorem is automatically satisfied by selecting the highest sampling rate of $t = 1$. Thus, aliasing is avoided.

2.2 Shape Signature Selection

Two or more dimensional signatures are non-injective, or have many to one mapping. Thus, region-based FD are ruled out for this task. Many 1-D signatures are proposed in literature [19, 27], such as the centroid distance, the angular function, the chord length, boundary area, complex coordinates, polar representation, triangular centroid area, triangular area representation, angular radial coordinates, and farthest point distance. Although the farthest point distance [19] is the best in its class for image retrieval, it is deemed not suitable for our work because a loss in dimensionality generally occurs when distances or areas are used, making the original boundary irrecoverable through IFT. An exception is the Centroid Distance (CD) signature [7].

A function is said to be bijective when every element of the codomain is mapped to exactly one element of the domain. The distinctiveness of a signature is preserved when the real parts and imaginary parts of an input signature are bijective to its respective complex output components. Polar representation is bijective, but the filtering of phase information forces the shape to be irrecoverable through IFT. A more suitable bijective signature is the Complex Coordinates (CC) signature because the real and imaginary parts of the synthesised signature return boundary coordinate values.

Base on the above justifications, we select the CD and the CC signature. For both signatures, we first offset the x , y coordinate pairs of a boundary off their respective means, x_m , y_m , forming X and Y :

$$X = x - x_m \text{ and } Y = y - y_m. \quad (2)$$

The CD signature z_{cd} only has real components and is given as the Euclidean distance of each point from the mean:

$$z_{cd}(n) = \sqrt{X(n)^2 + Y(n)^2} \text{ for } n = 0, 1, 2, \dots, N - 1. \quad (3)$$

For the CC signature z_{cc} , the real and imaginary parts are defined directly as mean shifted coordinates:

$$z_{cc}(n) = (X(n)) + j(Y(n)) \text{ for } n = 0, 1, 2, \dots, N - 1. \quad (4)$$

z_{cd} and z_{cc} are subsets of the selected signature, z_s .

2.3 Fourier Descriptor Generation and Usage

The final step to generating FD is to apply DFT on the shape boundary:

$$Z(k) = \sum_{n=0}^{N-1} z_s(n) e^{-j \frac{2\pi k n}{N}} \text{ for } k = 0, 1, 2, \dots, N - 1. \quad (5)$$

Fast Fourier Transform (FFT) is a common method used to perform the above computation, and is similarly used in our work. The complex coefficients, $Z(k)$ are known as the FD. Normalisation is done by dividing through the magnitudes of each complex coefficient $|Z(k)|$ for $k = 0, 1, 2, \dots, N - 1$ with its highest magnitude value or the first coefficient $|Z(\frac{(N-1)}{2} + 1)|$. An object similar to that in a library can only be retrieved once a similarity measure is evaluated.

2.4 Spatially Normalised Fourier Shape Signature

Instead of using normalised FD, boundaries are recovered using IFT and are superimposed upon the original BP. This spatial domain process is analogous to applying normalisation towards the magnitude and phase at the same time. The IDFT that restores $z_r(n)$ is given by:

$$z_r(n) = \frac{1}{N} \sum_{k=0}^{N-1} Z(k) e^{j \frac{2\pi k n}{N}} \text{ for } n = 0, 1, 2, \dots, N - 1. \quad (6)$$

Similarly, Inverse Fast Fourier Transform (IFFT) is applied. The choice on the number of descriptors P for reconstruction can affect the boundary as follows:

$$\hat{z}_P(n) = \frac{1}{P} \sum_{k=0}^{P-1} Z(k) e^{j \frac{2\pi k n}{N}} \text{ for } n = 0, 1, 2, \dots, N - 1. \quad (7)$$

Generally, $P/2$ descriptors are sufficient to recover the entire shape due to its symmetric property. When P is set as 2, the $k + 1$ th and $k - 1$ th term from midpoint $k = (N - 1)/2$ are used to synthesize its Fourier boundary.

The equivalent expression for the generation of IP, $w = [x, y]$ through the superposition of the BP and FSB recovered with P descriptors is:

$$w_P = \text{round}(\hat{z}_P) \cap z \text{ where } w \subset z. \quad (8)$$

Computationally, the two lists of boundary coordinates, \hat{z}_P and z are cycled through and compared. When their x and y values are successfully matched,

these coordinates are recorded and a counter M_P gets incremented. Finally, a ratio that parametrises the shape boundary is defined below as:

$$R_P = \frac{M_P}{N}. \quad (9)$$

The corresponding ratio, $R = \{R_1, R_2, R_3, \dots, R_N\}$ is proportional to the number of IP, $M = \{M_1, M_2, M_3, \dots, M_N\}$ produced, and increases from 0 to 1 when more number of descriptors, P are used. The number of BP, N however, remains constant. This new shape signature, R is coined as the SNFSS.

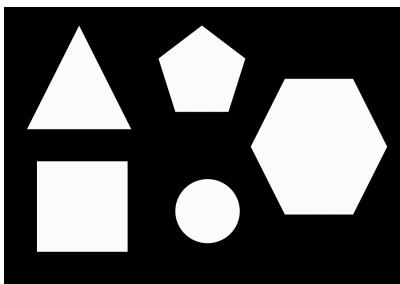
3 Experimental Setup

We examine the properties exhibited by SNFSS through a series of experiments.

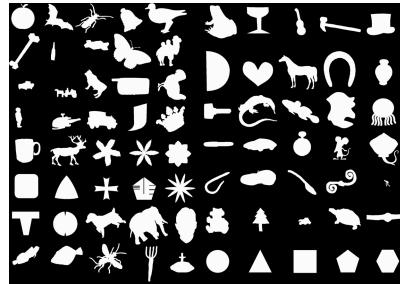
Experiment I : The image shown in Figure 3a consists of a circle $q = \infty$ and four q-sided regular polygons $q = [3, 4, 5, 6]$ of different scales. It is used to identify features of SNFSS for circle and regular polygon detection.

Experiment II : In this experiment, we test the robustness of SNFSS features found in Experiment I. The initial step is to observe the effects of adding Gaussian noise, with 0 mean and 0.25 to 2 standard deviation, to the original shape boundary. To test for scale invariance, each of the aforementioned shapes are enlarged four times, at 25% increments. Similarly, to test for rotational invariance, the shapes are individually rotated clockwise four times at increments of 30°.

Experiment III : Figure 3b shows an image of a circle, an equilateral triangle, a square, a pentagon and a hexagon amidst 70 other objects extracted from the MPEG7_CE-Shape-1_Part_B database [28]. It is used to test the robustness of the SNFSS features for shape-based object retrieval.



(a) Test shapes for Experiment I.



(b) Test shapes for Experiment III.

Fig. 3. Images used to examine the properties of SNFSS

4 Results and Discussion

The results from the experiments previously described in Section 3 are given in the following subsections:

4.1 Circle and Regular Polygon Detection

A circle is the simplest shape described only by its fundamental frequency. All higher frequencies are suppressed due to the orthogonality of complex exponential function. Therefore, the convergence of the circle SNFSS to $R = 1$ occurring at low values of P , in Figure 4a and 4b is justified. When CC shape signature (Figure 4b) is used as an input, a distinct crest can be identified at P_8 for a circle. In contrast, when CD signature (Figure 4a) is used, modulating its frequency content does not result in a drop of R values.

The drop is contributed by both the magnitude and phase component of the CC signal shown in Figure 5. The inclusion of Fourier coefficients from -4 to 4 makes the FSB trace the BP in a sinusoidal fashion, causing a decline in the number of IP generated. The repeating patterns at P_{16} , P_{24} are due to its harmonic contents. Observe that in Figure 4b that for regular polygon shapes,

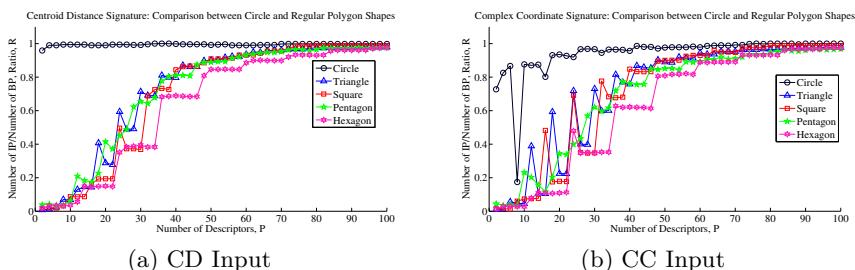


Fig. 4. Plot of SNFSS from P_2 to P_{100} for all five objects in Figure 3a

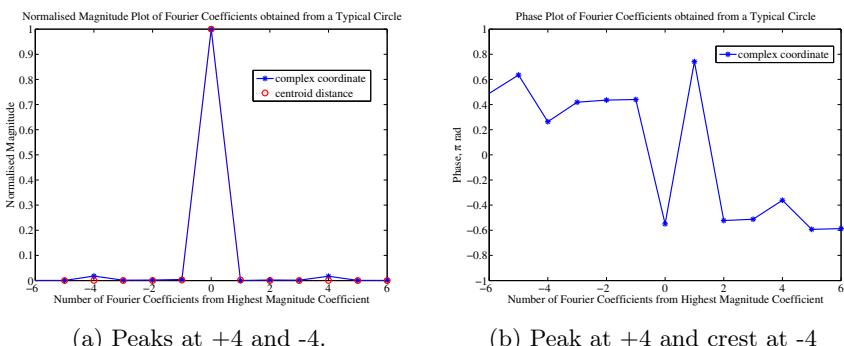


Fig. 5. The contributions of magnitude and phase results in a drop in IP numbers

more prominent peaks (Triangle, P_{12} , Square, P_{16} , Pentagon, P_{10} , Hexagon P_{24}) can be found as compared to Figure 4a. To form a q -sided regular polygon detector, we utilise the fact that there are no significant jumps in R values from P_2 to P_{q-1} as an initial guess before verifying it against the expected key peaks.

4.2 Robustness to Noise, Scale, and Rotational Variance

Figure 6a highlights the IP generation process through the use of spatial normalisation. By looking at the graphs in Figure 6b, we found that the key features (flatness from P_2 to P_4 and peak at P_{12}) remain prominent with additive boundary noise level of up to one standard deviation.

Varying the size of a shape affects the rate at which SNFSS increases. Figure 7a illustrates that the flatness in P_2 to P_8 values discussed in Section 4.1 remains available for shape classification. The key peak at P_{10} however, is evidently less prominent at lower scales. The same scale invariant trend is observed for all other regular polygons. We can observe from Figure 7b that when a square is rotated, its SNFSS properties remain unchanged. The flatness in R values last

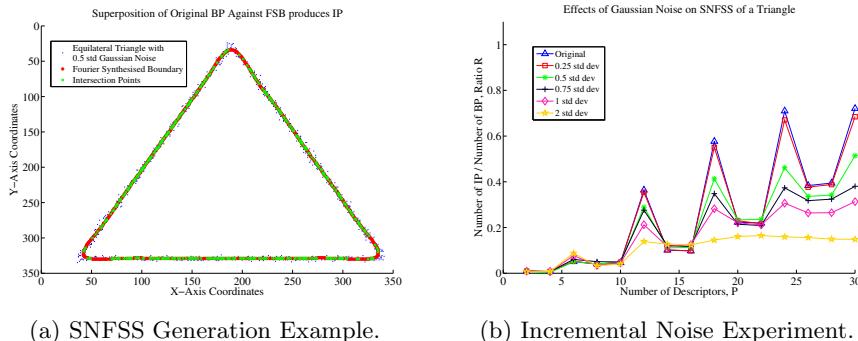


Fig. 6. Affect of additive noise on SNFSS

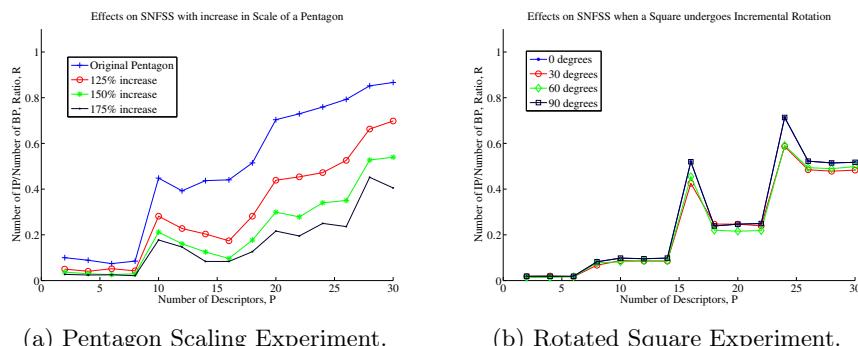


Fig. 7. Extracts of results from Experiment II

from P_2 to P_6 and the key peak remains prominent at P_{16} . With the exception of an equilateral triangle, all other shapes do not experience a shift in peak values.

4.3 Robustness for Shape Classification

In our work, we use the binary classification test to record our results. Out of the 75 objects in Figure 3b, one was classified as a circle, four as a triangle, three as a square, two as a pentagon and four as a hexagon as shown in Figure 8. The exact classification rates for this experiment are shown in Table 1.

Table 1. Classification Rates

Shape	Manual Count	True +	True -	False +	False -	Recall	Precision
Circle	1	1	74	0	0	1	1
Triangle	2	2	71	2	0	1	0.5
Square	2	1	72	2	1	0.5	0.33
Pentagon	2	1	73	1	1	0.5	0.5
Hexagon	1	1	71	3	0	1	0.25

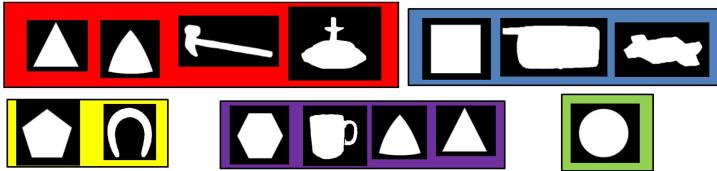


Fig. 8. Retrieved images sorted based on shape information

When utilising a circle, triangle, square, and pentagon as the query object, classification results were satisfactory from a visual perspective. The hexagon query object however, produces two false-positive detection of triangle like objects. This is because repeating harmonics of P_{12} at P_{24} were misinterpreted as the prominent feature peak. Additional pre-screening rules can instead be imposed to future systems.

5 Conclusion

In this paper, we perform spatial normalisation on Fourier Descriptors to produce a circle and regular polygon detector that is simple to implement. Complex Coordinate FD is proven to be better than Centroid Distance FD for basic shape classification because it produces distinct flatness feature, peaks and crests. For preliminary detection of n -sided polygons, only $P \times 2n$ amount of FD needs to be evaluated. Experimental results show that the newly introduced Spatially Normalised Fourier Shape Signature is robust against noise, scale, and rotation

variance. Furthermore, when implemented on a content based image retrieval system, the possibility of a shape-based sorting application becomes evident.

References

1. Zhang, D., Lu, G.: Review of shape representation and description techniques. *Pattern Recognition* 37, 1–19 (2004)
2. Barnes, N., Loy, G., Shaw, D.: The regular polygon detector. *Pattern Recognition* 43, 592–602 (2010)
3. Larsson, F., Felsberg, M., Forssén, P.E.: Correlating fourier descriptors of local patches for road sign recognition. *IET Computer Vision* 5, 244–254 (2011)
4. Wong, C., Lin, S., Ren, T., Kwok, N.: A survey on ellipse detection methods. In: 2012 IEEE International Symposium on Industrial Electronics, ISIE, pp. 1105–1110. IEEE (2012)
5. Rice, K.P., Saunders, A.E., Stoykovich, M.P.: Classifying the shape of colloidal nanocrystals by complex fourier descriptor analysis. *Crystal Growth & Design* 12, 825–831 (2012)
6. Mumford, D.: Mathematical theories of shape: Do they model perception? In: San Diego 1991, San Diego, CA, International Society for Optics and Photonics, pp. 2–10 (1991)
7. Yadav, R.B., Nishchal, N.K., Gupta, A.K., Rastogi, V.K.: Retrieval and classification of shape-based objects using fourier, generic fourier, and wavelet-fourier descriptors technique: A comparative study. *Optics and Lasers in Engineering* 45, 695–708 (2007)
8. Zhang, D., Lu, G.: Evaluation of mpeg-7 shape descriptors against other shape descriptors. *Multimedia Systems* 9, 15–30 (2003)
9. Zahn, C.T., Roskies, R.Z.: Fourier descriptors for plane closed curves. *IEEE Transactions on Computers* 100, 269–281 (1972)
10. Zhang, D., Lu, G.: Generic fourier descriptor for shape-based image retrieval. In: Proceedings of the 2002 IEEE International Conference on Multimedia and Expo, ICME 2002, vol. 1, pp. 425–428. IEEE (2002)
11. El-ghazal, A., Basir, O., Belkasim, S.: Invariant curvature-based fourier shape descriptors. *Journal of Visual Communication and Image Representation* 23, 622–633 (2012)
12. Vranić, D.V., Saupe, D.: 3d shape descriptor based on 3d fourier transform. In: Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services, ECMCS 2001, Budapest, Hungary (2001)
13. Solachidis, V., Pitas, I.: Watermarking polygonal lines using fourier descriptors. *IEEE Computer Graphics and Applications* 24, 44–51 (2004)
14. Granlund, G.H.: Fourier preprocessing for hand print character recognition. *IEEE Transactions on Computers* 100, 195–201 (1972)
15. Personn, E., Fu, K.S.: Shape discrimination using fourier descriptors. *IEEE Transactions on Systems, Man and Cybernetics* 7, 170–179 (1977)
16. González, E., Adán, A., Feliú, V., Sánchez, L.: Active object recognition based on fourier descriptors clustering. *Pattern Recognition Letters* 29, 1060–1071 (2008)
17. Lin, C., Chellappa, R.: Classification of partial 2-d shapes using fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 686–690 (1987)
18. Kunttu, I., Lepisto, L., Rauhamaa, J., Visa, A.: Multiscale fourier descriptor for shape classification. In: Proceedings of the 12th International Conference on Image Analysis and Processing, pp. 536–541 (2003)

19. El-ghazal, A., Basir, O., Belkasim, S.: Farthest point distance: A new shape signature for fourier descriptors. *Signal Processing: Image Communication* 24, 572–586 (2009)
20. Timm, F., Martinetz, T.: Statistical fourier descriptors for defect image classification. In: 2010 20th International Conference on Pattern Recognition, ICPR, pp. 4190–4193. IEEE (2010)
21. Crimmins, T.R.: A complete set of fourier descriptors for two-dimensional shapes. *IEEE Transactions on Systems, Man and Cybernetics* 12, 848–855 (1982)
22. Bartolini, I., Ciaccia, P., Patella, M.: Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 142–147 (2005)
23. Gorman, J.W., Mitchell, O.R., Kuhl, F.P.: Partial shape recognition using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, 257–266 (1988)
24. Bai, X., Yang, X., Latecki, L.J.: Detection and recognition of contour parts based on shape similarity. *Pattern Recognition* 41, 2189–2199 (2008)
25. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 679–698 (1986)
26. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital image processing using MATLAB, vol. 2. Gatesmark Publishing Tennessee (2009)
27. Zhang, D., Lu, G.: Study and evaluation of different fourier methods for image retrieval. *Image and Vision Computing* 23, 33–49 (2005)
28. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape descriptors for non-rigid shapes with a single closed contour. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 424–429. IEEE (2000)

Normalized Matting of Interest Region

Jaehwan Kim and Ilkwon Jeong

Creative Content Research Lab., ETRI, Korea
{jh.kim,jik}@etri.re.kr

Abstract. In this paper, we present an improved method of current closed-form solution for digital image matting. This method, which we call ‘normalized matting of interest region’, adopt the normalized cut technique where the objective function is normalized with the total degree of color similarities of foreground region. Unlike the existing solution, our method measures both the total dissimilarity between the foreground and background regions as well as the total similarity within foreground regions, which leads to better separation results, especially in case of extracting a specific region, rather than the closed-form solution. In addition, we employ a quadratic programming approach to solve the objective function to obtain a globally near-optimal matting result. Our method is empirically verified through several sample images.

1 Introduction

Image matting, whose goal is to extract only an interesting foreground component from an arbitrary and natural image, plays an important role in a variety of areas such as computer vision and graphics. Especially, digital image matting has been widely used in film production lately in order to provide an efficient way of tackling a complicated composition [1], [2], [3], [4]. However, it is hard to generate a perfect matte from given image without having any prior information since the matting problem is intrinsically ill-posed. This prior information is usually fed by means of trimap or scribbles. The trimap in which pixels are marked as definitely belonging to the foreground, background, or unknown areas, is a rough segmentation of given image. Given a complex image containing lots of holes and thin components like smoke, fur, or hair, it has difficulty in making a precise trimap. A trimap defined loosely results in poor alpha mattes directly. In order to overcome the problem a trimap based approach has, a scribble-based matting method has been introduced [4],[2]. Scribbles which are kinds of brushes for specifying foreground and background, provide user with more flexible interaction than a trimap-based way. Despite the fact that the scribble-based approach complements weak points of the trimap-based matting methods, the precise matte result can be obtained only when the user specifies enough scribbles or constraints on the input image. When user applies the scribble-based matte method to image and video editing problems, the requirements for specifying sufficient scribbles on the input image can be a serious drawback definitely.

Hence, the closed-form solution to image matting was recently proposed [2] as a scribble-based matting method. Levin *et al.* contribute toward formulating a well-defined quadratic cost function in foreground opacity from a severely underconstrained matting problem. It was also shown in [2],[3] that the closed-form solution to image matting is closely related to spectral clustering. However, the method has a limitation that small windows on a fine resolution image cause erroneous alpha mattes.

In this paper, we present a new criterion for digital image matting problems, referred to as *normalized matting of interest region*, which is derived in the way of incorporating a normalized term regarding foreground region into the current closed-form solution's objective. By taking the normalized factor into account, we can not only minimize the sum of weights of connections across the different foreground and background regions, but also maximize the cohesion of alpha matte within the foreground region, which helps the foreground constraint to propagate to the only foreground-like color region strictly. We also employ a quadratic programming which theoretically guarantees a globally near-optimal solution to obtain good matte results. Through some experiments with toy and real images, we confirm the useful behavior of the proposed method.

2 Normalized Matting of Interest Region

We begin with revisiting the closed-form solution to image matting. Then, we illustrate how to define the normalized matting algorithm as well as how to relax it into a quadratic programming.

2.1 Closed-Form Solution to Image Matting

The matting problem is to estimate the opacity, called the alpha matte, under an assumption of a linear combination of the corresponding foreground and background colors, which are related each other by the following equation:

$$\mathbf{c}_i = \alpha_i \mathbf{f}_i + (1 - \alpha_i) \mathbf{b}_i, \quad (1)$$

where $\alpha_i \in [0, 1]$ represents the pixel's foreground opacity; $\mathbf{c}_i = [c_{1i} \cdots c_{ni}]^\top \in \mathbb{R}^n$, the color vector; $\mathbf{f}_i = [f_{1i} \cdots f_{ni}]^\top \in \mathbb{R}^n$ and $\mathbf{b}_i = [b_{1i} \cdots b_{ni}]^\top \in \mathbb{R}^n$ (in here, n denotes the number of color channels) represent the foreground and background color vectors at i th pixel, respectively.

The closed form solution for the matting problem lead a well-defined quadratic cost function from an underconstrained problem inherently existing in the image matting. In [2], two main assumptions on which Levin *et al.* rely are that foreground and background color elements f_{ji} and b_{ji} are approximately constant over a local window around each pixel, and that each of \mathbf{f}_i and \mathbf{b}_i images is a mixture of colors. In this case α_i can be expressed as a linear function of the local image \mathbf{c}_i :

$$\alpha_i \approx \sum_j a_j c_{ji} + b, \quad \forall i \in w, \quad (2)$$

where $a_j = 1/(f_{ji} - b_{ji})$, $b = -b_{ji}/(f_{ji} - b_{ji})$, w is a local image window, and j indicates color dimensions. This relation provides a foundation on which the above underconstrained problem being in the matting problem can be transformed into a closed form expression of a least square solution of the following cost function:

$$\mathcal{J}(\boldsymbol{\alpha}, a, b) = \sum_{k \in \mathbf{C}} \left\{ \sum_{i \in w_k} (r_i^k)^2 + \epsilon \sum_{j=1}^n (a_j^k)^2 \right\}, \quad (3)$$

where $r_i^k = \alpha_i - \sum_{j=1}^n a_j^k c_{ji} - b^k$, w_k is a small window (in here, usually 3 x 3 windows are used) around pixel k and $\{\mathbf{c}_t\}_{t=1}^N$ is a total image color vector. Levin *et al.* show in [2] that the above function can be rewritten as a quadratic function with respect to α using algebraic manipulation:

$$\mathcal{J}(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \mathbf{L} \boldsymbol{\alpha}, \quad (4)$$

where $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_N]^\top$ and $\mathbf{L} \in \mathbb{R}^{N \times N}$ is a *matting Laplacian*. This Laplacian can also be written as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where \mathbf{A} is referred to as a *matting affinity* [2], [3], [5] and \mathbf{D} is a diagonal matrix which consists of diagonal elements $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. The affinity's element \mathbf{A}_{ij} is

$$\sum_{k|(i,j) \in w_k} \frac{1}{|w_k|} \left\{ 1 + (\mathbf{c}_i - \boldsymbol{\mu}_k)(\boldsymbol{\Sigma}_k + \frac{\epsilon}{|w_k|} \mathbf{I})^{-1} (\mathbf{c}_j - \boldsymbol{\mu}_k) \right\},$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ denote the mean vector and the covariance matrix of color vectors in the k th local window w_k , respectively. Also, ϵ represents a regulation parameter. The objective function (4) is then incorporated with constraints provided by a scribble-based interface: user can use a brush to discriminate between background and foreground pixels through definitely assigning an alpha matte to the two different regions. Finally the constrained the quadratic cost function can be expressed as following:

$$\begin{aligned} & \arg \min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathbf{L} \boldsymbol{\alpha}, \\ \text{s.t. } & \mathbf{C}_s \boldsymbol{\alpha} = \mathbf{h}_s, \\ & 0 \preceq \boldsymbol{\alpha} \preceq 1, \quad \forall s \in \mathcal{S}, \end{aligned} \quad (5)$$

where \mathcal{S} is the group of scribbled (i.e., constrained) pixels, \mathbf{C}_s is a diagonal matrix whose diagonal elements are one for constrained pixels and zero all other pixels, and \mathbf{h}_s is the vector containing predefined alpha values for the constrained pixels and zero for all others. Finally, Levin *et al.* [2], [6] obtain an optimal alpha matte solution through minimizing the Lagrangian of (5).

However, as addressed in [2] the closed-form solution to image matting has a tendency to assign some erroneous non-opaque alpha values to subregions located far away from any constraints when using small windows on a fine resolution image (see Fig.1 and Fig.3).

2.2 Proposed Normalized Matting of Interest Region

Partitioning a graph with its edge weights assigned by pairwise similarities, serves as an important tool for data clustering (e.g., image segmentation). The image matting problem can be considered as the classical binary partitioning (bipartitioning) on the foreground and background regions. Moreover, normalized cuts [5] is a method to deal with a graph partitioning problem in an efficient way, where the criterion for partitioning the graph will be to minimize the sum of weights of connections across the different groups and maximize the sum of weights of connections within each of the groups. In our proposed method, by incorporating the normalized factor regarding foreground region into the current closed-form criterion, we maximize the cohesion of alpha matte within the foreground region. It helps the foreground constraint to propagate to the entire foreground-like texture region in spite of using small windows to construct the matting Laplacian.

In a graph theoretic term, the closed-form solution to image matting criterion (5) can be written as a total dissimilarity between the foreground and background subregions, which is given by:

$$\boldsymbol{\alpha}^\top \mathbf{L}\boldsymbol{\alpha} = \sum_{i \in \mathcal{F}} \mathbf{D}_{ii} - \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{F}} \mathbf{A}_{ij} = \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{B}} \mathbf{A}_{ij}, \quad (6)$$

where the alpha matte, $\boldsymbol{\alpha}$, as an *indicator vector* represents partitions,

$$\alpha_i = \begin{cases} 1, & \text{if } i \in \mathcal{F} \\ 0, & \text{if } i \in \mathcal{B} \end{cases}, \quad \text{for } i = 1, \dots, N, \quad (7)$$

where \mathcal{F} and \mathcal{B} indicate the set of pixels of the foreground and background images, respectively.

Currently the existing closed-form approaches only utilize the sum of weights of connections across the distinct subregions(foreground and background). On the other hand, our proposed method does not only utilize the sum of weights of inter-connections, but also employs the total weights of intra-connections being in the foreground itself. The basic idea of our normalized matting is to pursue minimizing coherence of two disjunctive sets as well as maximizing cohesion within our interesting foreground region at the same time. Then, the *normalized matting* criterion is determined by:

$$\begin{aligned} \arg \min_{\boldsymbol{\alpha}} & \frac{\boldsymbol{\alpha}^\top (\mathbf{D} - \mathbf{A})\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \mathbf{D}\boldsymbol{\alpha}}, \\ \text{s.t.} & \quad \mathbf{C}_s \boldsymbol{\alpha} = \mathbf{h}_s, \\ & \quad 0 \preceq \boldsymbol{\alpha} \preceq 1, \quad \forall s \in \mathcal{S}. \end{aligned} \quad (8)$$

Note that the only the volume of \mathcal{F} in the denominator of the objective function is considered. Moreover, we need to obtain an optimal matte result from our

new criterion. For this, we employ a global optimization technique to find the solution. Motivated by existing work [7], we use a quadratic programming after getting its quadratic one. Defining $\mathbf{W} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$, $q = (\boldsymbol{\alpha}^\top \mathbf{D} \boldsymbol{\alpha})^{1/2}$ and $\boldsymbol{\omega} = (\frac{1}{q} \mathbf{D}^{1/2} \boldsymbol{\alpha})$, the normalized matting problem (8) is equivalent to another constrained quadratic problem given by:

$$\begin{aligned} & \arg \min_{\boldsymbol{\omega}} \boldsymbol{\omega}^\top \mathbf{W} \boldsymbol{\omega}, \\ \text{s.t. } & q \mathbf{C}_s (\mathbf{D}^{\frac{1}{2}})^{-1} \boldsymbol{\omega} = \mathbf{h}_s, \\ & 0 \preceq \boldsymbol{\omega} \preceq \frac{1}{q} \text{diag}(\mathbf{D}^{\frac{1}{2}}), \quad \forall s \in \mathcal{S}, \end{aligned} \quad (9)$$

If the objective function like (9) is convex quadratic and the constraint functions are affine, the above convex optimization problem provides a globally near-optimal solution in general [7].

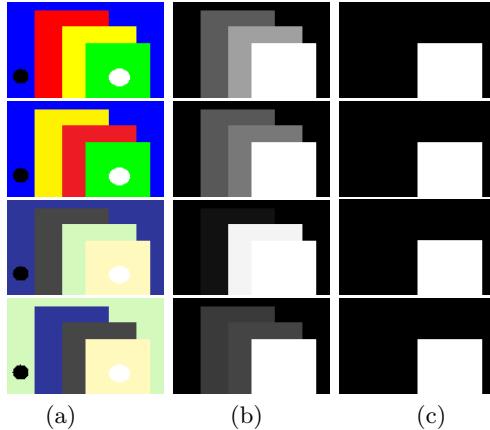


Fig. 1. Results of applying two different matting methods to sample images are shown (using 3x3 window). (a) Input images: white and black colors represent foreground and background constraints, respectively, and images of size 150 × 100. (b) Alpha matte using the closed-form solution (we used $\lambda = 100$). (c) Alpha matte using the proposed method.

3 Experiments

In this section, we show the usefulness of the proposed matting method, through the empirical comparison to the current closed-form solution. We applied our method to some toy images as well as complex natural images. As shown in Figs.1, the closed-form solution matting method fails to obtain complete alpha mattes and assigns some erroneous non-opaque values (highly depend on their color similarities between subregions) to the intermediate subregions because some of the small windows for propagation deviate from the constraints' color

models, whereas our proposed method shows remarkably more robust results due to the fact that the criterion Eq.(8) enforces color dissimilarities between foreground and other regions and just maximizes the foreground color cohesion at the same time. In Fig.3, Fig.2 and Table.1, three matting results of applying the two methods(closed-form solver & our method) to natural images, some intermediate alpha mattes according to their convergence level to the optimal minimizer, and performance(final objective value and computation time¹) comparison of two methods are shown, respectively. Although our method requires lots of computation time until our method converges a near-optimal matting result, from several experimental results, we can notice that the performance of our method is more stable and prominent. Such a good result is due to the consideration of adding foreground normalized factor.

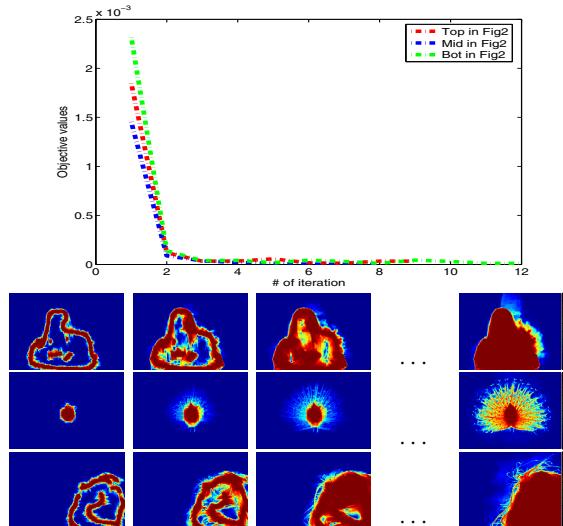


Fig. 2. Top:objective values for three data of Fig.3 when applying our method. Bottom: some intermediate alpha mattes according to convergence level.

Table 1. Performance on matting with data images of Fig.3

	Our method		CF solver	
	$\omega^\top \mathbf{W} \omega$	time(sec)	$\alpha^\top \mathbf{L} \alpha$	time(sec)
Top	3.346×10^{-5}	309.42	4.966×10^{-1}	26.31
Mid	1.538×10^{-6}	509.13	2.831×10^{-2}	47.91
Bot	1.122×10^{-5}	376.64	7.494×10^{-2}	34.80

¹ All times are recorded on a 2.9GHz dual-core Processor with 4GB of RAM in Matlab implementation.

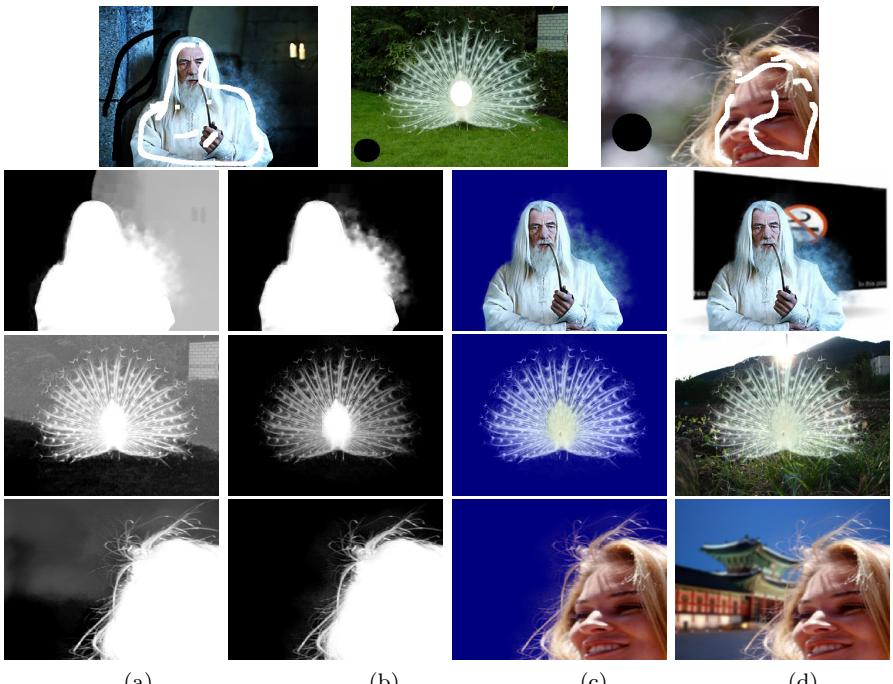


Fig. 3. Results of applying two different matting methods to the top natural three images are shown (using 3x3 window & considering the same foreground and background constraints). (Top) Input images (images of size 400×300): white and black scribbles represent foreground and background constraints, respectively. (a) Alpha matte using the closed-form solution (we used $\lambda = 100$). (b) Alpha matte using the proposed method, normalized matting. (c, d) Compositing matte result shown in (b) over a solid background and natural image, respectively.

4 Conclusions

We have presented a matting method, “normalized matting of interest region”, where we normalize the closed-form solution, employing a total weights of intra-connections in the foreground region. Useful aspects of our proposed matting method could be summarized as follows: It is simple but the normalized term for our interesting foreground region helps its constraint to propagate to the only foreground-like color region strictly in spite of using a small window. This feature can provide an efficient way of extracting an interesting region from other backgrounds with a few constraints (scribbles in here).

Acknowledgments. This work was supported by the Ministry of Culture, Sports, and Tourism (MCST) and the Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2011 [211A5020021098502001].

References

1. Chang, Y.Y., Curless, B., Salesin, D., Szeliski, R.: A bayesian approach to digital matting. In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, pp. 264–271 (2001)
2. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. *IEEE Trans. Pattern Analysis and Machine Intelligence* 30, 228–242 (2008)
3. Levin, A., Rav-Acha, A., Lischinski, D.: Spectral matting. In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, pp. 1–8 (2007)
4. Wang, J., Cohen, M.F.: An iterative optimization approach for unified image segmentation and matting. In: Proc. Int'l Conf. Computer Vision (2005)
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence* 22, 888–905 (2000)
6. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, pp. 61–68 (2006)
7. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)

Speeding Up SURF

Peter Abeles

Robotic Inception

pabeles@roboticinception.com

Abstract. SURF has emerged as one of the more popular feature descriptors and detectors in recent years. While considerably faster than SIFT, it is still considered too computationally expensive for many applications. In this paper, several algorithmic changes are proposed to create two new SURF like descriptors and a SURF like feature detector. The proposed changes have comparable stability to the reference implementation, yet a byte code implementation is able run several times faster than the native reference implementation and faster than all other open source implementations tested.

1 Introduction

The problem of associating features inside one image against another related image is image correspondence. Knowing image correspondence allows for the scene's structure and camera motion to be determined, in addition to object recognition. A typical processing flow for point based image correspondence involves, interest point detection, description of local regions, and feature association.

In recent years, Bay *et al.*'s Speeded-Up Robust Features (SURF) [1] has emerged as a popular choice for interest point detection and region description. Building upon previous work (e.g. SIFT [2]), SURF is primarily designed for speed and invariance to scale and in-plane rotation. While skew, anti-isotropic scaling, and perspective effects are considered second order.

SURF's speed is a significant improvement over its predecessors, but it is still considered to be too slow for many applications. In particular, embedded systems with constrained computational resources. More recently developed feature detectors/descriptors [3,4,5] have focused on improving speed while maintaining about the same level of stability found in SURF.

In this paper, several algorithmic changes to SURF are proposed. With the intent of improving upon the original algorithm's runtime performance while maintaining stability. The proposed algorithmic changes are justified through adherence to the smoothness rule (defined below) and a performance study. Two different SURF like descriptors are created from these proposed changes, SURF-S and SURF-F, along with a SURF like detector. SURF-S provides comparable stability to the reference implementation while running several times faster than the reference library. SURF-F sacrifices a bit of stability for more than two times speed improvement relative to SURF-S. Source code for the proposed

algorithm is freely available and included in the open source computer vision library BoofCV [6].

2 Speeded-Up Robust Features

The following is a high level overview of the SURF detector and descriptor. For a complete discussion consult the SURF paper [1]. SURF achieves speed across a range of scales through the use of integral images [7,8]. Transforming an image into an integral image allows the sum of all pixels contained inside arbitrary axis aligned rectangle to be found in four floating point operations.

The value of each pixel (x, y) in the integral image I_Σ is computed by summing pixel intensities within a rectangle up to (x, y) :

$$I_\Sigma(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (1)$$

Then to find the sum of pixel values contained in a rectangle R_Σ compute:

$$\begin{aligned} R_\Sigma(x_1, y_1, x_2, y_2) &= I_\Sigma(x_2, y_2) - I_\Sigma(x_2, y_1 - 1) \\ &\quad - I_\Sigma(x_1 - 1, y_2) + I_\Sigma(x_1 - 1, y_1 - 1) \end{aligned} \quad (2)$$

where $(x_1, y_1) \leq (x_2, y_2)$.

Interest point detection is done using an approximation of the Hessian determinant scale-space detector [9]. The Hessian's determinant is found by approximating the Gaussian's second order partial derivatives (D_{xx}, D_{yy}, D_{xy}) using box integrals, as described in [1].

$$\det(H) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (3)$$

This is done across different sized regions and scales. Interest points are defined as local maximums in the 2D image and across scale-space. Scale and location are interpolated by fitting a 3D quadratic [10] to feature intensity values in the local 3x3x3 region.

Several different variations on the SURF descriptor are described in [1], but only the oriented SURF-64 descriptor is considered in this paper. Orientation is estimated by computing the Haar wavelet (effectively the image gradient) inside a neighborhood of radius of $6s$, where s is the feature's scale. The gradient is weighted by a Gaussian centered at the interest point, its angle computed, and saved into an array. Using a moving window of $\frac{\pi}{3}$ radians, the window with the largest gradient sum is found and the feature's orientation computed from its sum.

The feature description is computed inside a square region of size $20s$, aligned to the found orientation. This region is then broken up into a 4 by 4 grid for a total of 16 sub-regions, which are of size $5s$. For each sub-region the sum of the gradient and sum of the gradient's absolute value is computed:

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (4)$$

These responses are weighted using a Gaussian distribution. Each subregion contributes 4 features (v), resulting in a total of 64 features for the descriptor.

The gradient can only be efficiently computed along the image's axis. To accommodate for the feature's orientation, the gradient is rotated so that it is oriented along the feature's axis.

3 Implementation Details

To create a stable region descriptor or feature detector, small changes in location and scale must cause a proportionally small change to the feature's description or location, respectively. The preceding statement is referred to as the smoothness rule. Similar statements are made by D. Lowe [2] and justified by a biological vision model [11].

The following are several general techniques for enforcing the smoothness rule: 1) Use interpolation functions with continuous values when sampling pixel intensity. 2) Increase a sample region's size to reduces the fractional change in value when crossing a pixel border. 3) Avoid interacting with image and object borders. Technique 2 and 3 can be conflicting since as the region size increases it is more likely to interact with the boundary conditions.

3.1 Descriptor Interpolation

Interpolation of the gradient's response when computing the descriptor (v in Eq. 4) is not fully described in [1]. This has resulted in several different algorithmic interpretations. The most straight forward interpretation is to use nearest-neighbor interpolation. However, this method does not have a smooth transition between pixel boundaries, degrading descriptor stability.

Agrawl *et al.* [12] propose to have each subregion overlap by adding a padding of 2s and to weigh the gradient using a subregion centered Gaussian distribution. The resulting descriptor has a region of size 24s. Pan-o-Matic [13] samples the gradient using a variable number of points depending on the ratio of region size to sample size and then uses trilinear interpolation to compute the descriptor values, similar the approach used by SIFT. SURF-S uses the technique proposed by Agrawl *et al.*, while SURF-F uses nearest-neighbor interpolation.

3.2 Image Border

Interest points can have descriptors which extend outside of the image border. How this edge case is handled has a significant effect on stability. For example, simply discarding features which touch the border causes a 17% drop in stability when compared to the method proposed below. The SURF paper does not discuss how to handle this case. A good balance between speed and stability is found by setting the response of any operator crossing the image border to be zero. This approach is used by both SURF-S and SURF-F.

3.3 Interest Point Interpolation

In SURF, after an interest point has been detected using non-maximum suppression, its position (x, y, s) is interpolated as the extreme of a 3D quadratic, see Brown and Lowe [10]. This technique uses second order derivatives (Laplacian) computed using pixel differences, which amplifies noise. Ad hoc modifications are required to filter out illogical solutions generated with this approach.

To avoid these issues, it is proposed that a 1D quadratic is used instead. If the minimum number (which is 3) of points are used and the center point is the peak, the interpolated peak must lie inside the local region. Both SURF-S and SURF-F work by fitting quadratic 1D polynomials across each (x, y, s) axis independently. While not capturing off axis structural information, empirical tests show comparable stability, is easier to implement, requires fewer operations, and does not require numerical differentiation.

3.4 Derivative Operator

The SURF paper states that a Haar wavelet is compute at regularly spaced sample points inside each sub-region when computing the description. A common interpretation of this statement is to use a template similar to the one shown in the left side of Figure 1, as has been done by OpenSURF [14,15] and OpenCV [16]. This template lacks symmetry, which causes a directional bias. An alternative symmetric derivative operator is proposed that overcomes this issue, see Figure 1. The alternative kernel has a width of $w = \text{rnd}(2rs) + 1$, where r is the radius at a scale of one. A value of $r = 1$ is recommended for descriptor computations.

Both SURF-S and SURF-F use the symmetric derivative operator.

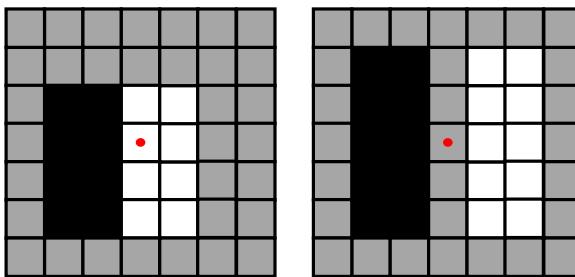


Fig. 1. Left: A common interpretation for the Haar-like derivative operator. Right: Proposed symmetric derivative operator. Red dot indicates the region's center. The Haar kernel lacks symmetry, causing a bias. Dark squares indicate a weight of -1 and white squares +1.

3.5 Orientation Estimation

The region orientation algorithm proposed by SURF is computationally expensive, see Section 2 for a summary. An alternative and much faster approach is to

compute a weighted sum of the gradient ($\sum d_x, \sum d_y$) and then find the angle using atan2($\sum d_y, \sum d_x$), where the gradient is found using pixel differences, e.g. Sobel. However, the improved speed comes at the cost of some stability.

SURF-F uses the orientation estimation technique described above, while SURF-S uses the original algorithm proposed in SURF.

3.6 Laplacian Sign

Another smaller performance boost can be found in delaying the Laplacian's sign computation. It is stated in [1] that the Laplacian's sign can be computed with no loss in performance. This is not quite true; the computation requires an additional operation for each pixel and scale, plus storage. Instead if the Laplacian sign is computed for found interest points only, then 24 additional operations are required per feature. Since the number of pixels is much greater than the number of found features, the latter is many times faster and requires no additional storage.

4 Test Setup

Since SURF was originally proposed, numerous implementations have released for various programming languages and hardware platforms. A reference implementation [17] has been released by the original authors, but only in an out-of-date binary format. To validate the performance of the proposed changes, a comparison of several open source libraries and the reference library is performed.

Implementation	Cite	Version	Language	Comment
SURF-F	[6]	v0.12	Java	Faster but less accurate
SURF-S	[6]	v0.12	Java	Slower but more accurate
JavaSURF	[18]	SVN r4	Java	No orientation
JOpenSURF	[15]	SVN r24	Java	Java port of OpenSURF
OpenCV	[16]	2.4.3	C++	
OpenSURF	[14]	12/04/2012	C++	
Pan-o-Matic	[13]	0.9.4	C++	
Reference	[17]	1.0.9	C++	Original author

Fig. 2. List of evaluated implementations in alphabetical order. If a formal version is lacking or insufficient then a repository version is referenced. The proposed algorithms (SURF-F and SURF-S) are included in the BoofCV library.

Evaluation is performed using test image sequences from Mikolajczyk and Schmid [19]. Each sequences has a set of known image homographies relating images to the first image in the sequence. A homography provides a one-to-one relationship for pixel locations between two images. Each sequence is designed to test different types of distortion and image noise. The evaluated data sets include “bark”, “bikes”, “boat”, “graf”, “leuven”, “trees”, “ubc”, and “wall”.

Evaluated libraries are listed in Table 2. Only single threaded libraries are considered. The two proposed modifications to SURF have been written in Java, but both C++ and Java libraries are considered. Libraries written in C++ have a 2 to 3 times speed advantage due to less overhead.

Several parts of SURF are easily parallelized, such as the descriptor calculation. While beyond the scope of this paper, multi-threaded [16] and GPU [20] assisted implementations are also readily available. These implementations are not considered since the focus of the paper is on algorithmic improvements which are independent of hardware.

5 Performance Metrics

Performance metrics used to evaluate detector stability, and descriptor stability are described in the sub-sections below. The performance metric for runtime speed is elapsed time.

5.1 Descriptor

Descriptor stability is evaluated using F-measure. F-measure is defined using precision and recall statistics. Precision is defined as true positives divided by true positives and false positives. Recall is defined as true positives divided by true positives and false negatives. Recall combines both metrics into a single number.

$$F = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

A true positive is defined as an associated feature for which its location is within tolerance of the true location, as defined by the known homography. False negatives are features with corresponding points in both image, but which are not associated. Association is done using a greedy algorithm that minimized Euclidean error.

All libraries use interest points detected by the reference library to compute descriptors from. By doing so, the descriptor's stability can be decoupled from the detector's performance. Each library is configured to describe SURF-64 features. Summary statistics shown in the left side of Figure 4 are found by summing the F-measure across each image in every sequences.

5.2 Detector

Detector stability is measured using a modified version of repeatability. As stated in [21] repeatability is a measure that “signifies that detection is independent of changes in imaging conditions”. One problem with repeatability is it favors detectors that detect more features [22]. Excessive detections increase computational cost without improving association quality. An extreme example is if every pixel is marked as an interest point, it would have perfect repeatability.

To compensate for this issue, the definition of repeatability has been modified to ignore regions with closely packed points. By only considering interest points with unambiguous matches, repeatability bias is reduced.

The modified repeatability measure r_i is defined below:

$$r_i = \frac{|A_i| - |T_i|}{|P_i| - |T_i|} \quad (6)$$

where P_i is the set all points, A_i is the set of actual matches, and T_i is the set of ignored matches.

$$P_i = \{x_a \in F_o | H_i x_a \in I_i\} \quad (7)$$

$$A_i = \{x_c \in F_i | \|H_i x_b - x_c\| < \epsilon, x_b \in P\} \quad (8)$$

$$T_i = \{x_d \in F_i | \|x_c - x_d\| < \epsilon, x_d \neq x_c\} \quad (9)$$

where I_i is image i , H_i is the homography transform from image 1 to i , F_i is the set of all detected interest points, x is an interest point, and ϵ is the match tolerance.

Two interest points are considered a match if their position and scale are within tolerance. The true position is found using the provided homography. Scale is computed by 1) sampling four evenly spaced points one pixel away from the interest point, 2) applying homography transform to each sample point and interest point, 3) finding the distance of transformed sample points from transformed interest point, and 4) setting expected scale to average distance.

Tuning each library to detect the same number of features in all images proved to be impossible. Instead they are tuned to detect about 2,000 features in image 1 in the graf sequence.

Detector configuration:

1. Octaves: 4
2. Scales: 4
3. Base Size: 9
4. Pixel Skip: 1

Tolerance for position is 1.5 pixels and 25% for scale. Relative ranking were found to be insensitive to changes in threshold values.

Summary statistics shown in right side of Figure 4 are found for each implementation by summing repeatability across each image in every sequence and dividing by the best implementation's score.

5.3 Runtime Speed

Runtime performance is measured by having each library detect and describe about 2,000 features inside image 1 in graf sequence. Evaluation procedure:

1. Measure elapsed time to detect and describe features.
2. Repeat 10 times in the same process and output best result.
3. Run the whole experiment 11 times for each library and record the median time.

All tests are performed on a desktop computer with Ubuntu 10.10 installed and an Intel Q6600 2.4GHz CPU. Native libraries are compiled using g++ 4.4.5 with the -O3 flag. Java libraries are compiled and run using Oracle JDK 1.6.38 64 bit. No additional flags are passed to the Java Runtime Environment, the -server flag is implicit.

Native library runtime speeds are highly dependent upon the level of optimization done by the compiler and which instructions they are allowed to use. Additional hardware specific flags are not manually injected into build scripts beyond what was provided by the authors. Elapsed time is measured in the actual application using System.currentTimeMillis() in Java and clock() from time.h in C++.

6 Performance Results

Summary results for runtime performance, descriptor stability, and detector stability are shown in Figures 3, and 4. Stability results for describe and detection for individual sequences are shown in Figures 5 and 6, respectively.

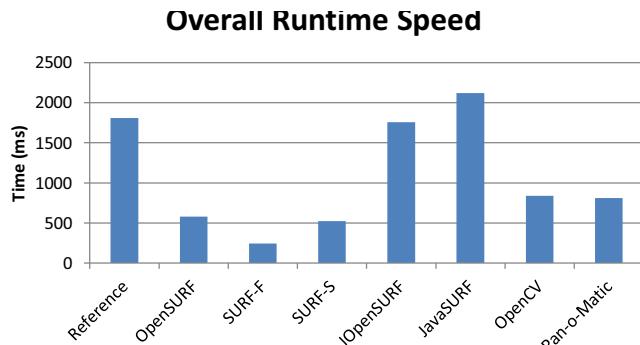


Fig. 3. Runtime speed for detecting and describing image features. Lower bars are better. Each library is tuned to detect approximately 2000 features in a 850x680 image.

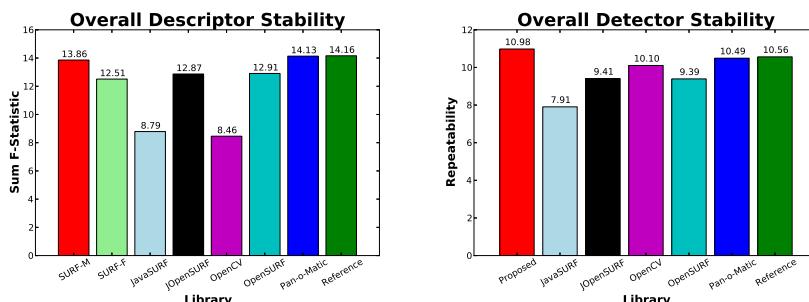


Fig. 4. Left: Summary of descriptor stability using correct association fraction across all image sequences. Right: Summary of detector stability using a modified repeatability measure across all image sequences. Higher bars are better.

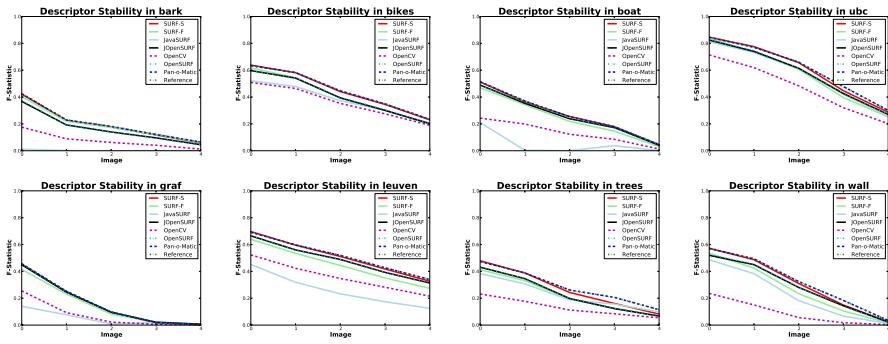


Fig. 5. Overall descriptor stability. Sum of F-statistic across all image sequences. All implementations use the same set of interest points (provided by reference library) to decouple descriptor performance from detector performance.

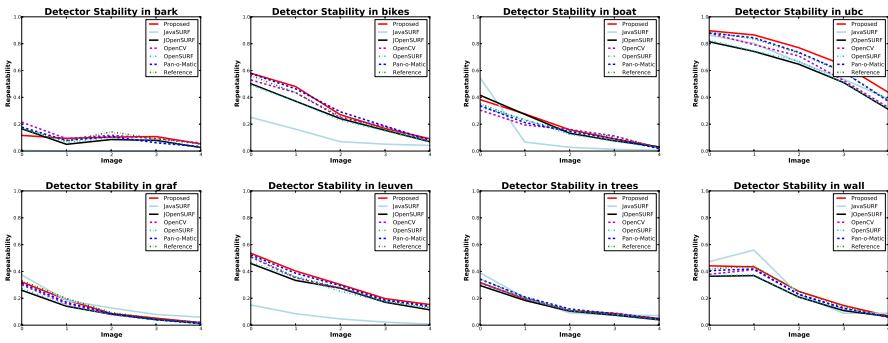


Fig. 6. Overall detector stability. Sum of repeatability across all image sequences.

Pan-o-Matic and the reference library have the best descriptor stability, closely followed by SURF-S. For detector stability, the proposed detector has the best stability, followed by Pan-o-Matic and the reference library. SURF-F is the fastest implementation, despite being written in Java. The runners-up are Open-SURF and SURF-S, which have nearly the same runtime speed, but are two times slower than SURF-F.

7 Conclusions

To improve the runtime speed of SURF, several algorithmic changes have been proposed, resulting in two new descriptor variants and one detector. The proposed changes are designed to be more computationally efficient and follow the smoothness rule to ensure stability. Performance of the proposed changes are

validated by a stability and runtime performance study of eight SURF implementations. Source code is available inside the BoofCV open source library.

Descriptor stability varied significantly by implementation, with SURF-S having comparable performance to the reference library, the top performer. SURF-F was shown to be about 10% less stable than SURF-S, which was still comparable to or significantly better than four other implementations. The proposed detector had the best stability. The proposed detector, combined with SURF-S and SURF-F, were the two fastest SURF implementations. SURF-F running more than twice as fast as SURF-S. It is worth noting that proposed algorithms were implemented in Java and run in a virtual machine. Due to overhead, a well written port to C++ is likely to run 2 to 3 times faster.

References

1. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)* 110, 356–359 (2008)
2. Lowe, D.: Distinctive image features from scale-invariant keypoints, cascade filtering approach. *International Journal of Computer Vision (IJCV)* 60, 91–110 (2004)
3. Tola, E., Lepetit, V., Fua, P.: Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. *Pattern Analysis and Machine Intelligence* 32, 815–830 (2010)
4. Juan, L., Gwon, O.: A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)* 3, 143–152 (2009)
5. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
6. Abeles, P.: Boofcv (Version 0.5), <http://boofcv.org>
7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, CVPR*, pp. 511–518 (2001)
8. Simard, P., Bottou, L., Haffner, P., LeCun, Y.: A fast convolution algorithm for signal processing and neural networks. In: *NIPS* (1998)
9. Lindeberg, T.: Feature detection with automatic scale selection. *IJCV* 30, 79–116 (1998)
10. Brown, M., Lowe, D.: Invariant features from interest point groups. In: *BMVC* (2002)
11. Edelman, S., Intrator, N., Poggio, T.: Complex cells and object recognition (1997), <http://kybele.psych.cornell.edu/~edelman/archive.html>
12. Agrawal, M., Konolige, K., Blas, M.R.: CenSurE: Center surround extremas for realtime feature detection and matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008*, Part IV. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
13. Orlinski, A.: Pan-o-matic (Version 0.9.4), <http://aorlinsk2.free.fr/panomatic/>
14. Evans, C.: The opensurf computer vision library, <http://www.chrisevansdev.com/computer-vision-opensurf.html> (Build May 27, 2010)
15. Stromberg, A., Jojopotato, N.: Jopensurf (SVN r24) Note: Port of OpenSURF, <http://code.google.com/p/jopensurf/>
16. Liu, L., Mahon, I.: Opencv (Version 2.3.1 SVN r6879), <http://opencv.willowgarage.com/wiki/>

17. Bay, H., Gool, L.V.: Surf: Speeded up robust feature (Version 1.0.9),
<http://www.vision.ee.ethz.ch/~surf/>
18. Fantacci, C., Martini, A., Mitreski, M.: Javasurf (SVN r4) Note: Refactored P-SURF, <http://code.google.com/p/javasurf/>
19. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1615–1630 (2005)
20. Cornelis, N., Van Gool, L.: Fast scale invariant feature detection and matching on programmable graphics hardware. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2008* (2008)
21. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. *Int. J. Comput. Vision* 37, 151–172 (2000)
22. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision* 94, 335–360 (2011)

Distortion Adaptive Image Classification – An Alternative to Barrel-Type Distortion Correction

Michael Gadermayr¹, Andreas Uhl¹, and Andreas Vécsei²

¹ Department of Computer Sciences, University of Salzburg, Austria

{mgadermayr, uhl}@cosy.sbg.ac.at

² St. Anna Children's Hospital, Endoscopy Unit, Vienna, Austria

Abstract. The endoscopes, utilized in computer aided celiac disease diagnosis are equipped with wide-angle lenses, which introduce significant barrel-type distortions. Previous work is on rectifying the distortions prior to the feature extraction. However, due to new arising inadequacies this often even decreases the classification accuracy. The idea of this paper is based on the fact, that there is a correspondence between the position of a patch in the image and the amount (and orientation) of lens distortions. Therefore, in order to classify an image patch, only a reduced training set of similarly distorted patches is considered. We show that in most cases with the new approach higher classification rates can be achieved compared to traditional distortion corrected and uncorrected image classification.

Keywords: Endoscopy, classification, celiac disease.

1 Introduction

Celiac disease is an autoimmune disorder which affects the small bowel in genetically predisposed individuals after introduction of gluten containing food. Characteristic for this disease is an inflammatory reaction in the mucosa of the small bowel caused by a dysregulated immune response triggered by gluten proteins. Thereby the mucosa loses its absorptive villi and hyperplasia of the enteric crypts occurs leading to a decreased ability to absorb nutrients.

Computer aided celiac disease diagnosis relies on images taken during endoscopy. The employed cameras are equipped with wide angle lenses, which suffer from a significant amount of barrel-type distortions. Whereas the distortion in central image pixels can be neglected, peripheral regions are highly distorted. Thereby, the feature extraction as well as the following classification is compromised. Based on camera calibration, distortion correction (DC) techniques are able to rectify the images. However, although the barrel-type distortion can be undone, especially in peripheral regions there remains a lack of information, as the DC method stretches the image. The lack of information has to be compensated using an interpolation technique.

In recent studies [1, 2], the impact of barrel type distortions and distortion correction on the classification rate of celiac disease endoscopy images has been investigated. In [1], the authors have shown that patches in peripheral regions, which are more affected by the distortions are more likely to be misclassified. Furthermore, the higher the distortion difference between a patch and its nearest-neighbor patch the more likely

a patch gets misclassified. With distortion correction, the classification rate on average even suffers. In [3], different distortion correction techniques have been investigated. The computer aided celiac disease diagnosis [1–3] is based on 128×128 pixel patches, which are manually extracted from reliable regions in the original images (with e.g. 768×576 pixels). A computer-aided detection of sensible patches is a separate problem definition.

In this paper, the focus is not on distortion correction, but on increasing the classification performance (i.e. the overall classification accuracy), by considering the position of the respective image patches. As the position of a patch correlates with the orientation and strength of the barrel-type distortions, the classifier is enabled to adaptively handle variably distorted patches. In the following, our new approach is called Distortion-Adaptive-Classification (DAC). Actually, DAC is not limited to barrel-type distortions. It can also be applied in case of other systematic image degradations in combination with patch-based image classification, which has become an important field of research (e.g. see [4]).

The paper is organized as follows: In Sect. 2, our new approach is introduced and compared with distortion corrected and uncorrected classification. In Sect. 3, experiments are shown and the results are discussed. Section 4 concludes this paper.

2 Distortion-Adaptive-Classification (DAC): Classification Based on Patch Position Information

2.1 Barrel-Type Distortions

Fig. 1a shows a checkerboard pattern, captured with an endoscope used in celiac disease diagnosis. Especially in peripheral regions, significant distortions can be recognized. It can be seen, that e.g. the scale (i.e. the size of the squares) in these regions considerably decreases.

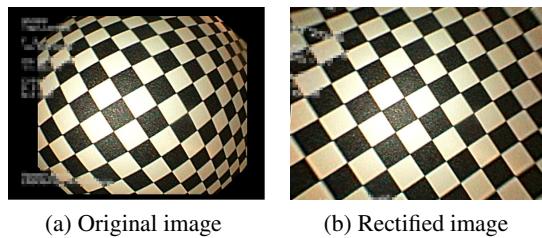


Fig. 1. Distorted and undistorted image of a planar checkerboard pattern

A general model for barrel-type distortions is given by

$$x_u(x_d) = \hat{x}_c + \frac{(x_d - \hat{x}_c)}{\|x_d - \hat{x}_c\|_2} \cdot r_u(\|x_d - \hat{x}_c\|_2), \quad (1)$$

where x_u is an undistorted point, x_d is a distorted point and \hat{x}_c is the center of distortion. The function r_u , which maps a distance from the center of distortion in the distorted

image ($\|x_d - \hat{x}_c\|_2$) to a distance in the undistorted image, is modeled differently. For comparisons in this work, the distortion correction approach introduced in [5], which turned out to be appropriate for our purpose, is utilized. In this approach, r_u is modeled by the division model.

$$r_u(r_d) = \frac{r_d}{1 + \xi \cdot r_d^2}. \quad (2)$$

Other well known approaches are based on polynomial or parameter free models. Figure 1b shows an undistorted image corresponding to the distorted image shown in Fig. 1b. As can be seen, the geometrical properties are effectively rectified.

2.2 New Approach – An Alternative to Distortion Correction

In [1], the authors concluded, that there is a correlation between the misclassification probability and the radial distance (see Eq. (5)) to the nearest neighbor patch, in case of classifying barrel-type distorted celiac disease images. That means, if the patch with the smallest feature distance (i.e. the nearest neighbor patch) has a large radial distance, the patch is more likely to be misclassified. Although distortion correction techniques are able to rectify the geometrical image properties quite well (see Fig. 1), problems within computer aided diagnosis cannot be fully eliminated, as new difficulties arise within the distortion correction. One major problem is that the image has to be distinctly stretched, especially in peripheral regions. This leads to (variably) blurred images.

The basic idea of our new approach is based on the fact, that there is a correspondence between the position of a patch in the image and the amount as well as the orientation of the lens distortions. Therefore, in order to classify a certain image patch, an individually reduced training set of similarly distorted patches is generated. Patches with a high geometrical distance to the current patch show another level of distortions, and therefore are not included in the training set.

For the geometrical distance between two patches, we consider their center points p_1 and p_2 in the original (distorted) image. We identified the following potentially sensible metrics:

– Euclidean Distance:

The quite simple and commonly known Euclidean distance is given by:

$$d_E = \|p_1 - p_2\|_2. \quad (3)$$

– Distortion Level Euclidean Distance:

The Euclidean distance does not incorporate the strengths of distortions, depending on the patch position (peripheral regions vs. center regions). This metric utilizes undistortion to get higher distances in peripheral regions which correspond to higher distortions.

$$d_{DLE} = \|x_u(p_1) - x_u(p_2)\|_2. \quad (4)$$

– Radial Distance:

Patches with a similar distance to the center of distortion are similarly distorted as far as its strength is concerned (see (1): r_u only depends on the distance to the

center of distortion \hat{x}_c). The idea of this metric is, to totally ignore the distortion orientation and to focus on the strength only.

$$d_{RAD} = \text{abs}(\|p_1 - \hat{x}_c\|_2 - \|p_2 - \hat{x}_c\|_2) . \quad (5)$$

– Distortion Distance:

This metric is based on the geometrical shape of the patch in case of undistortion. Similarly warped patches have small distances and vice versa. For simplicity, we only consider the diagonal lengths which are good indicators for distortions. Experiments with other indicators (e.g. using the parameters of an approximated affine matrix) did not lead to significantly different results. The diagonals $diag_i$ are achieved by adding different offsets to the patch center point p and undistorting these corner points (s denotes the side length of a patch (in our case 128)).

$$d_{DD}(p) = |diag_1(p_1) - diag_1(p_2)| + |diag_2(p_1) - diag_2(p_2)| . \quad (6)$$

$$diag_1(p) = \|x_u(p + (\frac{-s}{2}, \frac{-s}{2})) - x_u(p + (\frac{+s}{2}, \frac{+s}{2}))\|_2 . \quad (7)$$

$$diag_2(p) = \|x_u(p + (\frac{+s}{2}, \frac{-s}{2})) - x_u(p + (\frac{-s}{2}, \frac{+s}{2}))\|_2 . \quad (8)$$

For DAC, one distance metric and one threshold must be chosen. The training set for a certain patch consists of all patches with a distance below this threshold. All other patches are simply ignored during classification. We have defined 20 sensible thresholds (e.g. for the Euclidean metric between 80 pixels and infinity (infinity means, DAC is disabled)) for each distance metric. For each feature, the best threshold has been evaluated during exhaustive search.

The classification of images suffering from barrel-type distortions is only one application scenario for DAC. Actually, the new approach is potentially sensible in each case of classifying images with systematic image degradations. With such degradations, we refer to the problem of having variable (average) image properties over the original image where the patches are extracted from. For example, within endoscopy a source of light mounted on top of the endoscope could cause a slightly varying exposure over the image. Features being not invariant to illumination, might suffer from such slight differences, and could profit from DAC. A crucial issue is the identification of an adequate distance metric. The metrics defined in (3) - (6) have been identified to be potentially useful in case of barrel-type distortions. However, especially the Euclidean distance (3) is expected to be good choice in general.

2.3 Features for Classification

In order to investigate the effects of the proposed approach we utilize a couple of texture and shape features. Instead of evaluating the best configuration for each feature, we are computing classification rates with various configurations in order to make a general statement on the effect of our new approach.

The following feature extraction methods are investigated:

- **Local binary patterns [6] (LBP):**

LBP is used with varying radii (r) and a varying number of neighbors (s). The feature vector's dimensionality is 2^s .

- **Extended local binary patterns [7] (ELBP):**

ELBP, which is an edge based LBP derivative, is used with varying radii and a varying number of neighbors.

- **Local ternary patterns [8] (LTP):**

LTP is used with varying radii and a varying number of neighboring samples and a fixed threshold ($\Theta = 3$), which turned out to be a good choice for various configurations. Although the authors proposed a coding scheme to get only $2 \cdot 2^s$ dimension, to get a very high dimensional feature for our experiments, we implemented LTP with 3^s dimensions.

- **Contrast [9] (CONTRAST):**

The feature vector consists of the Haralick contrast feature [9] calculated for different offsets $(0, r)^T, (r, 0)^T, (r, r)^T, (r, -r)^T$ (i.e. the feature has 4 dimensions). Varying ranges r are investigated.

- **Fourier power spectrum (FOURIER):**

After computing the Fourier power spectrum, rings with varying radii and a thickness of 2 pixels are extracted and the means of these values is calculated. For our experimental usage, we only consider the discriminative power of single rings. (i.e. the dimensionality of a single feature is 1).

- **Shape Curvature Histogram [10] (SCH):**

SCH is a shape feature, especially developed for celiac disease diagnosis. A histogram contains the occurrences of the contour curvature values. In our experiments, we consider various histogram bin numbers (the bin number corresponds with the dimensionality of the feature).

3 Experiments

3.1 Experimental Setup

The image test set used contains images of the *duodenal bulb* taken during duodenoscopies at the St. Anna Children's Hospital using pediatric gastrosopes (with resolution 768×576 pixels). In a preprocessing step, texture patches with a fixed size of 128×128 pixels were manually extracted. The size turned out to be optimally suited in earlier experiments on automated celiac disease diagnosis [11]. Before features are extracted, all patches are converted into gray value images. Using additional color information, no significant improvements are achieved. In case of distortion correction, the patch positions are adjusted according to the distortion function.

To generate the ground truth for the texture patches used, the condition of the mucosal areas covered by the images was determined by histological examination (i.e. biopsies) from the corresponding regions. Severity of villous atrophy was classified according to the modified Marsh classification, as proposed in [12].

Although it is possible to distinguish between the different stages of the disease, we only aim in distinguishing between images of patients with (Marsh 3A-3C) and without



(a) Marsh 0: Patches clearly showing the villous structure of healthy mucosa.
(b) Marsh 3A-3C: Patches showing the villosus atrophy.

Fig. 2. Example patches of patients without (a) and with the disease (b)

the disease (Marsh 0). We decided for this strategy, as the two classes case is more relevant in practice. Our experiments are based on a database containing 135 (Marsh 0) and 115 (Marsh 3A-3C) images, respectively. Example texture patches are shown in Fig. 2. To study the effect of our approach on the classification rate (accuracy), we use leave-one-patient-out cross validation combined with the nearest-neighbor classifier. This rather weak classifier is chosen, as its results can be easily interpreted (see [1]). In an extended future work, the impact of various classifiers will be investigated.

3.2 Results

In Fig. 3, an overview about our results is given. The bars indicate the average improvements with a certain feature and a certain distance metric in comparison to the approach being not based on DAC or distortion correction. The higher (white-colored) top of the bars represent the results, achieved if the threshold is optimized separately for each feature and each distance metric. The lower bars (colored in shades of gray) indicate the mean above the best rate and its 4 neighbors (2 neighbors with smaller and 2 with higher threshold) as far as the distance-threshold is concerned. In order to avoid over-fitting, we will primarily focus on these lower values. The quite simple Euclidean distance metric seems to be the best choice, as for each feature it delivers the best of at least highly competitive results (if considering either the higher or the lower rate). Although the differences are quite small, we decided to choose the Euclidean metric for a more detailed comparison.

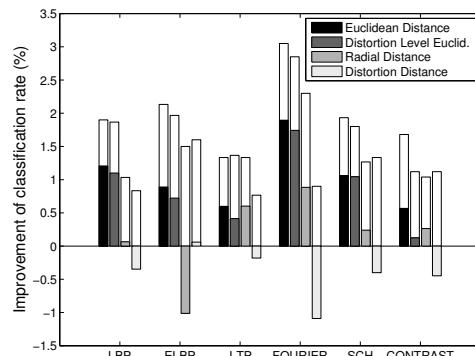


Fig. 3. Overview about the features and the distance metrics

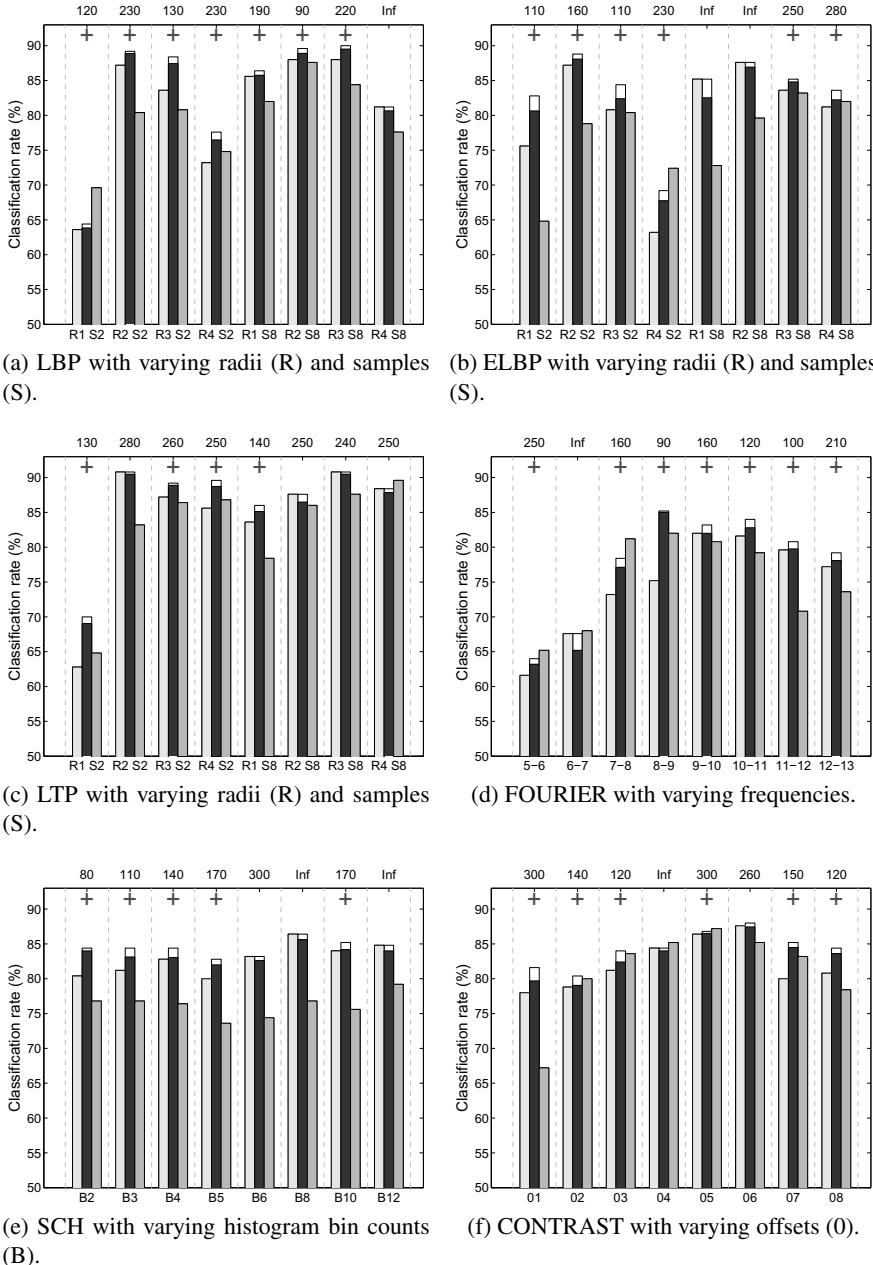


Fig. 4. For each feature and each configuration, the center bar shows the classification rates achieved with our new DAC approach in comparison to the traditional approach without (left bar) and with distortion correction (right bar)

In Fig. 4, the overall classification accuracy results achieved with varying configurations for each feature are visualized. For comparability, the left of the three bars shows the classification rate achieved without distortion correction (noDC) and the right bar shows the rates achieved with distortion correction (DC). The center bar shows 2 different rates achieved with our new DAC approach. The higher (white-colored) top of the bar indicates the best rate achieved with the Euclidean distance metric and varying thresholds. The threshold which corresponds to this rate is shown on top of each column. The lower (dark-colored) bar indicates the mean above the best rate and its 4 neighbors (2 neighbors with smaller threshold and 2 with higher threshold) as far as the distance-threshold is concerned. As in the overview, we will focus on the lower value. A large higher value can be achieved in case of a deceptive search-space. With the averaged lower value, this problem is circumvented. An indicator for the deceptiveness of the respective search-space is given by the difference between the higher and the lower rate. In our plots, a “+” indicates features which profit from our new approach (i.e. the lower rate is greater than ($>$) the rate achieved with traditional classification).

With LBP and ELBP, especially with the smaller sample size 2, in each case an improvement to noDC can be observed. With 8 samples, only with specific configurations, DAC turns out to be advantageous. The best overall classification rate can be improved with the location consideration. Considering LTP, our new approach seems to be slightly less competitive (especially in combination with samples size 8). The best overall classification rate cannot be outreached. With FOURIER, in most cases an increase of the discrimination power can be achieved and the best overall classification rate can be improved. Although DC also improves the results (in case of low frequencies), our approach obtains the best overall rates. SCH profits from our approach in case of a small histogram bin size (from B2 to B5). The best overall rate cannot be improved. CONTRAST slightly profits in most cases. However, the best overall rate cannot be improved, considering the lower classification value of DAC.

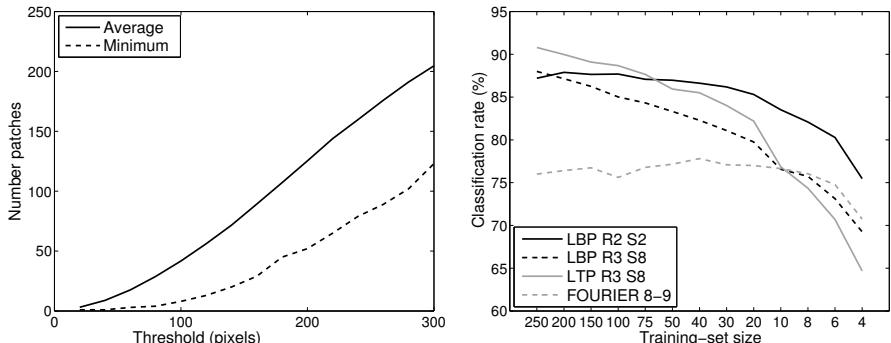
3.3 Discussion

For all features, considering specific configurations, our approach can improve the discriminative performance, however, with some configurations no benefit can be achieved.

With LBP-like features, by tendency with a lower number of samples, the improvement with our method is higher. A similar behavior can be observed with the histogram bin count and the SCH feature. Seemingly, high dimensional features suffer from DAC. Especially with the very high dimensional LTP feature in combination with 8 samples, mostly an increase cannot be achieved. In opposite the 1-dimensional FOURIER features on average lead to the highest benefit of DAC (see Fig. 3).

High dimensional features might suffer because with DAC the training-dataset size is decreased (depending on the chosen threshold). In Fig. 5a, the average and the minimum considered patch counts are given for variable thresholds. Whereas the average count for e.g. a distance threshold of 100 pixels is quite acceptable, the minimum count (which typically corresponds with peripheral patches), is very low.

Figure 5b shows for some example features the average decrease of the classification rate, in case of reducing the training-set randomly (in opposite to DAC where the count is reduced intelligently). Therefore, we averaged the classification rates achieved with



- (a) This plot shows the average and the minimum considered patch count, dependent on the chosen threshold (the Euclidean distance is used).
 (b) Classification rates (y-axis) achieved with example features and shrinking randomly chosen training-sets (the sizes are given on the x-axis).

Fig. 5. The decrease of the training-set size (a) with DAC and the impact in case of a random (instead of an intelligent) decrease (b)

30 randomly chosen image sets for each training-set size. We notice that especially, the high dimensional features (LTP and also LBP R3 S8) highly suffer from a reduced training-set. In opposite, with the low-dimensional FOURIER and LBP R2 S2, the classification rate suffers less distinct. As expected, a randomly decreased training-set size on average leads to a loss of discriminative power. This confirms that the increases of the classification rates with DAC are definitely not due to advantages of small training-set sizes. On the contrary, having a larger overall training-set, the patch counts (Fig 5a) could be increased and therefore we anticipate an improvement with DAC even with high dimensional features.

Another quite interesting aspect is to separately investigate features operating in a smaller and others operating in a larger neighborhood. Distortion correction, by tendency profits from larger neighborhood sizes (see LBP, LTP or ELBP with R4 (compared to R1) and FOURIER with small frequencies (compared to large frequencies)). Larger neighborhoods (i.e. lower image frequencies are considered) are more affected by barrel-type distortions on the one hand and less affected by interpolation on the other hand, which is a problem within DC. With DAC, this effect is at least less distinct which is obvious as it is not based on distortion correction in combination with interpolation.

To put it in a nutshell, with DAC and our dataset the classification rate of 24 out of 27 features (88.9 %) with up to 4 dimensions and 29 out of 36 features (80.6 %) with up to 16 dimensions can be robustly improved.

4 Conclusion

We introduced a Distortion-Adaptive-Classification approach, which considers the position of a patch (which corresponds to the distortion level) in the original image for choosing patches for classification. For most features with a traceable dimensionality, the achieved classification rates are highly competitive. Whereas the traditional

classification of barrel-type distortion corrected images leads to benefits only in few cases (especially if lower image frequencies are considered), with DAC in most cases an increased discriminative power can be observed. With very high dimensional features (especially LTP with 8 neighboring samples), we do not achieve systematically improvements. We anticipate an even more advantageous behavior (even for high dimensional features) in the case of a larger image database.

Acknowledgment. This work is partially funded by the Austrian Science Fund (FWF) under Project No. 24366.

References

1. Liedlgruber, M., Uhl, A., Vécsei, A.: Statistical analysis of the impact of distortion (correction) on an automated classification of celiac disease. In: Proc. of the 17th Intern. Conf. on Digital Signal Processing, DSP 2011, Corfu, Greece (2011)
2. Gschwandtner, M., Liedlgruber, M., Uhl, A., Vécsei, A.: Experimental study on the impact of endoscope distortion correction on computer-assisted celiac disease diagnosis. In: Proc. of the 10th Intern. Conf. on Information Technology and Applications in Biomedicine, ITAB 2010 (2010)
3. Gschwandtner, M., Hämerle-Uhl, J., Höller, Y., Liedlgruber, M., Uhl, A., Vécsei, A.: Improved endoscope distortion correction does not necessarily enhance mucosa-classification based medical decision support systems. In: Proc. of the Intern. Workshop on Multimedia Signal Processing, MMSP 2012, pp. 158–163 (2012)
4. Häfner, M., Liedlgruber, M., Puespoek, A., Thomas, S., Schoefl, R., Wrba, F., Gangl, A., Uhl, A.: Computer-assisted pit pattern analysis of colonic lesions. Gastrointestinal Endoscopy 63, AB247–AB247 (2006)
5. Melo, R., Barreto, J.P., Falcao, G.: A new solution for camera calibration and real-time image distortion correction in medical endoscopy-initial technical evaluation. IEEE Trans. Biomed. Eng. 59, 634–644 (2012)
6. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distribution. Pattern Recognition 29, 51–59 (1996)
7. Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: Lee, S.-W., Li, S.Z. (eds.) ICB 2007. LNCS, vol. 4642, pp. 828–837. Springer, Heidelberg (2007)
8. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 168–182. Springer, Heidelberg (2007)
9. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. on Systems, Man, and Cybernetics 3, 610–621 (1973)
10. Gadermayr, M., Liedlgruber, M., Uhl, A.: Shape curvature histogram: A shape feature for celiac disease diagnosis. Technical Report 2012-06, Dept. of Computer Sciences, University of Salzburg, Austria (2012), <http://www.cosy.sbg.ac.at/research/tr.html>
11. Hegenbart, S., Kwitt, R., Liedlgruber, M., Uhl, A., Vécsei, A.: Impact of duodenal image capturing techniques and duodenal regions on the performance of automated diagnosis of celiac disease. In: Proc. of the 6th Intern. Symp. on Image and Signal Proc. and Analysis, ISPA 2009, pp. 718–723 (2009)
12. Oberhuber, G., Granditsch, G., Vogelsang, H.: The histopathology of celiac disease: time for a standardized report scheme for pathologists. European Journal of Gastroenterology and Hepatology 11, 1185–1194 (1999)

Moving Horizon Estimation of Pedestrian Interactions Based on Multiple Velocity Fields^{*}

Ana Portelo¹, Sandra Pacheco¹, Mário A.T. Figueiredo^{2,4},
João M. Lemos^{1,4}, and Jorge S. Marques^{3,4}

¹ INESC-ID,
² IT,
³ ISR,
⁴ IST,
Portugal

Abstract. This paper describes a model for the interaction between pedestrians in a scene. We consider a recently proposed model for isolated pedestrians in the scene and extend it by adding an interaction term that accounts for attractive/repulsive behaviors among pedestrians. The proposed model combines multiple velocity fields that represent typical motion regimes in the scene and a time-varying interaction term. The estimation of the active velocity field and interaction parameters is achieved by assuming that they remain constant within every instance of an analysis window that slides in time. This strategy is known as the Moving Horizon Estimation (MHE) method. The proposed algorithm is assessed both by using synthetic data and pedestrian trajectories extracted from video streams.

1 Introduction

The analysis of pedestrian interactions in outdoor scenes is a challenging problem with applications in video surveillance [1]. It aims to detect the presence of interacting pedestrians and to recognize the type of interaction. Examples of activities are: people meeting, walking together, stop and go or persecuting. A typical approach to tackle this problem consists of extracting the pedestrian trajectories from the video signal. The trajectories are then analyzed using a probabilistic motion model learned from the video data [2][3]. The model should be able to discriminate non-interacting pedestrians from interacting ones and should also provide information for characterizing the interaction type.

This paper considers a recently proposed model that describes the motion of isolated pedestrians using multiple velocity fields [4][5]. Each pedestrian trajectory is assumed to be a concatenation of trajectory segments, generated by

* This work was supported by FCT in the framework of contract PTDC/EEA-CRO/098550/2008, PEst-OE/EEI/LA0009/2011 and PEst-OE/EEI/LA0021/2011.

different velocity fields learned from the video data. The model allows probabilistic switching between velocity fields according to switching probabilities that depend on the pedestrian position in the scene. This model describes the movement of non-interacting pedestrians. To account for pedestrian interactions, the model is extended with an interaction term that describes attractive/repulsive behaviors between pairs of pedestrians. This term is inspired in the social force model proposed by Helbing and Molnar [6],[7] that has been adopted in several works (*e.g.*, see [8][9]).

The paper has two main contributions:

- it extends the multiple velocity fields model to deal with pedestrian interactions. The interaction between pairs of pedestrians is described by attractive/repulsive terms;
- describes a moving horizon algorithm to estimate the active field driving each trajectory and interaction parameters at each instant of time.

The paper is organized as follows. Section 2 describes the interaction model. Section 3 describes the parameter estimation method. Section 4 presents experimental results and section 5 draws conclusions.

2 Motion Model with Interaction

The multiple motion field model proposed in [5], [4], assumes that we know K velocity fields $\mathbf{T}_k : [0, 1]^2 \rightarrow \mathbb{R}^2$, for $k \in \{1, \dots, K\}$ that represent typical motion regimes of a pedestrian in the scene. For the sake of simplicity the image domain is assumed to be $[0, 1]^2$. This model assumes that only one motion field is active at each instant of time. For a single pedestrian, its trajectory is assumed to be generated by

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{T}_{k(t)}(\mathbf{x}(t)) + \mathbf{w}(t), \quad (1)$$

where $\mathbf{x}(t)$ denotes the pedestrian position at the discrete time t , $k(t)$ is the label of the active field and $\mathbf{w}(t)$ is a sequence of random and uncorrelated displacements, with normal distribution, $\mathbf{w}(t) \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$.

Equation (1) describes the motion of isolated pedestrians in the scene and does not consider the interaction between pedestrians. To overcome this gap, we will consider an extended version of the model that includes an attraction/repulsion term between pairs of interacting pedestrians. The generation of trajectories for two pedestrians 1 and 2 that interact is done according to (see Fig. 1)

$$\mathbf{x}^{(1)}(t+1) = \mathbf{x}^{(1)}(t) + \mathbf{T}_{k^{(1)}(t)}(\mathbf{x}^{(1)}(t)) + \alpha^{(1)} \phi(t) + \mathbf{w}^{(1)}(t) \quad (2)$$

$$\mathbf{x}^{(2)}(t+1) = \mathbf{x}^{(2)}(t) + \mathbf{T}_{k^{(2)}(t)}(\mathbf{x}^{(2)}(t)) - \alpha^{(2)} \phi(t) + \mathbf{w}^{(2)}(t), \quad (3)$$

where $\mathbf{x}^{(i)}$, $i = 1, 2$, denotes the position of pedestrian i ,

$$\phi(t) = \frac{\mathbf{x}^{(2)}(t) - \mathbf{x}^{(1)}(t)}{\|\mathbf{x}^{(2)}(t) - \mathbf{x}^{(1)}(t)\|}, \quad (4)$$

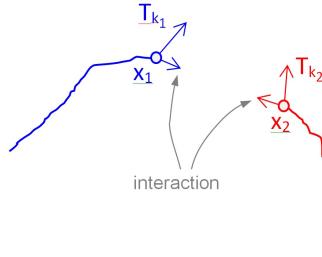


Fig. 1. Trajectory generation with velocity fields and interaction displacements. In the absence of the interaction term, the trajectory of pedestrian i is aligned with the active field $\mathbf{T}_{k^{(i)}}$. The interaction term modifies this movement.

is a unit vector pointing from pedestrian 1 to pedestrian 2; and $\alpha^{(1)}, \alpha^{(2)}$ are the interaction parameters: a positive $\alpha^{(i)}$ represents attraction while negative $\alpha^{(i)}$ represents repulsion. For $\alpha^{(i)} = 0$ there is no interaction.

An alternative would be to take $\phi(t) = \mathbf{x}^{(1)} - \mathbf{x}^{(2)}$. This has the drawback of leading to interaction terms that, when estimated, are larger when the distance between pedestrians grows, a feature that is undesirable. The choice of (4) has a singularity when the distance between the pedestrians vanish, but this can be dealt with without difficulty.

3 Parameter Estimation

Given a pair of trajectories $\mathbf{x}^{(1)}(t), \mathbf{x}^{(2)}(t), t = 1, \dots, T$, the objective is to estimate the evolution of the interaction parameters $\alpha^{(1)}(t), \alpha^{(2)}(t)$. These parameters can be independently estimated since (2, 3) are decoupled, provided $\phi(t)$ is known. However, the $\alpha^{(i)}(t)$ estimate depends on the active field $k^{(i)}(t)$ that drives the motion. Therefore, the two variables $\alpha^{(i)}(t), k^{(i)}(t)$ are strongly coupled and should be jointly estimated.

We will assume that $\alpha^{(i)}(t)$ changes sparsely along time and the label $k^{(i)}(t)$ has a small number of transitions. Therefore, we will adopt a sliding interval (moving horizon) with length H and assume that these two variables are constant within this time interval. For each pedestrian, the energy of the residue is computed in a sliding time window of length H extending to the past between the current time t_0 and $t_0 - H + 1$

$$E_{t_0}(\alpha, k) = \sum_{t=t_0-H+1}^{t_0} \|\mathbf{y}(t) - \alpha\phi(t-1)\|^2, \quad (5)$$

where $\mathbf{y}(t) = \mathbf{x}(t) - \mathbf{x}(t-1) - \mathbf{T}_k(\mathbf{x}(t-1))$. The index $i = 1, 2$ was dropped for the sake of simplicity. A change in the sign of $\phi(t-1)$ in (5) must also be done when $i = 2$.

The minimization of (5) with respect to α leads to a closed form expression

$$\hat{\alpha}(k) = R^{-1}r \quad (6)$$

where

$$R = \sum_{t=t_0-H+1}^{t_0} \phi(t-1)^T \phi(t-1), \quad r = \sum_{t=t_0-H+1}^{t_0} \phi(t-1)^T \mathbf{y}(t). \quad (7)$$

3.1 Joint Minimization

Two optimization strategies are considered. The first method is based on the optimization of $E_{t_0}(\alpha, k)$ with respect to k and α , simultaneously,

$$(\hat{\alpha}, \hat{k}) = \arg \min_{\alpha, k} E_{t_0}(\alpha, k). \quad (8)$$

Since a closed form expression for α is available, we can replace it by its optimal estimate $\hat{\alpha}$ defined in (5) and minimize $E(\hat{\alpha}(k), k)$ with respect to k . This is a straightforward step since we only need to compute the energy associated to each motion field $k \in \{1, \dots, K\}$ and choose the smallest. Although this approach leads to the global minimum of E , it does not always lead to meaningful estimates of the unknown parameters. Indeed, since the model is too flexible, there are several ways to explain the data *i.e.*, sometimes we can explain the observations $\mathbf{x}(t)$ using different velocity fields and use the interaction term to compensate the mismatch. Therefore we adopted an alternative approach described in the sequel.

3.2 Hierarchical Minimization

In order to tackle the identifiability problem mentioned in section 3.1, it is assumed that the motion field is the main responsible for explaining the observations. The interaction term is used only when the motion field model is not enough to describe the trajectories. Therefore, we estimate the motion field first, assuming no interaction

$$\hat{k} = \arg \min_k E_{t_0}(\alpha = 0, k). \quad (9)$$

After obtaining \hat{k} , we estimate α by solving the following optimization problem

$$\hat{\alpha} = \arg \min_{\alpha} E_{t_0}(\alpha, \hat{k}). \quad (10)$$

This procedure can be done analytically using (6) with \hat{k} estimated from (9). The whole algorithm is very fast and can be easily speed up by recursively computing R and r . The above hierarchical approach proved to be much more robust than the joint minimization procedure described in section 3.1.

After computing $\hat{\alpha}, \hat{k}$ for the instant t_0 we shift the analysis window by one sample (or more) and repeat the procedure. This technique is inspired in the *Moving Horizon Estimation (MHE)* method used in state estimation problems [10].

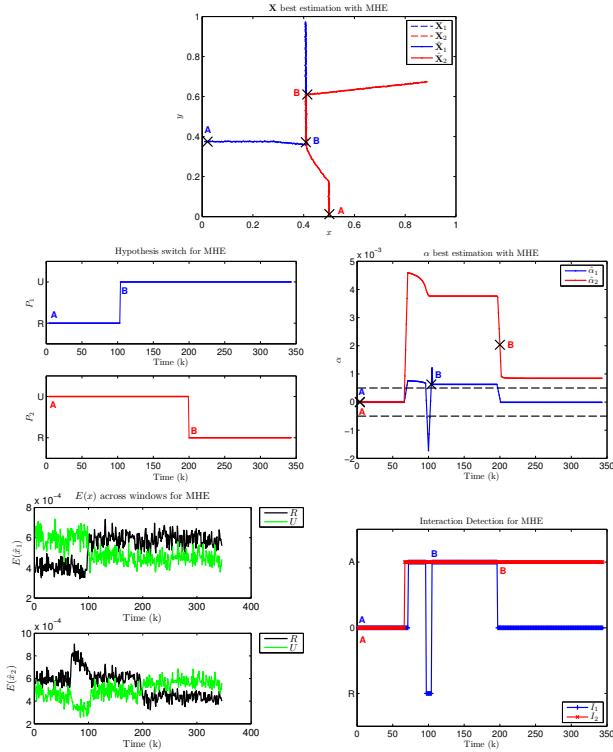


Fig. 2. Example with synthetic data: interacting trajectories (1st row); label and interaction parameter estimates (2nd row); cost functional and interaction intervals

4 Results

The proposed model and hierarchical MHE estimation were tested with synthetic data and with pedestrian trajectories extracted from video signals. We will present selected examples from both sets of tests, in order to characterize the performance of the estimator.

4.1 Synthetic Data

First, the model was evaluated with synthetic data. In each experiment, a pair of trajectories was generated using (2) and (3) with two velocity fields: a vertical (up) velocity field and an horizontal (right) one. Then, we applied the MHE method to estimate the fields labels and the interaction parameters, for each trajectory. A typical example is shown in Fig. 2. The figure shows two synthetic trajectories generated by the model (1st row), the estimated labels and interaction parameters (2nd row) and the cost energy for each hypothesis as well as the intervals of significant attraction/repulsion interactions. Since we know the

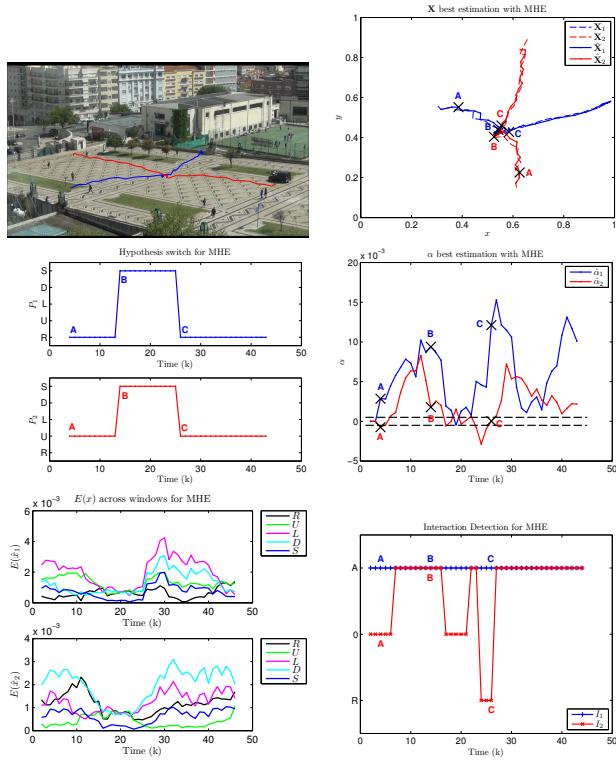


Fig. 3. Pedestrian interaction: interacting trajectories (1st row); label and interaction parameter estimates (2nd row); cost functional and interaction intervals

true labels and attraction parameters used to generate the data, we can compare these values with the estimates. It is concluded that, the estimates are close to the true values of the labels and interaction coefficients. The only exception is a spike that appears when the pedestrian direction (field) changes. When the field switches the estimates become unreliable since the stationarity condition that was assumed inside the analysis interval becomes invalid.

This example shows that the MHE method performs well with synthetic data, except in the vicinity of transition (switching) instants between active motion fields.

4.2 Pedestrian Interaction

The MHE algorithm was also applied in the analysis of pedestrian activities in an university campus. The video signal was acquired using a camera Sony HDR-CX260 with a resolution of 8.9 megapixels per frame and working at a frame rate of 30 frames per second. A tracking algorithm was used to extract the pedestrians trajectories. This procedure was done using a background

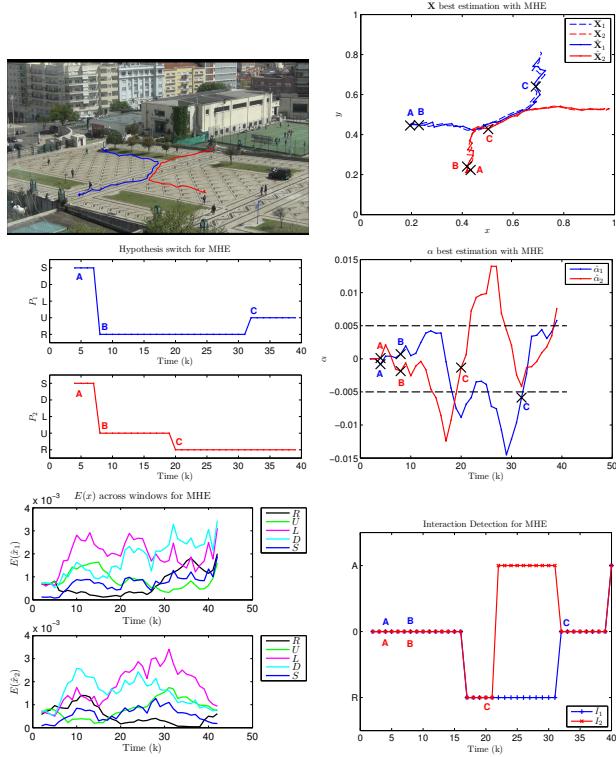


Fig. 4. Pedestrian interaction: interacting trajectories (1st row); label and interaction parameter estimates (2nd row); cost functional and interaction intervals

subtraction algorithm for the detection of active regions, followed by matching active regions at consecutive frames using the assignment algorithm described in [11]. The trajectories were then subsampled at a frame rate of 1 frame per second and the association errors were corrected. Finally, the trajectories were warped to compensate for the perspective projection distortion.

Figures 3,4 show examples of interactions between pedestrians. In this setup, we assume five velocity fields: two vertical fields (up, down), two horizontal fields (left, right) and the null velocity field (stopped). The first row shows the pedestrian trajectories before and after the warp. The second row shows the label estimates with five admissible values: stopped (S), down (D), left (L), up (U) and right (R). The interaction parameters are shown on the right. Finally, the evolution of the energy cost is displayed in the third row.

Both examples exhibit robust estimates of the field labels. The label estimates provided by the MHE method are the ones we would expect by looking at the trajectories. The attraction/repulsion coefficients are harder to interpret. In the first example (Fig. 3), we observe an attraction between time marks A and B

as expected. Then, there is no interaction during the time interval in which the pedestrians are stopped. When they separate we observe a repulsive effect in the red trajectory which also makes sense. Finally, we observe a long term attraction effect after mark C, when the pedestrians go apart. In fact none of the trajectories follows an horizontal or vertical direction as predicted by the vector field model. So, the attraction coefficients are used to compensate these deviations. The evolution of the cost functional for each of the 5 velocity fields is shown in the third row.

The second example (Fig. 4) can be interpreted in a similar way. Two people are initially stopped, they start walking along horizontal and vertical directions, they meet, they walk together and then separate. The label estimation step performs well and provides field estimates that can be easily interpreted. The α coefficients are harder to interpret. There is an initial attraction in both trajectories when the pedestrians are approaching. The attraction remains during the interval in which they walk together. This is explained by the fact that their trajectories are not well explained by none of the velocity fields (although the field R is the closest). There is then a repulsive effect when the two pedestrians separate. The cost functional associated to each of the five motion fields is shown in the third row of the figure.

5 Conclusions

The interaction model proposed in this article provides an extension of the multiple velocity field model recently presented in [4], [5]. The model was modified to account for the interaction between pairs of pedestrians. The pedestrians trajectories are generated using two complementary mechanisms: i) multiple velocity fields estimated from the video data and ii) attractive/repulsive effects that model the interaction between pedestrians.

An estimation algorithm was provided to retrieve the label of the active model at each instant of time as well as the attraction/repulsion coefficients. This algorithm assumed that the model parameters are constant inside a moving horizon (analysis window). It is therefore denoted as a moving horizon estimation (MHE) method. The experimental results obtained with the MHE method show a robust estimation of the active field and consistent estimates of the interaction parameters that are, however, harder to interpret since they model not only the interaction among pedestrians but also the deviation of the trajectories with respect to the nominal velocity fields.

References

1. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: A survey. *IEEE Trans. on Circuits and Systems for Video Technology* 18(11), 1473–1488 (2008)
2. Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22(8), 831–843 (2000)

3. Suk, H.-I., Jain, A., Lee, S.-W.: A network of dynamic probabilistic models for human interaction analysis. *IEEE Transactions on Circuits and Systems for Video Technology* 21(7), 932–945 (2011)
4. Nascimento, J., Figueiredo, M.A.T., Marques, J.: Activity recognition using mixture of vector fields. *IEEE Trans. on Image Processing* 22(5), 1712–1725 (2013)
5. Nascimento, J., Marques, J., Figueiredo, M.A.T.: Classification of complex pedestrian activities from trajectories. In: *IEEE Int. Conf. Image Processing*, pp. 3481–3484 (September 2010)
6. Helbing, D.: A mathematical model for the behavior of pedestrians. *Behavioral Science* (36), 298–310 (1991)
7. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. *Physical Review* (E51), 4282–4286 (1995)
8. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 935–942 (June 2009)
9. Zhang, Y., Qin, L., Yao, H., Huang, Q.: Abnormal crowd behavior detection based on social attribute-aware force model. In: *IEEE Int. Conf. Image Processing*, pp. 2689–2692 (September 2012)
10. Alessandri, A., Baglietto, M., Battistelli, G.: Moving-horizon state estimation for non-linear discrete-time systems: New stability results and approximation schemes. *Automatica* 44, 1753–1765 (2008)
11. Veenman, C.J., Reinders, M.J.T., Backer, E.: Resolving motion correspondence for densely moving points. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23, 54–72 (2001)

Evaluating and Comparing of 3D Shape Descriptors for Object Recognition

Alexander Ceron^{1,2} and Flavio Prieto¹

¹ Universidad Nacional de Colombia - Sede Bogotá, Carrera 30 No 45 - 03
Bogotá, Colombia

{aceronco,faprietoo}@unal.edu.co

² Universidad Militar Nueva Granada, Carrera 11 No. 101-80
Bogotá, Colombia

Abstract. In this work we show the results of a system for object recognition by using depth data. It is based on shape descriptors and PCA reduction. For obtaining the results, we evaluated different combination of three descriptors that are suitable for this work: Spin Images, VFH (Viewpoint Feature Histogram) and NARF (Normal Aligned Radial Feature). In addition, we created a method for extracting the NARF descriptor in order to obtain a global descriptor. The results show that the combination of descriptors can be used for object recognition in a database composed of point clouds obtained with a RGB-D sensor.

1 Introduction

The RGB-D sensors can acquire a lot of information about the morphology of objects that can be extracted by using 3D shape descriptors. With this information, it is possible to differentiate objects and recognize them. This is very important for developing a machine vision system for personal robots or in the automation of industrial processes.

The shape descriptors have been used in order to classify surfaces as planar or not and also to estimate if the surface is spherical, cylindrical or conic [1].

Some of these descriptors are based on the curvatures that are present in every 3D object. Their advantage relies on their invariance regarding the point of view. These descriptors can define eight fundamentals shapes of quadric surfaces.

Nevertheless, this kind of descriptors have not been good enough when working with information obtained from RGB-D low cost sensors like the Microsoft Kinect [2] because their computation could be expensive and their values inaccurate due to noise.

There is another kind of descriptors obtained through transformations from 3D information to 2D, like spin images [3] and their variations. These descriptors can be computed in different resolutions. They are invariant to the point of view too and have shown good results for object detection, but their computation could be expensive if the parameters are selected to obtain high resolution images or if the resolution of the point cloud is high.

In addition, the 3D information obtained with the Kinect is noisy. For this reason, it is more difficult to create a system for classification and recognition with this sensor than with more accurate sensors. There are some works that tackle this problem; in [4], a database and a recognition process that uses a combination of RGB information is presented.

Due to the fact that the mentioned descriptors have not presented good results with noisy point clouds from sensors like the Kinect and the long time for computing that it takes, recently a group of detectors and descriptors have been developed and have shown good results for discriminant capacity and featuring in short time.

This group of descriptors include: PFH (Point Feature Histograms), FPFH (Fast Point Feature Histograms) [5], VFH (Viewpoint Feature Histogram) [6] and NARF (Normal Aligned Radial Feature) [7].

In this work, we develop a method for computing the NARF descriptor that in combination with the VFH descriptor in a PCA (Principal Component Analysis) can be an input for a machine learning system that recognizes objects with point clouds obtained with a low cost RGB-D sensor like the Microsoft Kinect.

This paper is organized as follows: Section 2 presents the shape descriptors used, Section 3 gives a detailed view of the experiments in order to validate the process, Section 4 presents a complete view of the system, Section 5 presents the results and finally, Section 6 presents the conclusions and future work.

2 Shape Descriptors

In this work, we evaluate a set of descriptors that include Spin Images, VFH and NARF.

2.1 Spin Images

The spin image [3] is a local descriptor computed in an oriented point (p, n) (a point and its normal) that codifies two of the three coordinates of a cylindrical system in its neighbourhood. The spin image X for a point p in a surface is a 2D histogram where each pixel is a bin that stores the numbers of neighbors that are in a distance α from n and in a depth β from its tangent plane.

The α parameter can be estimated as the perpendicular distance from n to the x point. Where x is each point of the mesh or cloudfoint. The second parameter is the relative distance between n and x .

$$\begin{aligned}\alpha &= \sqrt{\|\mathbf{x} - \mathbf{p}\|^2 - (\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}))^2}, \\ \beta &= \mathbf{n} \cdot (\mathbf{x} - \mathbf{p}).\end{aligned}\quad (1)$$

2.2 Point Feature Histograms

In a simplified way, the computation of PFH (Point Feature Histograms) for a point p depends on the 3D coordinates and the normals of the estimated

surface, and it is computed as follows: i) For each point, all p neighbors inside a sphere of radius r are selected (k -neighbors); ii) For each couple of points p_i and p_j ($i \neq j$) of the k -neighbors of p and their estimated normals n_i and n_j (p_i is the point with the smallest angle between its associated normals and the line connecting the points), a Darboux reference framework uvn is defined as ($u = n_i, v = (p_j - p_i) \times u, w = u \times v$) and the angular variations of n_i and n_j in the following way:

$$\begin{aligned}\alpha &= v \cdot n_j \\ \phi &= (u \cdot (p_j - p_i))/d \\ \theta &= \arctan(w \cdot n_j, u \cdot n_j) \\ d &= \|p_j - p_i\|\end{aligned}\tag{2}$$

The principal purpose of this descriptor is to gather statistics for the relative angles between the normal surface to each point and the normal to the object centroid.

The structure composed of four values (α, ϕ, θ and d) is computed for each couple of points in the k -neighborhood.

The complexity of the PFH for a point cloud P with n points, is $O(nk^2)$, k is the number of neighbors.

The FPFH reduce the complexity to $O(nk^2)$. It is faster and has the same discriminant capacity of PFH [8].

2.3 Viewpoint Feature Histogram

The concept presented in FPFH evolved in order to include not only the shape information but also a point of view component. The view direction is composed of the relative angles α, ϕ and θ , but measured from the view direction [6].

2.4 Normal Aligned Radial Feature

NARF (Normal Aligned Radial Feature) is a technique for 3D feature extraction for range images that is invariant to rotation. This technique, in first instance, extracts interest points (keypoints) to create a shape feature descriptor. The process needs the computation of a normal aligned range value patch in the point. This is a small range image with the observer looking of the point along the normal. After that, a star pattern is superposed onto the patch,. Each beam corresponds to a value in the final descriptor that captures how much the pixels under the beam change. Finally, a unique orientation from the descriptor is extracted and the descriptor is shifted according to this value to make it invariant to the rotation.

2.5 NARF as a Global Descriptor

Like other detectors as SIFT [9] and SURF [10], NARF gives a sort of relevant key points in a scene with a descriptor for each one, but in this case with a

3D position. For this reason, we developed a process that allows us to convert NARF values in to a global descriptor.

The process is the following:

1. Computing the centroid of the point cloud.
2. Obtaining the longest distance from the centroid r .
3. Computing a number N of subdivisions of this distance r . This is $\delta r = r/N$
4. Computing a new radius $nr = \delta r$.
5. For $k = 1$ to N .
 - Computing the NARF descriptor for each keypoint inside the spheres of this radius.
 - Computing the mean of the descriptor in this region and store it as a $vector_i$.
 - Increasing the sphere radius nr by δr .
6. Concatenating all vectors obtained. This will be the new descriptor.

3 Experimental Setup

In this work we develop two tests. The first is done with the purpose of recognizing small objects. The second is for recognizing bigger objects. In our experiments, we use an Intel core i7 3.4 GHz with 8 GB of RAM, PCL (Point Cloud Library) [11] for point cloud descriptors extraction and OpenCV¹ functions for machine learning.

3.1 Small Object Recognition Test

In this test we use a established dataset [4]², which only contains small objects (objects that can be contained in a bounding box near to 12cm in each edge). In this case, the NARF descriptor cannot be computed because it requires a bigger support region. We use the Spin images and VFH descriptors.

Due to the fact that some datasets of 3D objects do not support the amount of information required to extract some descriptors or their inherent noise make it difficult to get important information, sometimes it is necessary to select clouds of points or meshes that can be used in order to extract coherent values for all descriptors required for the experiment.

After data validation, we took the following seven categories of elements of the mentioned database: apple, ball, banana, bowl, calculator, cell phone and dry battery.

For each of one of these categories, there are subsets because the dataset is hierarchical. In our case, we took 40 elements of each subset for training and

¹ <http://opencv.org/>

² <http://www.cs.washington.edu/rgbd-dataset>

other 10 for testing. The same process was followed with the rest of the elements obtaining 840 objects in total.

For each object, we obtain 308 values of the VFH and 153 of the Spin Images according with the PCL implementation. Our intention is to compare the performance of each one of these descriptors with their combination for object recognition.

3.2 Office Object Recognition Test

For the second tests, we create a new dataset of bigger objects (they occupy a bounding box bigger than 30cm in each edge) that are not segmented; this descriptor can be obtained without much difficulty. In this case, the database is composed by the following seven categories: computer, chair_1, chair_2, desk, microwave, printer and screen as it is shown in Figure 1.

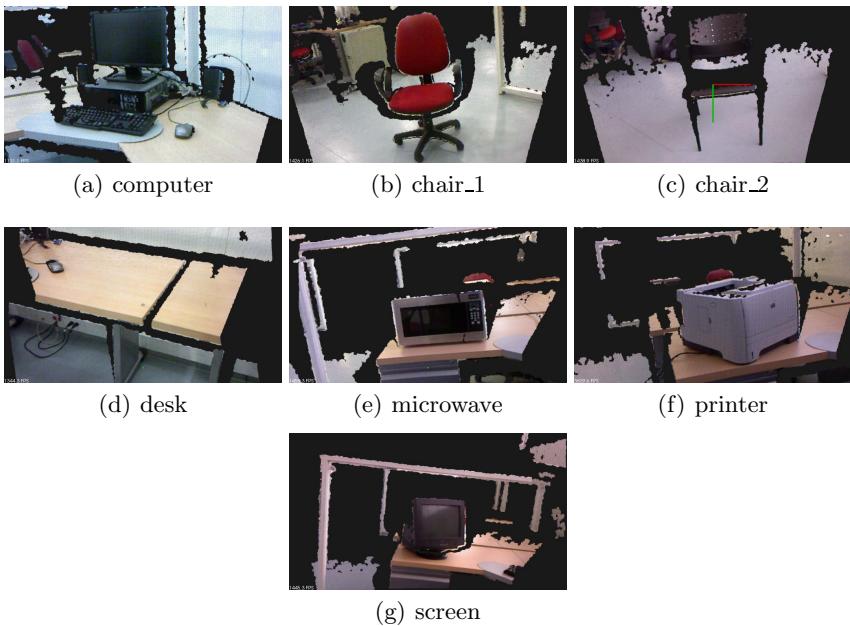


Fig. 1. Office object dataset examples

For each object, we took 50 point clouds in a distance of one meter from different points of view around the object. In this dataset the objects are not segmented and are part of the scene.

For this test, we used VFH and NARF descriptors because the spin images descriptor took too much time to compute.

4 Object Recognition System

To accomplish the recognition task, we use PCA (Principal Component Analysis), NN (Neural Networks) and lSVM (linear Support Vector Machine).

4.1 Feature Reduction

As the PCA has shown good results for reducing the space feature, we used this method in our work.

In the first experiment, we have a vector of 308 elements corresponding to the VFH descriptor concatenated with a vector of 153 elements of the Spin Images obtaining 461 values.

The reduction of the length of the input vector permits to reduce the time for the training process in the machine learning system and also to select the most relevant information for the learning process.

For these experiments, we evaluate different cases of PCA reductions in order to find the most suitable.

4.2 Classification

In this work, we use a NN and a linear SVM (lSVM) by using OpenCV in order to accomplish the first experiment.

For the NN, we evaluate several sizes. First, we use 50 neurons in the intermediate label, then 100, then 150 and finally 200. For the other case, we use a lSVM. In this case, the output is a label that contains categories.

We use 80 % of data for training and 20 % for testing.

5 Results

5.1 First Experiment

As we can see in Table 1, the classification results obtained with the SVM are better than NN in the classification process. In addition, the training process for the NN is near six seconds and the for lSVM is near two seconds. For these reasons, we think that this learning method is more suitable for real time systems.

On the other hand, the performance is not good enough when we make a PCA selection of 10 and when the selection is longer than 40 eigenvalues it gets worse. The recognition results of the combination of VFH and spin images are not better than with VFH. One reason could be that the support surface is too small to obtain enough relevant information to create the spin images.

From the computational cost point of view, the VFH descriptor presents a shorter computation time than spin images. Another disadvantage of spin images is that it requires setting up more parameters than VFH to improve the classification results. This makes it difficult to extract the spin images from a big dataset because there are objects that require different parameters.

Table 1. Classification results for the first experiment

	PCA Method	Spin Image	VFH	VFH+SPIN
10	NN_{50}	26.1 %	30.4 %	30.5 %
	NN_{100}	30.4 %	40.0 %	40.0 %
	NN_{150}	23.8 %	41.4 %	41.4 %
	NN_{200}	26.1 %	30.9 %	24.3 %
	SVM	58.0 %	70.0 %	69.5 %
20	NN_{50}	27.1 %	20.0 %	22.8 %
	NN_{100}	26.1 %	25.7 %	24.3 %
	NN_{150}	22.8 %	28.5 %	24.3 %
	NN_{200}	26.7 %	26.6 %	28.6 %
	SVM	58.0 %	76.6 %	76.1 %
30	NN_{50}	28.5 %	24.2 %	21.9 %
	NN_{100}	23.3 %	23.3 %	28.0 %
	NN_{150}	31.9 %	25.7 %	17.1 %
	NN_{200}	21.4 %	33.8 %	28.6 %
	SVM	58.0 %	75.7 %	75.7 %
40	NN_{50}	27.1 %	28.0 %	24.3 %
	NN_{100}	26.1 %	25.7 %	29.0 %
	NN_{150}	16.7 %	28.0 %	33.3 %
	NN_{200}	27.1 %	24.2 %	33.4 %
	SVM	58.0 %	76.2 %	76.2 %
50	NN_{50}	28.6 %	23.8 %	27.6 %
	NN_{100}	25.7 %	23.8 %	28.5 %
	NN_{150}	21.4 %	23.3 %	29.0 %
	NN_{200}	31.4 %	26.2 %	26.1 %
	SVM	58.0 %	75.2 %	75.2 %

For this reason the VFH overcomes the spin image in our test and is recommended for real time systems.

5.2 Second Experiment

For this case, we evaluate different amount of eigenvalues 10, 20, 30, 40, 50 and 60. The results for this experiments are shown in the Table 2. We observe that the best classifier is SVM again and is it fastest.

It is evident that the best PCA reduction is 30. Also, we can observe that an increment of the eigenvalues for PCA does not increase the performance.

The most important result is the fact that the performance is improved when the descriptors are combined for all PCA selection.

In addition, it is worth noting that the value of the performance of the descriptor proposed is not much greater than VFH, but it is faster than VFH.

Table 2. Classification results for the second experiment

PCA Method		VFH	NARF	VFH+NARF
10	NN_{50}	15.7 %	17.1 %	21.4 %
	NN_{100}	15.7 %	15.7 %	17.1 %
	NN_{150}	24.2 %	15.7 %	14.2 %
	NN_{200}	21.4 %	20.0 %	21.4 %
	SVM	62.8 %	55.7 %	71.4 %
20	NN_{50}	31.4 %	14.3 %	18.6 %
	NN_{100}	27.1 %	21.4 %	18.6 %
	NN_{150}	24.2 %	14.3 %	20.0 %
	NN_{200}	28.5 %	14.3 %	24.3 %
	SVM	62.8 %	54.3 %	81.4 %
30	NN_{50}	24.2 %	15.7 %	31.4 %
	NN_{100}	27.1 %	17.1 %	25.7 %
	NN_{150}	18.5 %	15.7 %	21.4 %
	NN_{200}	22.8 %	17.1 %	25.7 %
	SVM	67.1 %	52.8 %	82.8 %
40	NN_{50}	22.8 %	15.7 %	24.3 %
	NN_{100}	20.0 %	17.1 %	24.3 %
	NN_{150}	21.4 %	18.5 %	25.7 %
	NN_{200}	17.1 %	20.0 %	21.4 %
	SVM	65.7 %	54.3 %	81.4 %
50	NN_{50}	24.2 %	17.1 %	17.1 %
	NN_{100}	18.5 %	15.7 %	27.1 %
	NN_{150}	22.8 %	14.3 %	21.4 %
	NN_{200}	32.8 %	15.7 %	15.7 %
	SVM	65.7 %	57.1 %	80.0 %
60	NN_{50}	21.4 %	24.3 %	21.4 %
	NN_{100}	24.3 %	21.4 %	24.3 %
	NN_{150}	28.6 %	22.8 %	17.1 %
	NN_{200}	21.4 %	20.0 %	18.6 %
	SVM	65.7 %	57.1 %	80.0 %

6 Conclusions and Future Work

In this work, we create a new way for computing a global descriptor based on the NARF descriptor that shows interesting results. This descriptor takes less time to compute than spin images and VFH.

Our first experiment shows that there are cases when a combination of descriptors (spin images and VFH) does not represent an increase in performance. It is enough to use VFH.

On the other hand, there are other cases when descriptors that are used together present better results compared to when they are used separately as

the second experiment. It can be observed when the VFH is used with the NARF, the results in most of cases are better than use one of them for a SVM classifier.

We think that wrong classifications are due to a small support region obtained in the point clouds for some objects because the descriptor values become noisy and without discriminant information. This is more common in small objects.

In the case of the office objects classification, we did not use segmented point clouds, we expect to obtain better results with segmented objects as a future work.

The SVM has shown better results for classification than NN in most of the cases and takes less time to train; for these reasons, it could be suitable for real time systems like personal robotics or for industrial processes.

In a next stage, we expect to find different ways to improve the performance of the descriptor proposed. The approach used permits to add and evaluate other kind of descriptors and possibilities of combination. Also, we would like to include 2D visual features to the feature vector in order to evaluate a more complete vision system. We expect to integrate these results in a robotic platform that allows object recognition in indoor environments as a future work.

References

1. Flynn, P., Jain, A.: Surface classification: Hypothesis testing and parameter estimation. In: Computer Vision and Pattern Recognition (1988)
2. Kinect, M.: <http://www.xbox.com/en-us/kinect>
3. Johnson, A.: Spin-Images: A Representation for 3D Surface Matching. PhD thesis, Robotics Institute, Carnegie Mellon University (1997)
4. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgbd object dataset. In: 2011 IEEE International Conference on Robotics and Automation, ICRA, pp. 1817–1824 (2011)
5. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (fpfh) for 3d registration. In: The IEEE International Conference on Robotics and Automation, ICRA (2009)
6. Rusu, R., Bradski, G., Thibaux, R., Hsu, J.: Fast 3d recognition and pose using the viewpoint feature histogram. In: Intelligent Robots and Systems, IROS (2010)
7. Steder, B., Rusu, R.B., Konolige, K., Burgard, W.: Narf: 3d range image features for object recognition. In: Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS (2010)
8. Rusu, R.B., Holzbach, A., Blodow, N., Beetz, M.: Fast geometric point labeling using conditional random fields. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009 (2009)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60, 91–110 (2004)
10. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
11. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation, ICRA, Shanghai, China (2011)

Gender Recognition Using Fusion of Local and Global Facial Features

Anwar M. Mirza¹, Muhammad Hussain², Huda Almuzaini², Ghulam Muhammad¹,
Hatim Aboalsamh², and George Bebis^{2,3}

¹ Department of Computer Engineering,

² Department of Computer Science, College of Computer and Information Sciences,
King Saud University, Riyadh 11543, Saudi Arabia

³ Department of Computer Science and Engineering, University of Nevada at Reno, USA

Abstract. Human perception of the face involves the observation of both coarse (global) and detailed (local) features of the face to identify and categorize a person. Face categorization involves finding common visual cues, such as gender, race and age, which could be used as a precursor to a face recognition system to improve recognition rates. In this paper, we investigate the fusion of both global and local features for gender classification. Global features are obtained using the principal component analysis (PCA) and discrete cosine transformation (DCT) approaches. A spatial local binary pattern (LBP) approach augmented with a two-dimensional DCT approach has been used to find the local features. The performance of the proposed approach has been investigated through extensive experiments performed on FERET database. The proposed approach gives a recognition accuracy of 98.16% on FERET database. Comparisons with some of the existing techniques have shown a marked reduction in number of features used per image to produce results more efficiently and without loss of accuracy for gender classification.

Keywords: Fusion of features, principal component analysis (PCA), discrete cosine transformation (DCT), local binary pattern (LBP) approach, FERET database.

1 Introduction

Categorization of facial images, before passing it on to a face recognition system, can lead to significant improvements in the performance of the recognition system. This categorization can be achieved using visual cues like gender, race and age[1]. The notion of humans utilizing visual cues for face recognition is supported by a number of studies. Brigham [2], O'Toole et al. [3] and Philips et al. [4] have shown that people are more accurate at recognizing faces from their own race than faces from other races. Cheng et al. [5] and Baudouin et al. [6] have demonstrated that human use gender information along with other face categories for face recognition purposes. Gender recognition is important for its use in numerous applications including identity authentication, demographic data collection, search engine retrieval accuracy, human-computer interaction, access control, and surveillance.

Yamaguchi et al. [7] and Wild et al. [8] created prototypical male and female faces using a morphing technique. Their studies showed global structural differences between prototypical boy and prototypical girl faces. This clearly indicates that the global structural features of the face play an important role in the process of face categorization according to gender.

Determination of the salient features from facial images is one of the most important steps in the process of face and category specific face recognition. If the extracted features do not help in discriminating between different facial images, the use of an excellent classification algorithm can lead to very bad results. The feature extraction can be subdivided into either appearance based or geometric methods. Features including distance between eyes; size and location of the eyes, nose, mouth and ears; overall size of the face etc. are used in the geometric methods. In appearance based methods, features are extracted from the whole facial image.

SEXTNET was the first application based on the artificial neural networks to identify gender from face images [9]. A small database of 90 face images of 45 male and 45 female images was used in this study to train a 2-layer fully connected neural net for classification purposes. Using geometrical features as an input, Brunelli et al. [10] developed a radial basis function (RBF) based competing network for gender classification. They reported an accuracy of 79% in their work. Gutta et al. [11] proposed a hybrid approach based on the use of RBFs and decision trees for gender classification. A very important contribution was made by Moghaddam et al. [12] when they employed support vector machine (SVM) for gender classification and reported a misclassification rate of 3%. Yang et al. [13] improved gender classification using texture normalization. Several weak classifiers were combined using AdaBoost by Baluja and Rowley [14] in their gender classification system. They used small sized normalized images from FERET database. Their system showed an overall 90% recognition rate. Lu and Shi [15] used the fusion of left eye, upper face region and nose in their study. They showed that their fusion approach outperforms the whole face approach. This work was later extended by Alexandre [16] to combine shape and texture features from differed sized normalized images. They used local binary pattern (LBP) for the texture features.

The rest of the paper is organized as follows. Section 2 presents an overview of the feature extraction techniques. This section describes the mathematical foundations of the global and local feature extraction techniques used in this paper. In Section 3 the proposed methodology adopted in this paper is described. Experimental results and discussion are given in section 4. Concluding remarks about the present work are given in section 5.

2 Feature Extraction Techniques

Zhao et al. [17], Sinha et al. [18] and Ng et al. [19] have given detailed reviews of some of the most important face recognition techniques in general and gender recognition from facial images in particular. Most of the commonly used techniques either rely on just the global features or local features. The process of gender recognition by

humans involves first scanning of the face and then observing the detailed features of the face. In this paper we employ both global and local features for gender recognition, that are fused together to form a compact feature representation for the facial images.

2.1 Global Feature Extraction

We have employed two global feature extraction techniques, namely, the eigenface approach based on the reconstruction of face images employing principal component analysis (PCA) and the two-dimensional discrete cosine transform (2D-DCT) approach.

In the eigenface approach, a face image $f(x, y)$ of size $R \times C$, is converted into a column vector X of size $N = R \times C$, by concatenating all the columns of $f(x, y)$. The image (now represented by) X is projected onto a low dimensional vector Y of size M , such that ($M < N$) using

$$Y = UX \quad (1)$$

where U is the projection matrix of size $M \times N$. The column vectors of the projection matrix U are the eigenfaces obtained from the PCA approach. The projected vector Y is the global feature vector in this case.

Let X_i be one sample face image in the training dataset of size L images. The mean is given by

$$\mu = \frac{1}{L} \sum_{i=1}^L X_i \quad (2)$$

The scatter matrix S for all sample images is obtained by

$$S = \sum_{i=1}^L (X_i - \mu)(X_i - \mu)^T \quad (3)$$

In the PCA approach, eigenvectors of the scatter matrix S are obtained. These eigenvectors are sorted according to the descending values of the eigenvalues of S . First M eigenvectors of S form the column vectors of the projection matrix U . From equation (1), it could be noticed that each face image X_i now can be represented by a lower dimensional feature vector Y_i .

In the second feature extraction approach we have used 2D-DCT of the sample face image matrix $f_i(x, y)$ having R rows and C columns, using the transformation

$$g_i(u, v) = \frac{2}{\sqrt{RC}} a(u)a(v) \sum_{x=0}^{R-1} \sum_{y=0}^{C-1} f_i(x, y) \cos\left(\frac{(2x+1)u\pi}{2R}\right) \cos\left(\frac{(2y+1)v\pi}{2C}\right) \quad (4)$$

where

$$a(u) = \begin{cases} \sqrt{1/R} & \text{for } u = 0 \\ \sqrt{2/R} & \text{for } u = 1, 2, 3, \dots, R-1 \end{cases} \quad (5)$$

and similarly for $a(v)$. The discrete cosine transformation has a high information packing ability. This is the reason for its use in image compression technologies. The most significant DCT coefficients of Y_i are chosen by determining which of the coefficients have the greatest variance. For natural images, the DCT coefficient's energy drops off very rapidly for higher frequency components. This compaction property is used in the dimension reduction for face images. After obtaining the transformed matrix $g_i(u, v)$, its coefficients are sorted according to the zigzag scanning technique of image compression [Gonzalez ref]. A dimensionally reduced feature vector based on 2D-DCT is obtained by choosing first M coefficients of Y_i after performing zigzag scanning.

2.2 Local Feature Extraction

Local binary pattern (LBP) descriptor computed using the LBP operator was first introduced by Ojala et al. [20]. It was initially used as a texture descriptor giving very promising results in many applications [21], [22], and [23]. Ahonen et al. [24] used it for the first time for face recognition. Sun et al. [25] and Lian et al. [26] extended its use to gender recognition using facial images. The initial LBP operator associates a label with each pixel of an image; the labeling process involves converting each pixel value in the 3×3 neighborhood of a pixel into a binary digit (0 or 1) using the center value as a threshold and concatenating the bits, as shown in Figure 1. Later the operator was extended to general neighborhood sizes, and its rotation invariant and uniform versions were introduced [21].

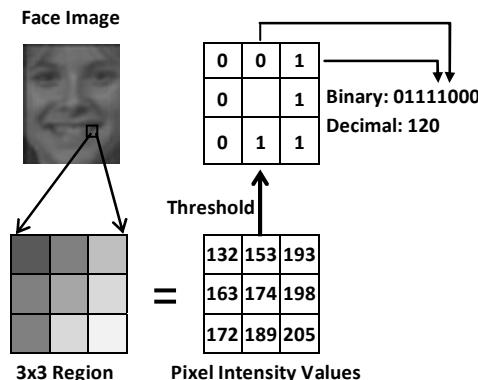


Fig. 1. Example of basic LBP operator

The general LBP operator is denoted by $LBP_{P,R}$ and is defined as:

$$LBP_{P,R} = \sum_{i=1}^{P-1} 2^i S(p_i - p_c) \quad (6)$$

where P is the total number of pixels in the neighborhood and R is its radius, p_c is the center pixel. The thresholding operation is defined as follows:

$$S(p_i - p_c) = \begin{cases} 1 & p_i - p_c \geq 0 \\ 0 & p_i - p_c < 0 \end{cases} \quad (7)$$

After applying the $LBP_{P,R}$ operation on the original image $f(x,y)$, a labeled age $f_l(x,y)$ is obtained. The histogram of the labels is used as a texture descriptor. The histogram $H(i)$ of the labeled image is defined as:

$$H(i) = \sum_{x=1}^R \sum_{y=1}^C I\{f_l(x,y) = i\}, \quad \text{for } i = 0, \dots, n-1 \quad (8)$$

where n is the number of different labels produced by the LBP operator and

$$I\{x\} = \begin{cases} 1 & x \text{ is true} \\ 0 & x \text{ is false} \end{cases} \quad (9)$$

General LBP operator has three parameters: circular neighborhood (P, R), rotation invariance (ri) and uniformity (u)[21]. For a particular application, it is necessary to explore this parameter space to come up with the best combination of these parameters.

Figure 2 shows the histogram extracted from an image with LBP operator. An LBP histogram in this approach contains information about facial micro-patterns like the distribution of edges, spots and flat areas over the whole image. In case of $(P, R) = (8, 1)$ neighborhood, there are 256 unique labels, and the dimension of LBP histogram descriptor is 256.

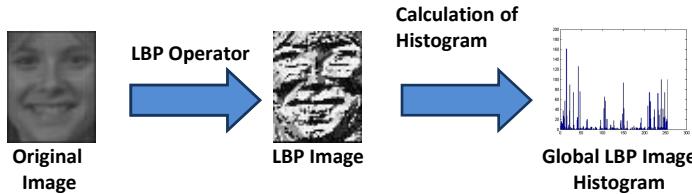


Fig. 2. LBP histogram calculation for the whole image

It could be noticed that, the basic LBP histogram descriptor obtained in this case is for the whole image and can be considered as a global descriptor for the image. The descriptor represents facial patterns but their spatial location information is lost due to the global histogram operation. We, however, have employed block by block version of the LBP labeled image (as explained in the proposed approach of section 4) which gives rise to a local representation of the facial features in the image.

3 Proposed Methodology

We followed the general steps of pre-processing, feature extraction, feature selection and classification of pattern recognition in our work. The main contribution of this

work is in the feature extraction stage of the overall procedure. The pre-processing stage consists of face normalization which crops out the background from each of the face image and centers it according to the location of the left and right eyes of the subject. For feature selection, we use the procedure given in the feature extraction part to form an overall feature vector of the most significant features for discriminating the faces according to gender.

A schematic diagram of our proposed methodology is shown in Figure 3. The input image, after the preprocessing step, passes through the feature extraction stage. We apply 2D-DCT or PCA on the whole image to obtain the global feature vector. Both these techniques are used for dimension reduction. Therefore, only the most significant NG feature vector values are selected after sorting the original global feature vector.

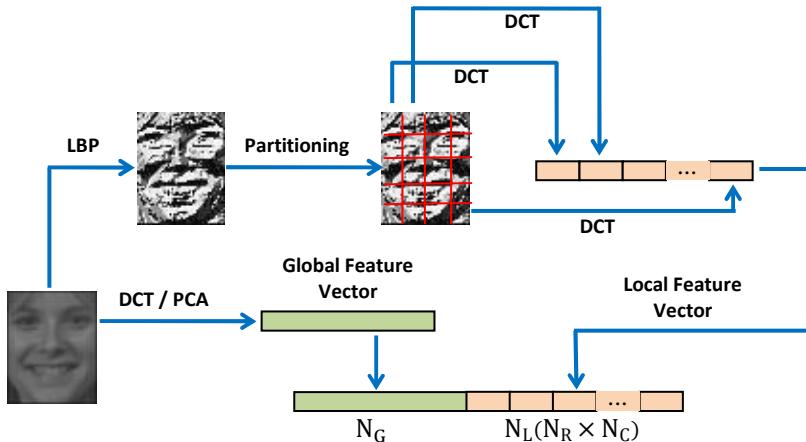


Fig. 3. Schematic diagram of the proposed methodology

For the local feature vector, we first apply LBP operator on the whole image. The labeled LBP obtained is partitioned into $NR \times NC$ blocks of pre-defined sizes. We then apply 2D-DCT on each of these LBP blocks to obtain the contribution of local features to the overall local feature vector. Only the most significant NL 2D-DCT components are selected after zigzag scanning of each of the resulting DCT coefficients. The block-wise local feature are concatenated to obtain the overall local feature vector. Specifically, the following steps are carried out in the feature extraction stage.

- Global Feature Extraction
 - All training and testing images are fed to the global feature extraction algorithm.
For each image, do the following steps.
 - Apply either 2D-DCT or PCA to obtain the global features.
 - Select NG global features to form the global feature vector FG .
- Local Feature Extraction
 - All training and testing images are fed to the local feature extraction algorithm.
For each image do the following steps.

- Apply the LBP algorithm to the whole image to obtain the LBP labeled image.
- Partition the LBP labeled image into $NR \times NC$ blocks.
- Apply the 2D-DCT on each of the block to get the DCT coefficients.
- Employ zigzag scanning function to select the low frequency DCT coefficients.
- Select only NL DCT coefficients from each block.
- Concatenate the selected DCT coefficients from each block to form the overall local feature vector FL of size $NL \times NR \times NC$ coefficients.
- Fusion of the Features
 - Combine the global and local feature vectors to form the overall feature vector for the whole image $F = FG, FL$.

We employ k-nearest neighbors (kNN) classifier to classify faces according to the overall feature vectors.

4 Experiments and Discussion

We have performed experiments on FERRET database [27], which is considered as a challenging database for face recognition. The database consists of frontal, left or right profile images and could have some variations in pose, expression and lightning. We used 1204 face images consisting of 746 male and 458 female images for training and 1196 images comprising 740 male and 456 female images for testing. Each image is normalized and cropped to the size **60 × 48** pixels. Some of the images taken from FERET database are shown below.



Different values for the parameters N_G and N_L have been used to check the performance of the algorithm. For the local feature extraction part, blocks of different sizes including 32×24 , 16×16 , 12×12 and 6×6 , have been used. Also LBP variants with uniform mapping, no mapping and $P = 8$, and $R = 1$ have been employed.

Our experiments have shown that the best performance for different block sizes is obtained for different combinations of the global and local features. Figure 4 shows the recognition rates for different block sizes. The lengths of the feature vectors (after performing fusion) were found to be 690, 512, 489 and 589 for blocks of sizes 32×24 , 16×16 , 12×12 and 6×6 , respectively. The overall best performance was given by block size 12×12 , yielding a recognition rate of 98.16% with 9 global 2D-DWT features and 24 local LPB+DCT features. A poorer performance was achieved in case of the use of PCA for global features. In all experiments a kNN classifier was used. Comparison of the performance evaluation with different distance measure from including city block, chi-square, cosine and Euclidean showed that the city block distance measure gives the best results in this case. It is noticed that decreasing the block size does not improve the performance of the system. In fact, the

number of features required to produce the best results for small size block also increases, resulting in more computations.

We also have compared our approach with the competing techniques including PCA (as a baseline approach), dyadic wavelets (DyWT) approach [28], histogram based LBP (LHBP) [25], and Multi-resolution Decision Fusion method (MDF) [16]. The results shown in Figure 5 indicate that our proposed system gives better recognition rates as compared to PCA, DyWT and HLBP approaches. The MDF approach however is reported to give better accuracies. However, it should be noticed that the proposed approach achieves 98.16% accuracy with fewer number of features as compared to MDF, which works at multiple scales (including 20×20 , 36×36 , and 128×128). It required extraction of shape as well as texture features for each of these scales and independent tuning of the classifier for each scale separately. Therefore, our proposed scheme comparable yields performance at a much lesser computational cost and better efficiency.

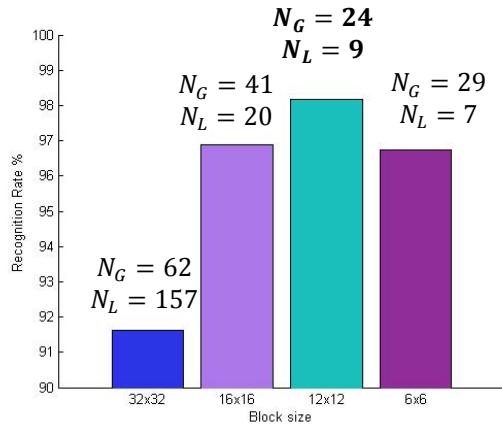


Fig. 4. Effect of the block size on the recognition rate

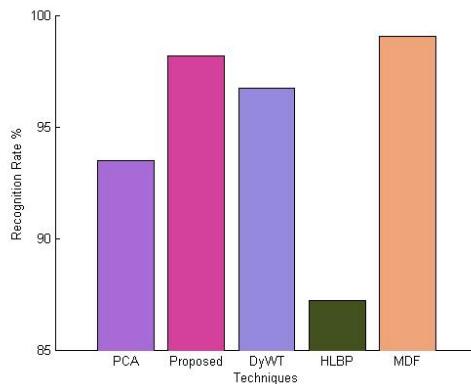


Fig. 5. Comparison of the proposed technique with the competing techniques

5 Conclusion

In this paper, we have presented a fusion strategy for the combination of local and global facial feature to address the gender recognition problem. The new approach make use selected 2D-DCT features for the overall face image. The local features are obtained by applying LBP approach on the facial image, which is then partitioned into blocks, followed by the application of 2D-DCT on each block and selection of the most significant features. The proposed methodology gives a recognition accuracy of 98.16% with only 489 features. We found that block size of 12×12 , with uniform mapping and neighborhood $(P, R) = (8,1)$ for LBP yield the best accuracy. A comparison of our proposed scheme with the competing techniques indicates a better performance in terms of accuracy as well as the number of feature vector length used in those techniques. In future, we plan to explore the use of this strategy for other face databases and sophisticated classifiers like SVM.

Acknowledgement. The work presented in this paper is supported by the National Plan for Science and Technology, King Saud University, Riyadh, Saudi Arabia under project number 10-INF1044-02.

References

1. Yang, S., Bebis, G., Hussain, M., Muhammad, G., Mirza, A.M.: Unsuper-vised discovery of visual face categories. International Journal on Artificial Intelligence Tools (May 2012) (accepted) doi: 10.1142/S0218213012500297
2. Brigham, J.: The influence of race on face recognition. In: Ellis, H., Jeeves, M., Newcombe, F. (eds.) *Aspects of Face Processing*, pp. 170–177 (1986)
3. O'Toole, A., Peterson, J., Deffenbacher, K.: An other-race effect for classifying faces by sex. *Perception* 25, 669–676 (1996)
4. Phillips, P.J., Jiang, F., Narvekar, A., Ayyad, J., O'Toole, A.: An other-race effect for face recognition algorithms. *ACM Transactions on Applied Perception* 8(2) (2011)
5. Cheng, Y.D., O'Toole, A., Abdi, H.: Classifying adults' and children's faces by sex: Computational investigations of subcategorical feature encoding. *Cognitive Science* 25(5), 819–838 (2001)
6. Baudouin, J.Y., Tiberghien, G.: Gender is a dimension of face recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 28(2), 362–365 (2002)
7. Yamaguchi, M.K., Hirukawa, T., Kanazawa, S.: Judgement of gender through facial parts. *Perception* 24, 563–575 (1995)
8. Wild, H., Barrett, S., Spence, M., O'Toole, A., Chenh, Y., Brooke, J.: Recognition and sex classification of adults' and children's faces: examining performance in the absence of sex-stereotypes cues. *Journal of Experimental Child Psychology* 77, 269–291 (2000)
9. Golom, A., Lawrence, D.T., Sejnowski, T.J.: SEXNET: A neural network identifies gender from human faces. In: *Advances in Neural Information Processing Systems*, vol. 3, pp. 572–577 (1991)
10. Brunelli, R., Poggio, T.: HyperBF network for gender classification. In: *DARPA Image Understanding Workshop*, pp. 311–314 (1992)

11. Gutta, S., Wechsler, H., Phillips, P.: Gender and ethnic classification of face images. In: 3rd IEEE International Conference on AUtomatic Face and Desture Recognition, FG 1998, pp. 194–199 (1998)
12. Moghaddam, B., Yang, M.-H.: Gender classification with support vector machines. In: Proc. IEEE International Conference on Automatic Face and Gesture Recognition, pp. 306–311 (2000)
13. Yang, Z., Li, M., Ai, H.: An experimental study on automatic face gender classification. In: Proc. IEEE Int. Conf. on Pattern Recognition, pp. 1099–1102 (2006)
14. Baluja, S., Rowley, H.: Boosting sex identification performance. International Journal of Computer Vision 71(1), 111–119 (2007)
15. Lu, L., Shi, P.: Fusion of multiple facial regions for expression-invariant gender classification. IEICE Electronics Express 6(10), 587–593 (2009)
16. Alexandre, L.A.: Gender recognition: A multiscale decision fusion approach. Pattern Recognition Letters 31(11), 1422–1427 (2010)
17. Zhao, W., Cellappa, R., Rosenfeld, A., Phillips, P.J.: Face Recognition: A Literature Survey. ACM Computing Surveys, 399–458 (2003)
18. Sinha, P., Balas, B., Ostrovsky, Y., Russell, R.: Face Recognition by Humans: 199 Results All Computer Vision Researchers Should Know About. Proceedings of the IEEE 94(11), 1948–1962 (2006)
19. Ng, C.B., Tay, Y.H., Goi, B.-M.: Recognizing Human Gender in Computer Vision: A Survey. In: Anthony, P., Ishizuka, M., Lukose, D. (eds.) PRICAI 2012. LNCS, vol. 7458, pp. 335–346. Springer, Heidelberg (2012)
20. Ojala, T., Pietikäinen, M., Harwood, D.: A Comparative Study of Texture Measures with Classification Based on Feature Distributions. Pattern Recognition 29(1), 51–59 (1996)
21. Ojala, T., Pietkainen, M., Maenpaa, T.: Multiresolution Gray-Scale and Rota-tion Invariant Texture Classification with Local Binary Patterns. IEEE Trans. Pattern Analysis and Machine Intelligence 24(7), 971–987 (2002)
22. Zhang, G., Huang, X., Li, S.Z., Wang, Y., Wu, X.: Boosting local binary pattern (LBP)-based face recognition. In: Li, S.Z., Lai, J.-H., Tan, T., Feng, G.-C., Wang, Y. (eds.) SINOBIOMETRICS 2004. LNCS, vol. 3338, pp. 179–186. Springer, Heidelberg (2004)
23. Liu, H., Sun, J., Liu, L., Zhang, H.: Feature selection with dynamic mutual information. Journal of Pattern Recognition 42(7) (July 2009)
24. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004, Part I. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
25. Sun, N., Zheng, W., Sun, C., Zou, C., Zhao, L.: Gender Classification Based on Boosting Local Binary Pattern. In: Wang, J., Yi, Z., Źurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 194–201. Springer, Heidelberg (2006)
26. Lian, H., Lu, B.: Multi-view gender classification using multi-resolution local binary patterns and support vector machines. International Journal of Neural Systems 17(6), 479–487 (2007)
27. Phillips, P.J., Hyeonjoon, M., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
28. Abdulkirim, T., Hussain, M., Niijima, K., Takano, S.: The Dyadic Lifting Schemes and the Denoising of Digital Images. International Journal of Wavelets, Multiresolution and Information Processing 6(3), 331–351 (2008)

Curvelet Transform and Local Texture Based Image Forgery Detection

Muneer H. Al-Hammadi¹, Ghulam Muhammad¹,
Muhammad Hussain¹, and George Bebis²

¹ College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia
Eng.muneer2008@gmail.com, {ghulam,mhussain}@ksu.edu.sa

² Department of Computer Science and Engineering, University of Nevada at Reno, USA
bebis@cse.unr.edu

Abstract. In this paper, image forgery detection method based on the curvelet transform and local binary pattern (LBP) is proposed. First, a color image is converted into the chrominance space. Then, the curvelet transform is applied to the chrominance component to decompose it into several scale and orientation wedges. The LBP normalized histogram is calculated from each of the wedges. The final feature vector is obtained by fusing all the histograms. The proposed method is evaluated on three image forgery datasets and compared with some state of the art methods. Experimental results demonstrate the superiority of the proposed method over the compared methods. The detection accuracy of the proposed method is 93.4% 97.0 % and 94.2% on the CASIA TIDE v1.0, CASIA TIDE v2.0 and Columbia color databases, respectively.

1 Introduction

The advancement of digital imaging technology increases the application of digital images in various aspects of our daily life, for example, newspaper, magazine, TV, commercials, security, insurance, to name a few. In one hand, the availability of low-cost and user friendly image editing tools make our life easier, and on the other hand, it raises a serious authentication issue due to its mishandling by some people. It has never been so easy to manipulate the images to gain illegal advantages or to make false propaganda using forged images [1]. Therefore, correctly detecting the forgeries in the images is a growing interest among the related researchers.

There are mainly two types of image forgeries, which are cloning and splicing. In the cloning forgery, a part of an image is copied and pasted to another part of the same image. In the image splicing, copy and pasting involve two or more images. The purpose of the image forgery is to duplicate or conceal a certain object into an image, or to make false propaganda. Performing of post-processing operations such as blurring, adding noise and JPEG compression or geometric operations such as scaling, shifting and rotation increases the hardness of the detection tasks.

The research on image forgery detection mainly started from early 2000. Many studies are reported on the literature to detect cloning forgery or splicing. Fridrich et al [2] and Huang et al [3] used discrete cosine transform coefficients to create a feature vector from the blocks of an image for block matching based methods of cloning forgery detection. Multi-resolution techniques, such as discrete wavelet

trans-form (DWT) is utilized in several methods [4], [5]. SIFT is another popular method in cloning forgery detection [6]. In image splicing detection, most of the works are evaluated using the Columbia authentic and spliced image dataset (color) [7] and CASIA tampered image detection evaluation dataset [8]. Ng et al used higher-order moments of the image spectrum to detect image splicing [9], while geometry invariants and camera response function are used in [10]. Shi et al proposed statistical features based on 1D and 2D moments, and transition probability features based on Markov chain in DCT domain for image splicing detection [11]. The method achieves 84.86% accuracy on the CASIA v2.0 database. Later, He et al improved the method by combining transition probability features in DCT and DWT domains [12]. For classification, they used support vector machine (SVM) - recursive feature elimination (RFE). Their method obtains 89.76% accuracy on the CASIA v2.0 database. The transition probability features extracted from chrominance channels of an edge-thresholded image was proposed in [13]. The method gets 95.6% accuracy in Cb chrominance channels in the CASIA v2.0 database, though in their experiments, they did not use the full database. The same method achieves 89.23% accuracy in Cb channel for Columbia color database. In another recent method, chroma-like channel is designed for image splicing detection [14], and improves the performance to 93.14% of the method [13] when applied to that chroma-like channel.

In this paper, an image forgery detection method based on a curvelet transform and local binary pattern is proposed. Curvelet transform is applied on chrominance components of a color image and LBP features are extracted from the resultant curvelet wedges. The LBP histograms of all the wedges are concatenated to form the feature vector. The SVM is used as the classifier. The main aim of this paper is to make a decision on image forgery, rather than localizing the forgery in the image.

The rest of this paper is organized as follows. In Section 2, the proposed method is described. Section 3 presents and discusses the experimental results, and section 4 concludes the paper with future work.

2 Proposed Method

Fig. 1 shows a block diagram of the proposed method. First, an RGB color image is converted into YCbCr chrominance space, where Y is the luminance, and Cb and Cr are the chrominance components. The Cb and Cr are the blue difference and the red difference, respectively. The human eyes is more sensitive to the luminance channel than the chrominance channels. As forgery is hard to detect in naked human eyes, chrominance channels are more suitable for forgery detection [13]. Therefore, the concentration is almost on Cb and Cr channels.

In the second step, the curvelet transform is applied to each individual chrominance component. Curvelet transform is a powerful multiscale multi-orientation image decomposition technique. It was developed to solve the problem of curve singularities. As an image analysis tool, it differs from other directional wavelet transforms in the degree of localization in orientation, which varies with scale. It provides a strong directional characterization in which elements are highly anisotropic at fine scales. With these properties, curvelet solve the isotropic and limited directional analysis of classic wavelet transform.

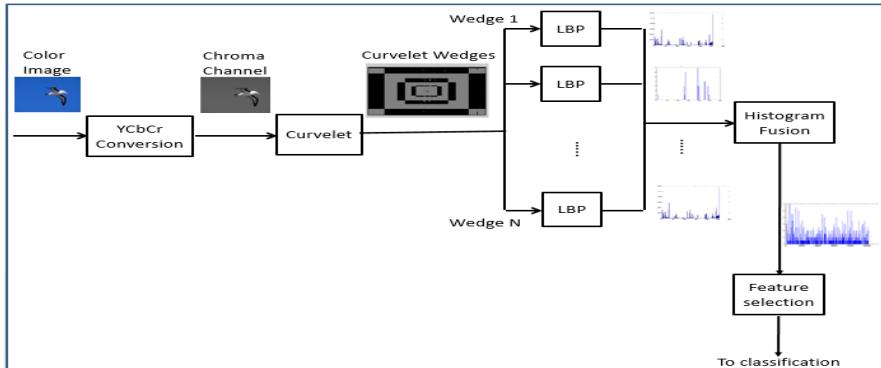


Fig. 1. Block diagram of the proposed curvelet and LBP based forgery detection system

The curvelet transform is actually an improvement for the ridgelet transform, which was proposed in 1999 as anisotropic geometric wavelet transform. The ridgelet transform can represent straight-line singularities perfectly. Unfortunately, global straight-line singularities rarely happen in real applications. The application of ridgelet transform on small partitions of the image to analyze local line or curve singularities, is the idea behind the curvelet transform proposed in 2000. A very efficient second-generation curvelet transform based on frequency partition technique was proposed after that [16]. To extract the curvelet coefficients, the corresponding image component (Y, Cb or Cr) is decomposed to different subbands of different frequencies. Then, each of these subbands is smoothly partitioned into squares of an appropriate scale. Each resulting square partition is renormalized to unit scale. The ridgelet transform is applied on each normalized partition. In this study, a 4 scales curvelet transform is used including the coarsest level. The second coarsest level is set to contain 8 different angles. The two higher frequency levels contain 16 different angles each and as a result we have a total number of 41 different angular wedges. Fig. 2 shows an example of a 3 scale curvelet transform and a total number of 25 curvelet wedges enclosed in the central square (the coarsest level) and different strips in the four Cartesian directions.

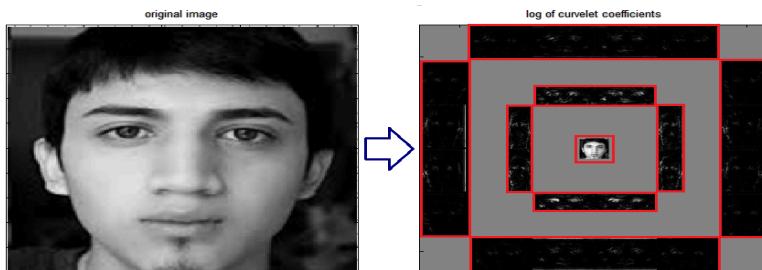


Fig. 2. Curvelet decomposition of an image in 3-scales and a total number of 25 curvelet wedges

The third step is the feature extraction, LBP is applied on each angular wedge of the curvelet to extract the LBP normalized histogram. LBP is a texture descriptor that labels each pixel in the image by thresholding the neighborhood pixels with the center pixel value and considering the result as a binary number as in Fig. 3. This type is called the basic LBP operator. Then the texture can be described by the histogram of these label values.

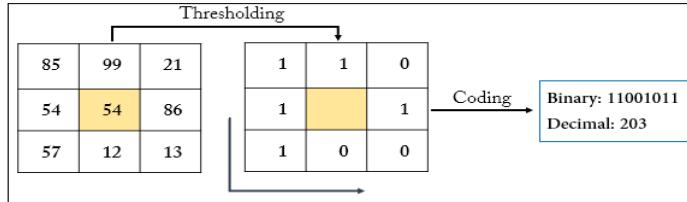


Fig. 3. The basic LBP operator

Another extension to the basic LBP operator is the uniform pattern in which only small subset of the total number of patterns is used to describe any texture. This subset of patterns is named uniform patterns. For any local binary pattern, it can be considered as uniform, if it contains at most two bitwise transitions from 0 to 1 or vice versa. For example, 00000000 (0 transitions), 11001111 (2 transitions) and 01110000 (2 transitions) are uniform, whereas 11001001 (4 transitions) and 01010010 (6 transitions) are not [17]. The length of the uniform LBP histogram of each wedge is 59 and as a result of histogram fusion, the resulting feature vector is of length 2419 (41×59).

While doing forgery, the original texture is distorted, and thereby, the LBP can encode texture differences at different scales and orientations of a curvelet transformed image.

In the fourth step, two different data reduction methods were used for feature selection, which are zero-norm minimization and local learning based (LLB) [18]. Zero-norm minimization ranks the features based on the statistical significance, while LLB removes features that contain redundant information. A cascading of the two methods is used to enhance the performance of the proposed method. In the cascading, first, zero-norm minimization is applied to select 50% of the total number of features (the most discriminative features). Then the LLB is applied on the selected features, and only those features having discriminative weight greater than the threshold of 10^{-10} according to LLB, are nominated. In the final step of the proposed method, the SVM classification with RBF (radial basis function) kernel and 10-fold cross-validation is used to evaluate the performance of the proposed method. Gamma and c parameters of the SVM are automatically set via a grid search process.

3 Experimental Results and Discussion

Three different datasets are used in our experiments, CASIA TIDE v1.0 [8], CASIA TIDE v2.0 [8], and DVMM images dataset of Columbia University [7]. Comparisons with other recent studies in the field of digital images forgery

detection are also provided on these datasets. In the experiments, a randomly selected 50% of the whole dataset samples are used for the feature selection step. The LIBSVM library [19] is then used for the classification. The optimal values for the parameters of kernel function and support vector classification (γ and c), are automatically set by an intensive grid search process using 25% of the whole dataset after reducing the number of features. The performance of the proposed method is given in terms of accuracy averaged over 10 iterations of the SVM. The accuracy is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \times 100. \quad (1)$$

where, True Positive (TP) is the number of forged images, which are classified as forged images, True Negative (TN) is the number of authentic images, which are classified as authentic images, False Positive (FP) is the number of authentic images, which are classified as forged images, and False Negative (FN) is the number of forged images, which are classified as authentic images.

Intensive experiments are carried out to test the performance of the proposed method. On different chrominance component with curvelet transform and LBP.

3.1 Experiments on CASIA TIDE v1.0 Dataset

CASIA TIDE v1.0 dataset contains a total number of 1721 images. 800 image are authentic and the remaining 921 are forged images of which 461 are cloned images and the remaining 460 are spliced. All the images have the size of 384×256 pixels, and they are in JPEG format.

3.1.1 Experimental Results on Cloned Images Subset of CASIA TIDE v1.0

The total number of images in this experiment is 719. The number of forged images are 461, and the rest are authentic images. Table 1 shows the averaged accuracies of the proposed method with and without feature selection in different chrominance channels on cloned images subset of CASIA TIDE v1.0 dataset.

Table 1. Results of the proposed method on cloned images subset of CASIA TIDE v1.0 dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$88.5\% \pm 5.5$	$88.0\% \pm 3.9$
Cr	$90.0\% \pm 3.8$	$91.1\% \pm 3.3$
Y	$74.1\% \pm 6.8$	$79.4\% \pm 7.2$
Gray	$75.6\% \pm 4.9$	$78.5\% \pm 2.7$

3.1.2 Experimental Results on Spliced Images Subset of CASIA TIDE v1.0

The total number of images in this experiment is 1082. The number of forged images are 460, and the rest are authentic images. Table 2 shows the averaged accuracies of the proposed method with and without feature selection in different chrominance channels on spliced images subset of CASIA TIDE v1.0 dataset.

Table 2. Results of the proposed method on spliced images subset of CASIA TIDE v1.0 dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$92.5\% \pm 3.4$	$92.5\% \pm 3.3$
Cr	$92.8\% \pm 2.5$	$94.5\% \pm 1.8$
Y	$66.7\% \pm 5.0$	$67.4\% \pm 5.4$
Gray	$65.2\% \pm 4.8$	$68.7\% \pm 5.3$

3.1.3 Experimental Results on the Whole CASIA TIDE v1.0 Dataset

The total number of images in this experiment is 1721. The number of forged images are 921, and the rest are authentic images. Table 3 shows the averaged accuracies of the proposed method with and without feature selection in different chrominance channels on the whole CASIA TIDE v1.0 dataset.

Table 3. Results of the proposed method on the whole CASIA TIDE v1.0 dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$91.2\% \pm 2.0$	$90.7\% \pm 2.5$
Cr	$93.0\% \pm 1.8$	$93.4\% \pm 1.7$
Y	$67.6\% \pm 4.5$	$69.9\% \pm 3.0$
Gray	$67.8\% \pm 4.1$	$67.6\% \pm 3.3$

The ROC curves in figure 4 illustrate the performance of the proposed method with and without feature selection on the whole CASIA TIDE v1.0, in different channels.

3.2 Experiments on CASIA TIDE v2.0 Dataset

CASIA TIDE v2.0 dataset contains a total number of 12614 images. 7491 image are authentic and the remaining 5123 are forged images of which 3300 are cloned images and the remaining 1823 are spliced. The image sizes varying from 240×160 to 900×600 pixels, and they are in JPEG BMP or TIFF formats.

3.2.1 Experimental Results on Cloned Images Subset of CASIA TIDE v2.0

The total number of images in this experiment is 5006. The number of forged images are 3300, and the rest are authentic images. Table 4 shows the averaged accuracies of the proposed method with and without feature selection in Cb and Cr channels on cloned images subset of CASIA TIDE v2.0 dataset.

Table 4. Results of the proposed method on cloned images subset of CASIA TIDE v2.0 dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$95.2\% \pm 0.7$	$95.3\% \pm 1.2$
Cr	$95.0\% \pm 0.8$	$95.1\% \pm 0.7$

3.2.2 Experimental Results on Spliced Images Subset of CASIA TIDE v2.0

The total number of images in this experiment is 3718. The number of forged images are 1823, and the rest are authentic images. Table 5 shows the averaged accuracies of the proposed method with and without feature selection in Cb and Cr channels on spliced images subset of CASIA TIDE v2.0 dataset.

Table 5. Results of the proposed method on spliced images subset of CASIA TIDE v2.0 dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$93.9\% \pm 1.0$	$94.0\% \pm 1.0$
Cr	$94.5\% \pm 1.0$	$94.6\% \pm 1.2$

3.2.3 Experimental Results on the Whole CASIA TIDE v2.0 Dataset

The total number of images in this experiment is 12614. The number of forged images are 5123, and the rest are authentic images. Table 6 shows the averaged accuracies of the proposed method with and without feature selection in Cb and Cr channels on the whole CASIA TIDE v2.0 dataset.

Table 6. Results of the proposed method on the whole CASIA TIDE v2.0 dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$97.0\% \pm 0.5$	$96.8\% \pm 0.8$
Cr	$96.7\% \pm 0.5$	$96.6\% \pm 0.7$

The ROC curves in figure 5 illustrate the performance of the proposed method with and without feature selection on the whole CASIA TIDE v2.0 dataset, in Cb and Cr channels

3.3 Experiments on DVMM Images Dataset

DVMM images dataset contains a total number of 363 images. 183 images are authentic and the remaining 180 are spliced images. The image sizes range from 757×568 to 1152×768 pixels, and they are in BMP or TIFF formats. Table 7 shows the averaged accuracies of the proposed method with and without feature selection in Cb and Cr channels on DVMM images dataset.

Table 7. Results of the proposed method on the DVMM images dataset

Channels	W/O Feature Selection	With Feature Selection
Cb	$93.9\% \pm 5.0$	$94.2\% \pm 3.1$
Cr	$91.7\% \pm 5.9$	$92.8\% \pm 4.0$

The ROC curves in figure 6 illustrate the performance of the proposed method with and without feature selection on the DVMM images dataset, in Cb and Cr channels.

For the purpose of comparison with other studies' performance, some of the recent researches, which are evaluated on the same datasets are selected. Table 8 shows the best accuracies achieved in each of those researches versus the accuracy achieved by the proposed method on both CASIA v2 and DVMM datasets. The proposed method outperforms all of those methods. The second highest accuracy achieved in [13], is 95.6%, but they did not use the full dataset.

CASIA TIDE v 2.0		DVMM	
Method	Accuracy	Method	Accuracy
[12]	89.8 %	[14]	93.1%
[13]	95.6%	[15]	85.0%
Proposed	97.0 %	Proposed	94.2 %

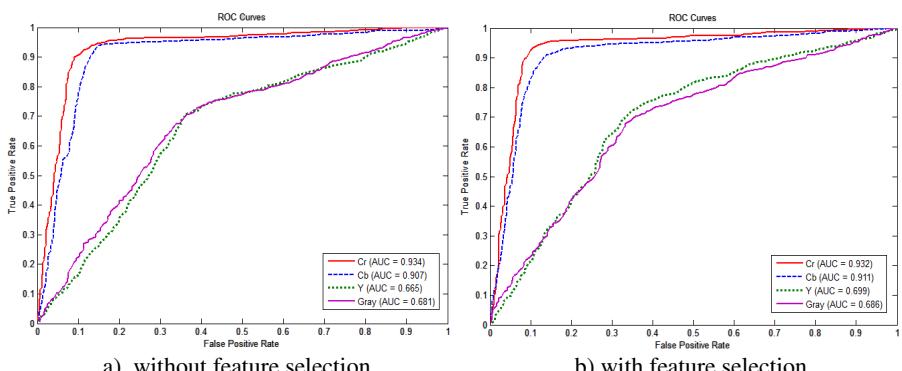


Fig. 4. ROC curves of the proposed method on CASIA TIDE v1.0 dataset

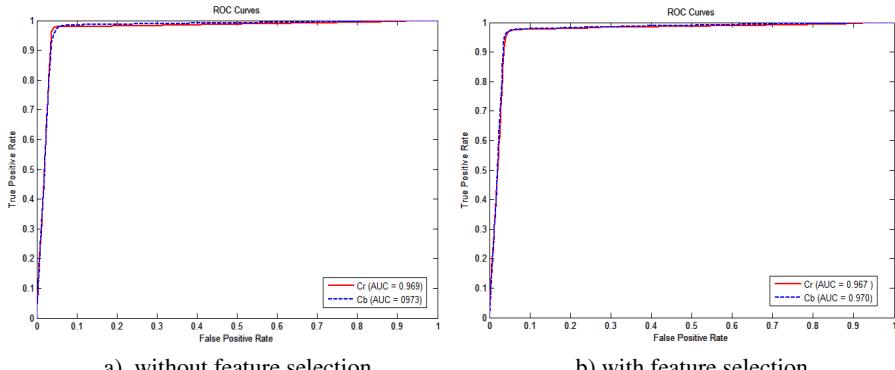


Fig. 5. ROC curves of the proposed method on CASIA TIDE v2.0 dataset

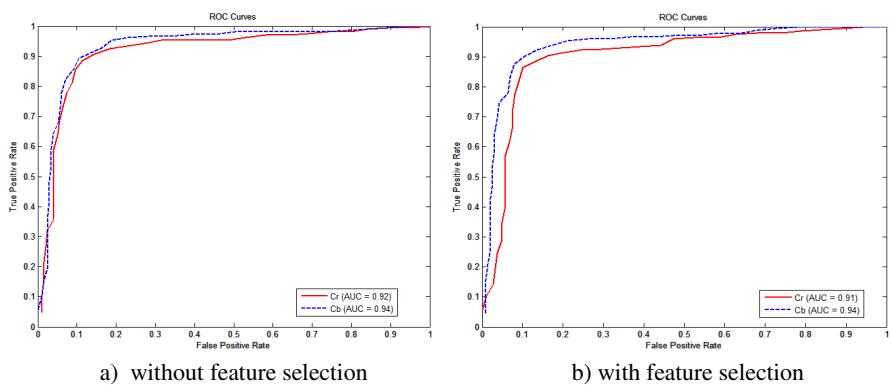


Fig. 6. ROC curves of the proposed method on DVMM images dataset

4 Conclusion

A curvelet and LBP based image forgery detection method is proposed. The chrominance channels of an image are first transformed into multiple curvelet wedges of different scales and orientations. Then, LBP normalized histogram extracted from each wedges is used as features. SVM with RBF kernel is used as the classifier. Optional feature selection techniques are also applied. According to the experiments, the best accuracy is achieved in Cr and Cb chrominance components. The best accuracies of the proposed method are 91.74% on CASIA TIDE v1.0 dataset, 97.0% for CASIA TIDE v2.0 dataset, and 94.2% on DVMM images dataset. These accuracies are significantly higher than those obtained by the other state of the art methods on these datasets. Our future work will be to localize the forgery in the images.

Acknowledgement. This work is supported by the National Plan for Science and Technology, King Saud University, Riyadh, Saudi Arabia under project number 10-INF1140-02.

References

1. Swaminathan, A., Wu, W., Liu, K.J.R.: Digital Image Forensics via Intrinsic Fingerprints. *IEEE Trans. Information Forensics and Security* 3(1), 101–117 (2008)
2. Fridrich, J., Soukal, D., Lukas, J.: Detection of Copy-Move Forgery in Digital Images. In: Proceedings of Digital Forensic Research Workshop (August 2003)
3. Huang, Y., Lu, W., Sun, W., Long, D.: Improved DCT-based detection of copy-move forgery in images. *Forensic Science International* 206(1-3), 178–184 (2011)
4. Li, G., Wu, Q., Tu, D., Sun, S.: A Sorted Neighborhood Approach for Detecting Duplicated Regions in Image Forgeries based on DWT and SVD. In: IEEE International Conference on Multimedia and Expo, ICME 2007, Beijing, pp. 1750–1753 (2007)
5. Muhammad, G., Hussain, M., Bebis, G.: Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digital Investigation* 9(1), 49–57 (2012)
6. Amerini, I., Ballan, L., Caldelli, R., Del Bimbo, A., Serra, G.: A SIFT-based forensic method for copy-move attack detection and transformation recovery. *IEEE Trans. Information Forensics and Security* 6(3), 1099–1110 (2011)
7. Ng, T.T., Chang, S.F.: A dataset of authentic and spliced image blocks. Technical Report 203-2004, Columbia University (2004), <http://www.ee.columbia.edu/ln/dvmm/downloads/>
8. Dong, J., Wang, W.: CASIA tampered image detection evaluation (TIDE) database, v1.0 and v2.0 (2011), <http://forensics.idealtest.org/>
9. Ng, T.T., Chang, S.F., Sun, Q.: Blind detection of photomontage using higher order statistics. In: IEEE Intl. Symposium Circuits and Systems, ISCAS, pp. 688–691 (2004)
10. Hsu, Y.F., Chang, S.F.: Detecting image splicing using geometry invariants and camera characteristics consistency. In: IEEE ICME 2006, pp. 549–552 (2006)
11. Shi, Y.Q., Chen, C., Chen, W.: A natural image model approach to splicing detection. In: ACM Multimedia & Security, MM&S 2007, pp. 51–62 (2007)
12. He, Z., Lu, W., Sun, W., Huang, J.: Digital image splicing detection based on Markov features in DCT and DWT domain. *Pattern Recognition* (2012), <http://dx.doi.org/10.1016/j.patcog.2012.05.014>
13. Wang, W., Dong, J., Tan, T.: Image tampering detection based on stationary distribution of Markov chain. In: IEEE Intl. Conference on Image Processing, ICIP 2010, pp. 2101–2104 (2010)
14. Zhao, X., Li, S., Wang, S., Li, J., Yang, K.: Optimal chroma-like channel design for passive image splicing detection. *EURASIP Journal on Advances in Signal Processing* (2012), doi:10.1186/1687-6180-2012-240
15. Zhao, X., Li, J., Li, S., Wang, S.: Detecting digital image splicing in chroma spaces. In: Kim, H.-J., Shi, Y.Q., Barni, M. (eds.) IWDW 2010. LNCS, vol. 6526, pp. 12–22. Springer, Heidelberg (2011)
16. Starck, J.-L., Candès, E.J., Donoho, D.L.: The curvelet transform for image denoising. *IEEE Transactions on Image Processing* 11, 670–684 (2002)
17. Ahonen, T., Hadid, A., Pietikainen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(12), 2037–2041 (2006)
18. Sun, Y., Todorovic, S., Goodison, S.: Local Learning Based Feature Selection for High Dimensional Data Analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence* 32(9), 1610–1626 (2010)
19. Chang, C.C., Lin, C.J.: LIBSVM - a library for support vector machine (2010), <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Camera Distance from Face Images

Arturo Flores, Eric Christiansen, David Kriegman, and Serge Belongie

University of California, San Diego
9500 Gilman Drive, La Jolla, CA, USA
`{aflores,echristiansen,kriegman,sjb}@cs.ucsd.edu`

Abstract. We present a method for estimating the distance between a camera and a human head in 2D images from a calibrated camera. Leading head pose estimation algorithms focus mainly on head orientation (yaw, pitch, and roll) and translations perpendicular to the camera principal axis. Our contribution is a system that can estimate head pose under large translations parallel to the camera’s principal axis. Our method uses a set of exemplar 3D human heads to estimate the distance between a camera and a previously unseen head. The distance is estimated by solving for the camera pose using Effective Perspective n -Point (EPnP). We present promising experimental results using the Texas 3D Face Recognition Database.

1 Introduction

When photographing a human head, the subject’s appearance can vary dramatically depending on the camera’s distance from the subject. This variation is caused by perspective distortion, and for 3D objects cannot be undone by simply adjusting focal length; see Figure 1 for an illustration using a synthetic head¹.

This distortion presents a problem for automatic cross-condition face recognition, e.g. webcam-based recognition from social media images. Even humans find such recognition difficult [1,2]. It is also a source of information, allowing camera pose estimation in cases where the subject is known [3]. This information could potentially be used to undistort the images and improve recognition results.

In this paper, we show camera distance estimation from 2D images is possible even when the subject is previously unseen. Our technique replaces the known-subject assumption with knowledge of the general distribution of fiducials across people. This distribution turns out to be sufficiently tight to allow surprisingly accurate distance estimation using only a small training set.

The paper is organized as follows. Section 2 covers related work from psychology and computer vision. Section 3 explains our method. In Section 4, we validate our method on the Texas 3D Face Recognition Database. Section 5 is the discussion and conclusion.

¹ Generated using FaceGen (<http://www.facegen.com>).

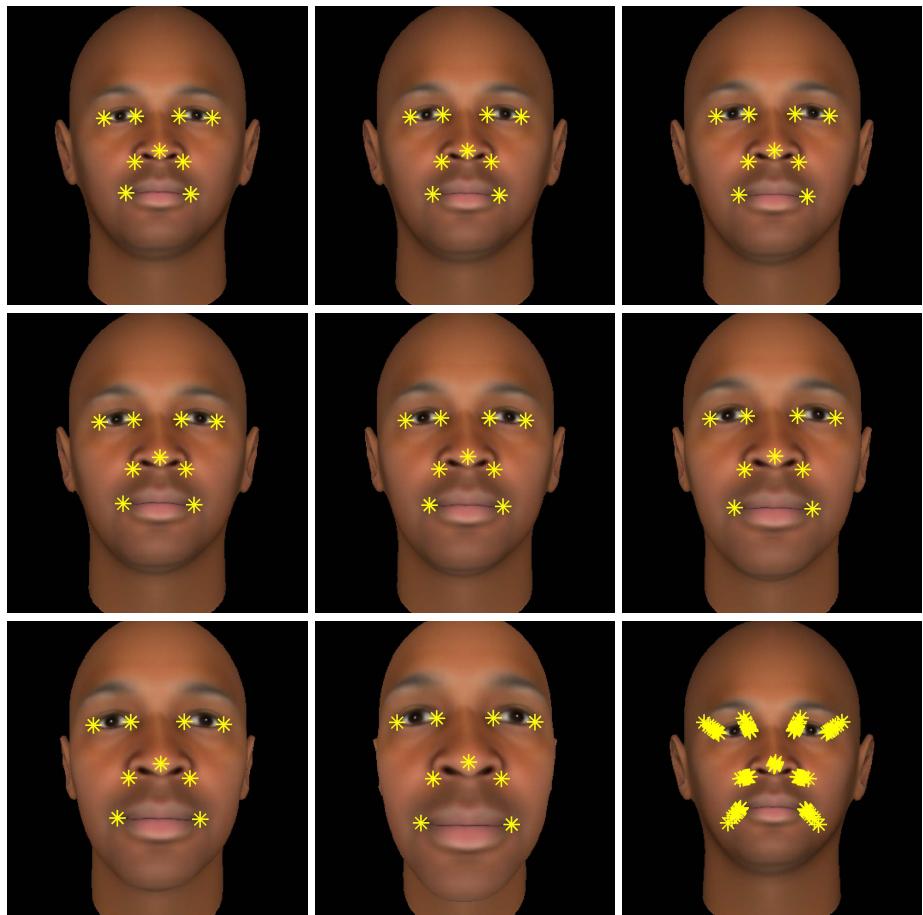


Fig. 1. A synthetic head viewed from different camera distances, illustrating projective distortion. Camera distance decreases in the first eight images, indexed in row-major order. Here, the focal length (zoom) is adjusted to keep the figure at a constant size. Fiducials are shown as red dots. In the first image, the camera is far away, resulting in near orthographic projection. In the eighth image (bottom row, middle column) the camera is very close to the human head. The last image is the same as the first, but with fiducial markers from all images. This illustrates the migration of fiducials as a function of camera distance and focal length.

2 Related Works

There is previous work on head pose estimation from 2D images; see [4] for a recent survey. However, most methods attempt to recover a subset of the yaw,

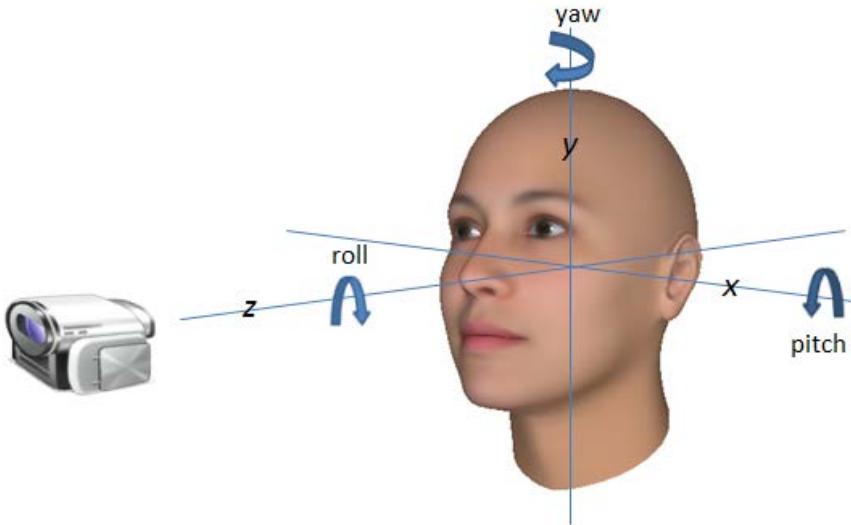


Fig. 2. An illustration of the six degrees of freedom governing head pose relative to a camera. Prior work has focused on estimation of yaw, pitch, and roll [4]. In this paper, we assume these parameters are known and estimate camera distance from the subject, shown here as distance along the z axis.

pitch, and roll of the head with respect to the camera. To our knowledge, these methods do not attempt to estimate the distance between the camera and head; Figure 2 illustrates the difference. In this section we discuss works most similar to our own.

In [1,2], Liu et al. study the effect of face recognition by humans when viewing faces at different perspective convergence angles (effectively focal length or field of view). The study involved a training phase in which face images were displayed to a human test subject. In a later recognition phase the subject was shown images of faces and asked to determine whether each face had been previously displayed. The field of view was changed to see if this had an effect on recognition. Their results show that even humans have a hard time recognizing a face when viewed under different levels of perspective distortion. This is a motivating factor since if humans have trouble with this task, a computer vision algorithm will likely also have the same troubles. Predicting the distance between camera and face is a first step in mitigating the effects of perspective distortion.

A similar psychology based study is presented in [5,6]. Here, Perona et al. investigate the effects of perspective distortion as visual cue for social judgement of faces. Human subjects were asked to judge an image of a face in terms of

trustworthiness, attractiveness, and competence. Their results show that for social judgements, pictures taken up close are generally rated lower, while pictures taken far away have higher ratings.

In automatic camera calibration, camera parameters are recovered using prior information about the imaged scene. In [7], Deutscher et al. recover camera parameters under the assumption the scene satisfies a Manhattan world criterion. This is similar to our technique, which assumes human fiducial locations are approximately distributed according to an estimated distribution. In [8], Lv et al. use a human subject for calibration, but unlike this work requires multiple frames of video. In [9], Krahmstoever and Mendonca use a full human body for calibration from a single image, where this work uses just the head.

In [3], Ohayon and Rivlin present head tracking as a camera pose estimation problem. Prior to head tracking, 3D points are acquired from the head. During tracking, correspondences between the 3D points and their imaged 2D points are used to estimate head pose by solving the inverse problem, namely camera pose. They use the Perspective n -Point (PnP) formulation to solve for the camera extrinsic parameters (rotation R and translation T). The PnP method is also known as the Location Determination Problem and was first coined in [10]. In effect, the human head is used as a calibration rig. In this paper, the authors show that head pose can be accurately estimated and tracked under varying yaw, pitch, and roll and translations about x and y axes. However, they assume knowledge of the ground-truth fiducial locations and do not address dramatic changes in camera distance.

3 Camera Pose Estimation from Face Images Using EPnP

Our work is based on [3], but we focus primarily on translation along the z -axis, which affects the level of induced perspective distortion. We present a method for estimating the pose of a previously unseen human head using a dataset of exemplar human heads.

Efficient Perspective n -Point (EPnP): The method uses EPnP, a fast, non-iterative, solution to the PnP problem [11]. We use code provided by the authors². As stated earlier, the PnP problem is to estimate the pose of a calibrated camera from n 3D-to-2D correspondences. In particular, EPnP enables us to estimate camera pose based on a set of 2D fiducial locations and their corresponding 3D locations.

Exemplar 3D heads: Note the geometric configuration of fiducial features varies from face to face, but in general fiducial locations tend to form clusters, as illustrated in Figure 3. This means the fiducial locations of a new person are

² <http://cvlab.epfl.ch/software/EPnP>

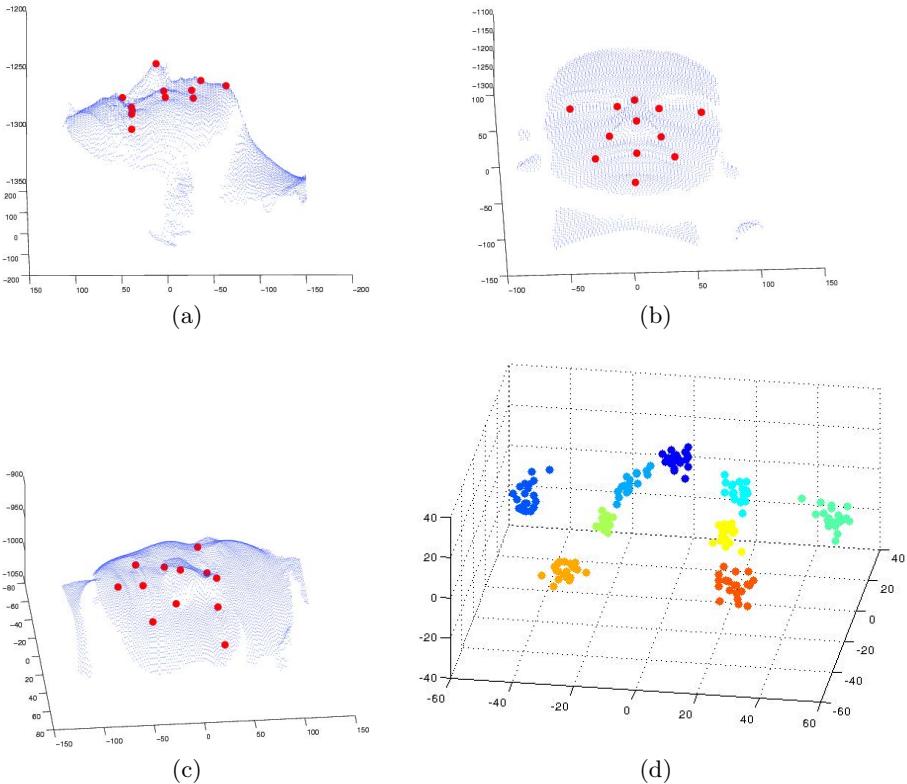


Fig. 3. (a-c) show point clouds of three different faces taken from the GavaBDB [12] dataset. Manually clicked fiducial locations are marked with red dots. (d) shows fiducials for 20 different faces, aligned by subtracting their respective means. Note the fiducials form tight clusters, meaning fiducials do not change dramatically across individuals. This phenomenon enables robust depth estimation even for previously unseen images.

likely to be similar to those in an exemplar set. We take advantage of this by using a set of exemplar 3D heads to estimate the camera pose of a novel head.

The method: The method is based on simple averaging, leveraging the observation from the previous paragraph. Suppose we get an image I of a previously unobserved head. For each exemplar 3D head E , the camera pose is estimated via EPnP under the assumption the fiducials of I match the fiducials of E . This assumption is incorrect, but as mentioned in the last paragraph, it is not far off. The estimated camera distance for I is just the average of the camera distances across all the exemplars.

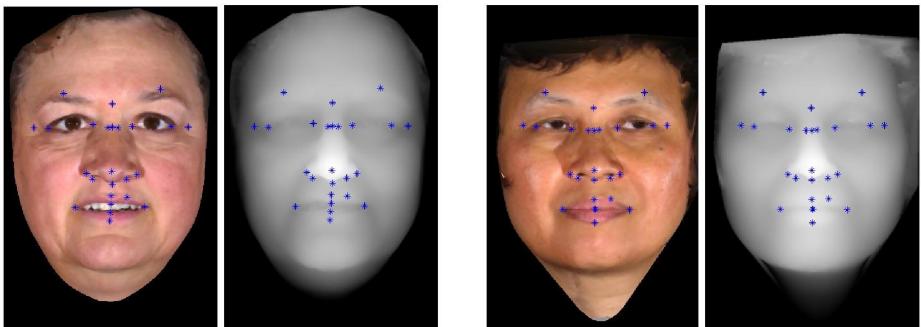


Fig. 4. Example images from the Texas 3DFRD, a database of 3D scans of human faces. These images show the 25 fiducials labeled for each face, as well as example depth maps.

4 Experimental Setup

In this section, we show the surprising effectiveness of our method for estimating camera distance in images of previously unseen subjects.

Our experiments are run against the Texas 3D Face Recognition Database (Texas 3DFRD) [13]; example images are shown in Figure 4. This database is a collection of 1149 pairs of frontal face color and depth images of 105 adult human subjects. Each face also has 25 manually labeled anthropometric facial fiducial points. Color and depth images were captured simultaneously and are perfectly coregistered. This provides ground truth 3D locations of the fiducial locations.

Our first experiment consists of the following:

- Project fiducials for a test subject onto an image plane.
- Use 3D fiducial locations from a set of reference individuals (not including the test individual) as exemplars.
- Estimate the camera distance for the test subject using the method described in Section 3.
- Repeat while simulating a dolly-zoom (or Hitchcock zoom) camera movement, in which the camera moves away as focal length is increased to keep the figure size constant.

We perform this experiment for the frontal and 3/4 profile view of the face. For the frontal case, we assume all fiducials are visible to the camera. For the 3/4 we assume only a subset of the fiducials are visible. We also assume we have a calibrated camera and all intrinsic camera parameters are known. Simulated camera distances range from approximately 10cm to 3m. Note if we did not

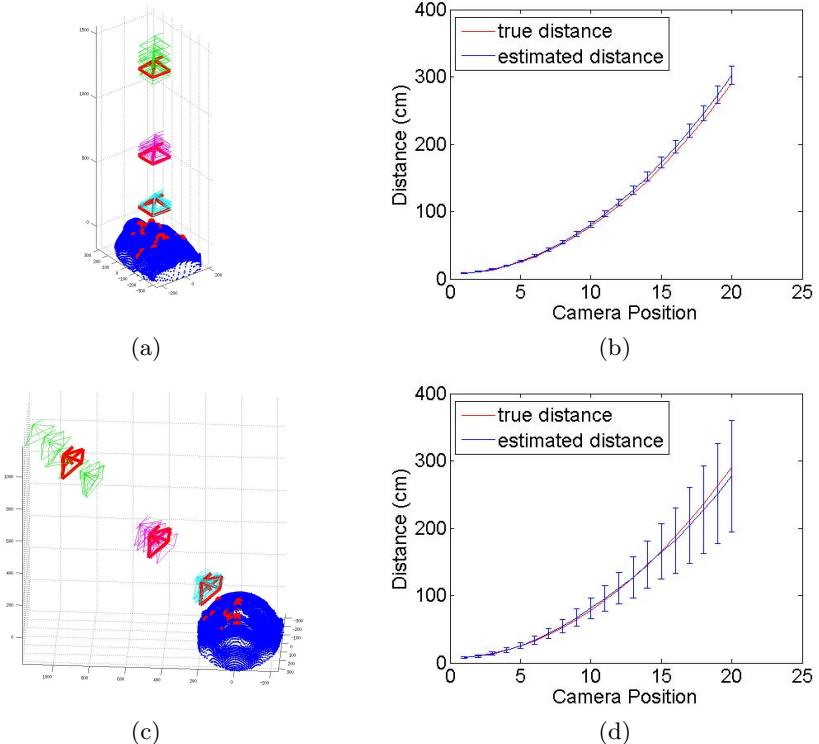


Fig. 5. Results of camera pose and distance estimation for the frontal and 3/4 profile views. (a) and (c) show clusters of estimated camera poses for three distances for the frontal and 3/4 profile views. The clusters of poses for each of the three distances are marked using cyan, magenta, and green camera frames, with the ground truth pose marked in red. (b) and (d) show true and estimated camera distances as a function of camera position, where distance is measured from the camera center to the tip of the nose. We test 20 camera positions. The estimated distances are shown with error bars to represent the variation across test subjects. Note the final distance estimate closely follows the true camera distance, though the error bars get larger as the distance increases. This is especially true for the 3/4 profile view, presumably due to a fewer number of visible fiducials.

adjust the focal length, the distance from the camera to the face could be trivially estimated by the size of the imaged face. For this reason, the focal length is adjusted to keep the outermost fiducials at a near constant distance, which also keeps the imaged face silhouette at a constant size. Results are shown in Figure 5. Our method nearly perfectly recovers camera distance.

Figure 6 shows the same experiment where the number of fiducials is varied. For this experiment, we manually selected the fiducial subsets to be evenly

distributed about the face. This figure shows the distance can be reliably estimated with as few as five fiducials. In the case of five fiducials, the fiducials used were outer corners of eyes and mouth, and center of top lip. When using only four fiducials, the center of top lip was removed, resulting in 4 nearly co-planar points. At this point, the distance estimate becomes unreliable.

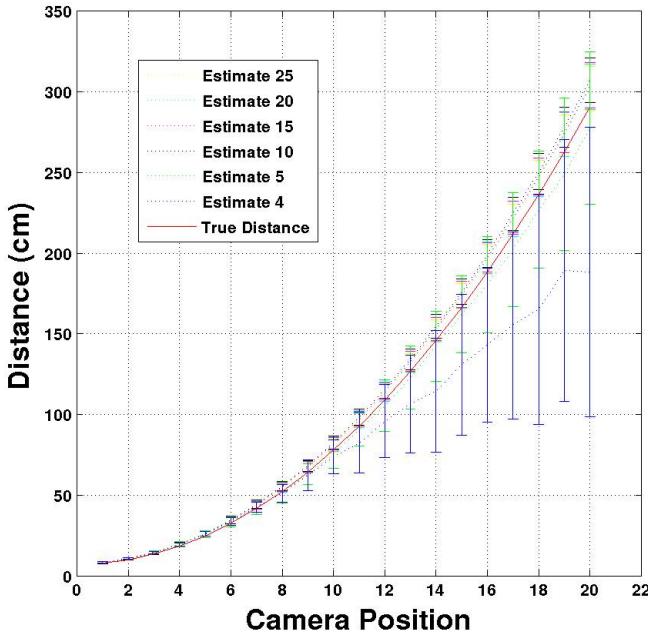


Fig. 6. Results of camera distance estimation for the frontal view for varying numbers of fiducials. As in Figure 5, error bars represent the variation in estimated distance across test subjects. This shows distance can be reliably estimated with as few as five fiducials.

For qualitative analysis, we use a heuristic to select a closest exemplar for each subject and camera distance. Let p_i be the 2D imaged fiducial using the true camera position. Let p'_{ij} be the imaged fiducial using the estimated camera position from the j -th exemplar. We use $\operatorname{argmin}_j \sum_i \|p_i - p'_{ij}\|$ as a heuristic to select the best exemplar. See Figure 7 for some illustrative examples. These examples show fiducial configuration is more than just a means to undistort images; it is a source of biometric information in its own right.

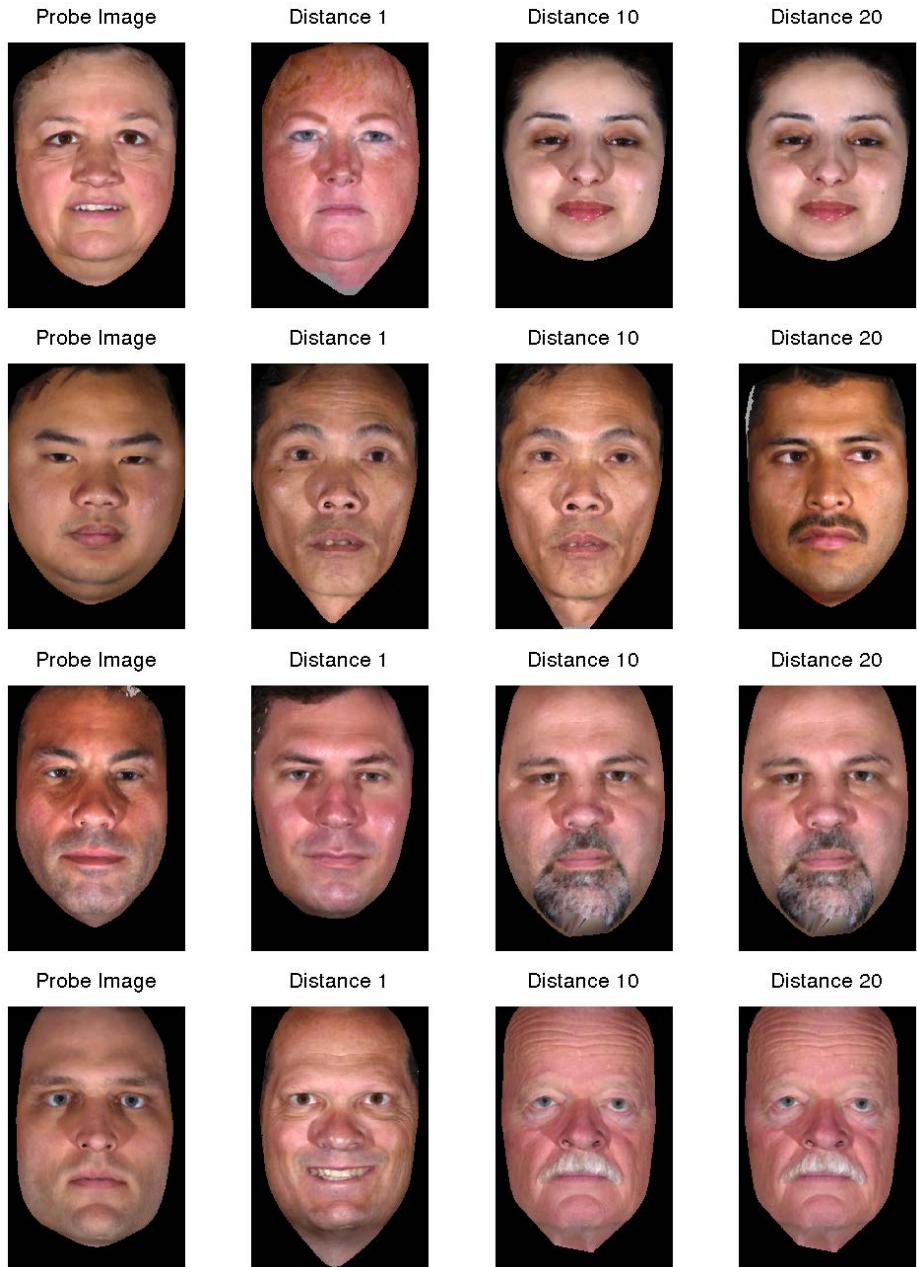


Fig. 7. The closest exemplar as a function of camera distance for four subjects (probes). The probes are shown in the first column. The best exemplars for 10cm, 75cm, and 300cm are shown in the second, third, and fourth columns, respectively. Note the probes tend to match exemplars with similar attributes, e.g. race and gender.

5 Conclusion and Future Work

We presented a method for estimating camera distance to a previously unseen face. The method uses correspondences of 2D image fiducials to 3D fiducial locations on exemplar heads. Though the method is simple, it is accurate for distances ranging from 10cm to 300cm for the frontal and 3/4 profile views. This estimate could be used to mitigate the effects of perspective distortion for face recognition. The method could also be used as a direct source of biometric information, by leveraging the attributes of the closest matching exemplars.

Future work may replicate these results on real images. We may also investigate automatic fiducial localization [14], or naturally occurring features such as in [3]. Finally, we may estimate more general poses, such as the full extrinsic camera parameters.

This work was supported by ONR MURI Grant #N00014-08-1-0638.

References

1. Liu, C.H., Chaudhuri, A.: Face recognition with perspective transformation. *Vision Research* 43, 2393–2402 (2003)
2. Liu, C.H., Ward, J.: Face recognition in pictures is affected by perspective transformation but not by the centre of projection. *Perception* 35, 1637 (2006)
3. Ohayon, S., Rivlin, E.: Robust 3d head tracking using camera pose estimation. In: 18th International Conference on Pattern Recognition, ICPR 2006, vol. 1, pp. 1063–1066. IEEE (2006)
4. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 607–626 (2009)
5. Perona, P.: A new perspective on portraiture. *Journal of Vision* 7, 992–992 (2007)
6. Bryan, R., Perona, P., Adolphs, R.: Perspective distortion from interpersonal distance is an implicit visual cue for social judgments of faces. *PloS One* 7, e45301 (2012)
7. Deutscher, J., Isard, M., MacCormick, J.: Automatic camera calibration from a single manhattan image. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*, Part IV. LNCS, vol. 2353, pp. 175–188. Springer, Heidelberg (2002)
8. Lv, F., Zhao, T., Nevatia, R.: Camera calibration from video of a walking human. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 1513–1518 (2006)
9. Krahnstoever, N., Mendonca, P.R.: Bayesian autocalibration for surveillance. In: Tenth IEEE International Conference on Computer Vision, ICCV 2005, vol. 2, pp. 1858–1865. IEEE (2005)
10. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395 (1981)
11. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision* 81, 155–166 (2009)
12. Moreno, A.B., Sanchez, A.: Gavabdb: a 3d face database. In: Proc. 2nd COST 275 Workshop on Biometrics on the Internet, Vigo, Spain, pp. 75–80 (2004)
13. Gupta, S., Castleman, K.R., Markey, M.K., Bovik, A.C.: Texas 3d face recognition database. In: 2010 IEEE Southwest Symposium on Image Analysis & Interpretation (SSIAI), pp. 97–100. IEEE (2010)
14. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 545–552. IEEE (2011)

Towards Robust Gait Recognition

Tenika P. Whytock*, Alexander Belyaev, and Neil M. Robertson**

Institute of Sensors, Signals and Systems, School of Engineering & Physical Sciences
Heriot-Watt University, Edinburgh, Scotland, UK

Abstract. Covariate factors, such as persons carrying a bag and wearing a jacket, continue to cause significant misclassification in gait recognition. A novel and efficient approach learns a “typical” Gait Energy Image representation free from covariate factors which aids their mitigation in test and training data. Combating the influence of covariate factors yields a significant improvement of 11% over existing state of the art performance for sequences capturing persons wearing a jacket.

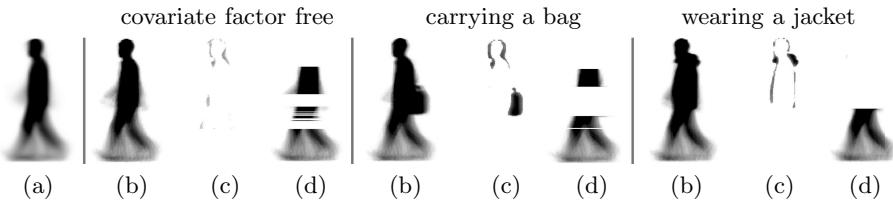


Fig. 1. Covariate factor removal: (a) “typical” Gait Energy Image, (b) test Gait Energy Images, (c) detected covariate factor influenced areas, (d) test Gait Energy Images with areas in (c) removed - for covariate factor free, carrying a bag and wearing a jacket

1 Introduction

Gait is defined as the walking manner of a person, therefore gait recognition permits person identification via gait. Biometrics, either physiological or behavioural, enable human identification. Physiological biometrics such as finger-print and iris are more established, however behavioural biometrics such as gait are trending in computer vision research. Gait is advantageous over alternative biometrics as it can be obtained without consent or cooperation, unobtrusively, at low resolution and at a distance. Early research in medicine [1] and psychophysics [2] confirm gait to be unique and demonstrate that a person can recognise their friend via gait. The fundamental walking pattern is similar across healthy persons, sans pathological conditions, however subtle variations in magnitude and timing support person discrimination. Further still, gait can

* Supported by BMVA.

** Supported by FP 7 LOCOBOT (ref 260101) www.locobot.eu

be difficult to disguise, fake and forget. Applications include, but are not limited to, surveillance and access control. The general gait recognition flow opens with gait representation, followed by feature extraction and ultimately classification.

Gait recognition research tends to investigate robustness during viewpoint variation or covariate factors e.g. persons wearing a jacket or carrying a bag - Fig. 1b. This paper focuses on covariate factors, specifically sequences capturing persons walking from a side view given the significant quantity of limb motion; this fact is demonstrated in daily life such as road signs and films where viewed actions contain the greatest quantity of limb motion for comprehension [3].

Motivation: Covariate Factors

Covariate factors cause misclassification and can be grouped into those affecting gait and its appearance. Gait itself is affected by shoe type: flip-flops and heels, carrying condition: e.g. rucksacks - a by-product of which is leaning to compensate for a shifted centre of gravity and clothing: e.g. jackets, skirts. Gait appearance is altered by factors including viewpoint, occlusions and time. Pixel-wise, covariate factors affect gait through: (1) pixel addition - e.g. clothing or carrying condition adds bulk uniformly or in specific locations, (2) pixel occlusion - e.g. a rucksack occludes trailing arm motion and (3) pixel shifting - e.g. leaning due to carrying a bag.

Table 1. Existing CASIA B dataset performance results for normal (nm), carrying condition (bg) and clothing (cl) sequences; results from the proposed approach are presented at the bottom and state of the art results are highlighted

Method	nm (%)	bg (%)	cl (%)
M_G [4]	100.0	91.0	80.6
Optical flow fusion [5]	97.5	83.6	48.8
GENI [6]	100.0	78.3	44.0
AEI [7]	98.4	91.9	72.2
P_{RW} GEI [8]	98.4	93.1	44.4
SEI + GSP [9]	99.0	64.0	72.0
CGI [10]	88.1	60.2	50.8
SGEI + GEI [11]	99	82.5	86.4
GEI-based [12]	99.2	80.6	75.8
Proposed “typical” GEI_1	98.4	77.4	93.2
Proposed “typical” GEI_2	99.6	81.1	88.3
Proposed “typical” GEI_3	99.6	89.9	71.4

The CASIA B dataset [13, 14] is utilised for evaluation due to its size and utilisation frequency; carrying condition (carrying a bag), clothing (wearing a jacket) and viewpoint covariate factors are present in the dataset, however this

paper considers the former two. Existing CASIA B dataset gait recognition performance is seen in Table 1 for normal: no covariate factors (nm), carrying condition (bg) and clothing (cl) sequences. Firstly, notice that normal sequence performance achieves in excess of 97%, while carrying condition and clothing sequences achieve approximately 80% and 70% respectively; secondly performance across sequence type varies significantly. Overall, clothing sequences are more problematic due to equally distributed additional pixels about the torso causing a broader appearance than normal; carrying condition is less problematic due to non uniformly distributed pixels associated with varying bag shapes. In either case, dynamic limb motion becomes occluded causing further misclassification. Such covariate factors are common in the real-world and overcoming their influence remains an open problem.

Related Work

Gait recognition is divided into model-based and model-free approaches. Model-based approaches [15–17] either model or track body segments via anthropometric data [18, 19] to create gait signatures. Despite the benefit of robustness, these approaches have high computational cost and image quality sensitivity. Model-free methods are commonly silhouette-based and receive a significant amount of attention. Low computational cost exists alongside low sensitivity to image quality; this comes at the cost of reduced robustness.

Boosting robustness to covariate factors with model-free approaches prompts two solutions: either construct a new gait representation or modify an existing representation in order to mitigate the influence of covariate factors. One of the most widely utilised model-free representations is the Gait Energy Image (GEI) [20]; whilst disregarding motion and time related information, the GEI condenses a silhouette sequence to a single compact 2D image - note silhouettes ignore texture. Static and dynamic GEI features correspond to high and low intensity pixel values respectively which are indicative of body motion frequency. Research tends to favour dynamic features for their discriminative ability and mitigation of covariate factors (given their static appearance) [21]. The compact nature is advantageous in terms of reducing memory and computational costs, whilst utilising space- and time-normalisation for natural noise mitigation.

The space- and time-normalisation aspect of the GEI has spawned a number of offshoot representations including: M_G [4], Gait Entropy Image (GEnI) [22], Active Energy Image (AEI) [7], Poisson Random Walk GEI (P_{RW} GEI) [8], Chrono-Gait Image (CGI) [10], Shifted Energy Image (SEI) [9] and Structural Gait Energy Image (SGEI) [11]. The M_G mask, GEI derived, retains leg based values whilst segmenting dynamic features present in the head and torso; this is beneficial for carrying condition sequences. The GEnI computes uncertainty at each gait image to highlight dynamic limb motion whilst suppressing static features. The AEI computes the difference between adjacent gait images. The P_{RW} GEI computes the Poisson Random Walk distance function approximation

and segments body extremities via thresholding; this is beneficial for carrying condition sequences. The CGI exploits the lack of temporal information through contour extraction in each gait image and encodes each using multichannel mapping. The SEI segments the body into the head, torso and legs where each is horizontally aligned; this is particularly suited to carrying condition sequences. The SGEI extracts only the head and feet to remove the influence of traditionally located covariate factors. Where [12] is GEI based, our approach differs as we consider the GEI as a complete region of interest as opposed to part based.

Contribution

The contribution of this paper is a simple and efficient approach to mitigating the influence of covariate factors during gait recognition. By learning the appearance of a “typical” GEI containing no covariate factors, it is possible to detect covariate factor influenced areas in test GEIs. The proposed approach yields a significant improvement of 11% over existing state of the art performance when persons are captured wearing jackets - highlighted in Table 1.

2 Detection and Removal of Covariate Factor Areas

While a human observer easily segments covariate factor influenced areas in a GEI, this ability requires translation for a computer to comprehend. A “typical” GEI is constructed to understand how the body is posed and distributed in terms of static and dynamic features when no covariate factors are present; comparison to test GEIs permits the detection of pixel locations and intensities linked to covariate factor influenced areas by means of a threshold. This however prompts the question of who looks like the computed “typical” person GEI?

The detection of covariate factor influenced areas is demonstrated in Fig. 2 for a carrying condition GEI: covariate factor influenced areas are highlighted in

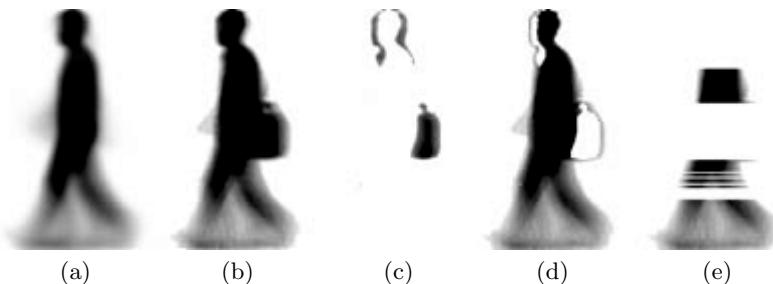


Fig. 2. Detection and removal demonstration of covariate factor influenced areas for a carrying condition GEI: (a) “typical” GEI, (b) test carrying condition GEI, (c) computed covariate factor influenced areas, (d) removal of covariate factor influenced areas and (d) complete row removal of covariate factor influenced areas

Fig. 2c and correspond to the rucksack and leaning - a by-product due to compensation for a shifted centre of gravity. These areas cannot simply be removed as seen in Fig. 2d as visible ringing effects occur in surrounding areas; this residual ringing subsequently contributes to an increased Euclidean distance during Nearest Neighbour classification. Note covariate factors can contain a degree of motion across the gait sequence. To ensure removal of covariate factor influenced areas and adequate removal of potential ringing effects, complete rows of covariate factor influenced areas are removed, demonstrated in Fig. 2e; in this manner the GEI has increased visual similarity to the “typical” and therefore training GEIs when comparing similar areas - note negligible ringing within closely neighbouring rows of the GEI, Fig. 2e, appearing similar to occluded dynamic trailing arm motion in this example.

2.1 Detecting Covariate Factor Influenced Areas

Three definitions of a “typical” GEI are implemented and range in strictness; note suppression of natural intra-class variance is a risk. Each begins with

$$\bar{X} = \frac{1}{c.s} \sum_{i=1}^{i=c.s} X_i, \quad (1)$$

where \bar{X} and X are the covariate factor free mean training and training GEIs respectively, c is the total subject class number and s is the total number of training sequences. Leniency is enabled by including

$$\sigma = \sqrt{\frac{1}{(c.s)} \sum_{i=1}^{i=c.s} (Y_i - \bar{X})^2}, \quad (2)$$

where σ is the standard deviation and Y are test GEIs. Now the three “typical” GEIs are defined

$$tGEI_1 = \bar{X} \quad tGEI_2 = \bar{X} \pm \sigma \quad tGEI_3 = \bar{X} \pm 2\sigma, \quad (3)$$

where $tGEI_{1,2,3}$ are “typical” GEIs where σ increases leniency and are considerably smoother in comparison to test and training GEIs, demonstrated in Fig. 2a vs. Fig. 2b.

To detect covariate factor influenced areas using $tGEI_1$,

$$CV = |Y - tGEI_1|, \quad (4)$$

where CV highlights covariate factor influenced areas. Given the leniency applied to “typical” $tGEI_{2,3}$, detection instead requires

$$V_{added} = Y - (tGEI_2 - 2\sigma) \quad V_{added} = Y - (tGEI_3 - 3\sigma), \quad (5)$$

$$V_{missing} = (tGEI_2 + 2\sigma) - Y \quad V_{missing} = (tGEI_3 + 3\sigma) - Y, \quad (6)$$

$$CV = \sum V_{missing} + V_{added}, \quad (7)$$

where $V_{added,missing}$ reject negative values and correspond to added and missing pixel locations and intensities where CV is their summation highlighting covariate factor influenced areas.

2.2 Covariate Factor Mask Construction and Application

Covariate factor influenced areas are finally chosen based on a threshold T_h ($\{T_h = 0 \text{ to } 1 \text{ in steps of } 0.1\}$) of CV . Note despite learning a covariate factor free “typical” GEI, covariate factor influenced areas are detected in normal test sequences - Fig. 1c-d. Mask M is initially populated with ones where rows are converted to zeros should $CV > T_h$ be satisfied. Finally, M requires dynamic leg related areas to be retained despite classification as covariate factor influenced areas (T_h dependent); areas are located in a bottom-up search where rows in M are converted to ones until a high intensity pixel value ($GEI(x, y) = 1$) is found.

Considering each test GEI in turn, M is applied to both test and training GEIs to mitigate the effects of covariate factor influenced areas; this approach ensures dimensionality reduction and classification, performed next, is applied to areas deemed covariate factor free.

3 Experimental Procedure

Dataset. The CASIA B dataset provides validation: 8 sequences per 124 subjects; GEIs provided [14]. Training data contains 4 normal (covariate factor free) sequences; test data contains 2 sequences each for normal, carrying condition and clothing. Carrying condition sequences include rucksacks, satchels and handbags ranging in location carried; clothing sequences include outdoor jackets.

Dimensionality Reduction and Classification. The GEI serves as both representation and feature vector; GEIs converted to row vector form become $57,600D$ thus calling for dimensionality reduction. Typical in gait recognition, two classical linear dimensionality reduction techniques are applied - Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [23] which reduce the overfitting problem and seek the best data representation and class separability respectively. PCA linearly maps feature vectors to lower dimensionality space and maximise the data variance; $(c-1)D$ accounts for approximately 97% of the variance. LDA subsequently maximises class separability by considering the subject labels. k-Nearest Neighbour classification $\{k = 1\}$ utilises the Euclidean distance yielding Rank 1, i.e. the best match, performance.

4 Results and Discussion

Performance as a function of T_h is seen in Fig. 3 for normal, carrying condition and clothing test sequences with three definitions of the “typical” GEI; the state of the art is highlighted for comparison.

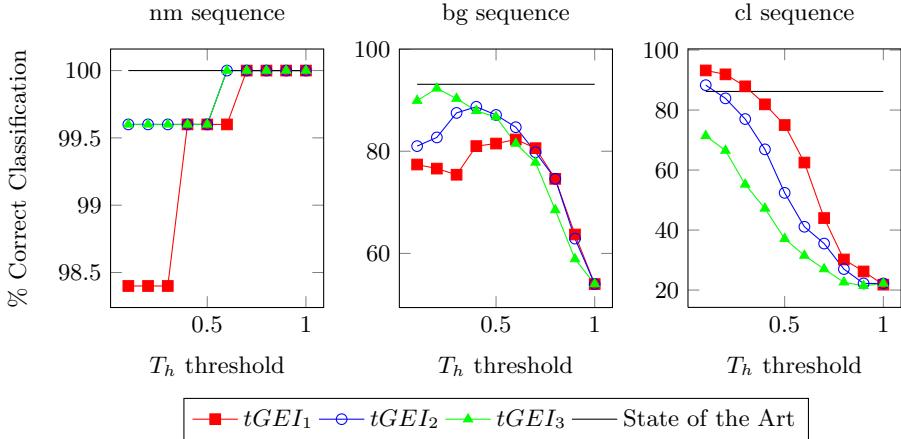


Fig. 3. CASIA B dataset performance as a function of threshold T_h for: normal (nm - left), carrying condition (bg - middle) and clothing (cl - right) test sequences and three definitions of “typical” GEIs $tGEI_{1,2,3}$ - existing state of the art is highlighted

Performance Trends. Normal sequence performance is inline with Table 1 and demonstrates approach capability in its easiest state. A strict “typical” GEI definition, $tGEI_1$, is marginally weaker performance wise suggesting some intra-class variance loss; increasingly lenient $tGEI_{2,3}$ perform similarly. State of the art performance is achieved with a larger threshold T_h given the difference between normal sequences and “typical” GEI is relatively small. Carrying condition sequence performance is close to state of the art with higher leniency “typical” GEI definitions - similar to normal sequence performance; note each definition tapers off with increasing threshold T_h . Clothing sequence performance opposes that of carrying condition sequences where performance is best, and exceeds the state of the art by 11% with a strict $tGEI_1$ definition; performance again tapers off with increasing threshold T_h .

The performance difference between normal and covariate factor sequences (carrying condition and clothing) is attributed to threshold T_h attempting to remove residual ringing (Fig. 2d-e); note smaller threshold T_h values can classify large portions of the GEI as covariate factor influenced areas therefore removing more of the GEI prior to dimensionality reduction and classification. Covariate factors contain motion and despite their removal in the proposed approach residual ringing exists; its visual similarity to occluded dynamic arm motion causes an increase in Euclidean distance during classification. Across sequence type, $tGEI_{1,2}$ definitions perform equally, where the former provides a new state of the art and the latter performs more steadily across sequence type. Leniency in “typical” GEI definitions is a double edged sword - benefiting some whilst detrimental to other sequences.

Comparison to State of the Art. The proposed approach yields a significant improvement of 11% over state of the art in Table 1 for clothing sequences; remaining test data ranks close to state of the art. When comparing approaches, silhouette quality causes performance fluctuations due to their associated extraction method; despite GEI noise mitigation via space- and time-normalisation, intensity values may be affected when holes or missing portions of the silhouette exist. Despite these promising results, as well as previous research, it is clear that exceeding 94% during covariate factor sequences remains an open problem.

5 Conclusion and Future Work

This paper proposes a novel approach to mitigating the effects of covariate factors by learning the appearance of a “typical” GEI (containing no covariate factors) in a simple and efficient manner. Covariate factor influenced areas are highlighted by computing the difference between test GEIs and a “typical” GEI varying in leniency where a threshold provides the ultimate decision; such areas form a mask which mitigates their influence in test and training data to ensure dimensionality reduction and classification are performed on areas free of covariate factors. A significant improvement of 11% is made over state of the art performance for clothing varying GEIs; remaining test data ranks close to state of the art. This research seeks further validation from the TUM Gait from Audio, Image and Depth (GAID) database [24, 25]. Nevertheless, equal and high performance across test sequence type remains an open problem. Future work points towards enhanced detection of covariate factor influenced areas and their removal.

References

1. Murray, M., Drought, A., Kory, R.: Walking patterns of normal men. *The Journal of Bone and Joint Surgery* 46, 335–360 (1964)
2. Cutting, J., Kozlowski, L.: Recognising friends by their walk: gait perception without familiarity cues. *Bulletin of the Psychonomic Society* 9, 353–356 (1977)
3. Rudoy, D., Zelnik-Manor, L.: Viewpoint selection for human actions. *International Journal of Computer Vision* 97, 243–254 (2012)
4. Bashir, K., Xiang, T., Gong, S.: Feature selection for gait recognition without subject cooperation. In: *Proceedings of the British Machine Vision Conference (BMVC)* (2008)
5. Bashir, K., Xiang, T., Gong, S.: Gait representation using flow fields. In: *Proceedings of the British Machine Vision Conference (BMVC)* (2009)
6. Bashir, K., Xiang, T., Gong, S.: Gait recognition without subject cooperation. *Pattern Recognition Letters* 31, 2052–2060 (2010)
7. Zhang, E., Zhao, Y., Xiong, W.: Active Energy Image plus 2DLPP for gait recognition. *Signal Processing* 90, 2295–2302 (2010)
8. Yogarajah, P., Condell, J., Prasad, G.: PRWGEI: Poisson random walk based gait recognition. In: *7th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 662–667 (2011)

9. Huang, X., Boulgouris, N.: Gait recognition with Shifted Energy Image and structural feature extraction. *IEEE Transactions on Image Processing* 21, 2256–2268 (2012)
10. Wang, C., Zhang, J., Wang, L., Pu, J., Yuan, X.: Human identification using temporal information preserving gait template. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 2164–2176 (2012)
11. Li, X., Chen, Y.: Gait recognition based on Structural Gait Energy Image. *Journal of Computational Information Systems* 9, 121–126 (2013)
12. Li, N., Xu, Y., Yang, X.: Part-based human gait identification under clothing and carrying condition variations. In: International Conference on Machine Learning and Cybernetics (ICMLC), vol. 1, pp. 268–273 (2010)
13. Yu, S., Tan, D., Tan, T.: A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In: 18th International Conference on Pattern Recognition (ICPR), vol. 4, pp. 441–444 (2006)
14. Zheng, S., Zhang, J., Huang, K., He, R., Tan, T.: Robust view transformation model for gait recognition. In: 18th IEEE International Conference on Image Processing (ICIP), pp. 2073–2076 (2011)
15. Lee, L., Grimson, W.: Gait analysis for recognition and classification. In: Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 148–155 (2002)
16. Dockstader, S., Berg, M., Tekalp, A.: Stochastic kinematic modeling and feature extraction for gait analysis. *IEEE Transactions on Image Processing* 12, 962–976 (2003)
17. Yoo, J., Nixon, M.: Automated markerless analysis of human gait motion for recognition and classification. *ETRI Journal* 33, 259–266 (2011)
18. Dempster, W., Gaughran, G.: Properties of body segments based on size and weight. *American Journal of Anatomy* 120, 33–54 (1967)
19. Drillis, R., Contini, R.: Body segment parameters. Office of Vocational Rehabilitation, Department of Health, Education and Welfare, New York, Report No. 1163.03 (1966)
20. Han, J., Bhanu, B.: Individual recognition using Gait Energy Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 316–322 (2006)
21. Martín-Félez, R., Xiang, T.: Gait recognition by ranking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part I*. LNCS, vol. 7572, pp. 328–341. Springer, Heidelberg (2012)
22. Bashir, K., Xiang, T., Gong, S.: Gait recognition using Gait Entropy Image. In: 3rd International Conference on Crime Detection and Prevention (ICDP), pp. 1–6 (2009)
23. van der Maaten, L.: Matlab toolbox for dimensionality reduction
24. Hofmann, M., Bachmann, S., Rigoll, G.: 2.5D gait biometrics using the Depth Gradient Histogram Energy Image. In: 5th IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), pp. 399–403 (2012)
25. Hofmann, M., Geiger, J., Bachmann, S., Schuller, B., Rigoll, G.: The TUM gait from audio, image and depth (GAID) database: multimodal recognition of subjects and traits. *Journal of Visual Communication and Image Representation* (in press, 2013)

Direct Encoding for Sampled Color Pictures with Location Consideration

Chulhee Lee, Jaehoon Lee, and Guiwon Seo

Dept. Electrical and Electronic Engineering, Yonsei University, Seoul, Korea

Abstract. Single image sensors with a color filter array (CFA) have been widely used in digital cameras, digital camcorders, and smart phones. These sensors generate sampled RGB color images and interpolation produces full resolution RGB color images. Typically, color format conversion is applied to convert the RGB images to Yuv images and these Yuv images are then encoded. However, the interpolation process followed by color format conversion increases the data size by 50~100%. In order to avoid this potential inefficiency, it was proposed to encode sampled RGB color images without interpolation. However, due to location mismatching, coding improvement has not been consistent. In this paper, we consider the location differences in intra-mode prediction and preliminary experiments show that consistent improvement was obtained.

1 Introduction

Presently, most consumer cameras such as digital cameras and smart phones use a single image sensor with a color filter array (CFA). One of the most widely used CFA patterns is the Bayer CFA (Fig. 1). In the Bayer CFA, half of the total pixels are green pixels. Red and blue pixels represent 25% of the total number of pixels, respectively.



Fig. 1. Bayer color filter array (CFA)

In most cameras, the interpolation process (i.e., demosaicking) produces full resolution RGB images. In most applications, color format conversion is applied to convert these full resolution RGB images to Yuv images. Depending on the color formats (e.g, Yuv 422 or Yuv 420), the final Yuv image is still usually 50% or 100% larger than the original sampled RGB image (Fig. 2). This increase in data size may cause coding inefficiencies.

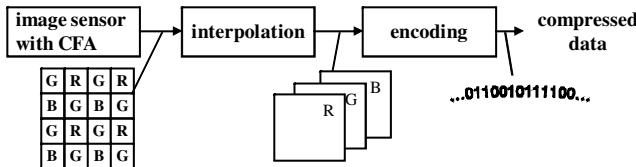


Fig. 2. Conventional method

To address these coding inefficiencies, it was proposed to directly encode sampled RGB images without demosaicking [1-8]. Figure 3 illustrates this process.

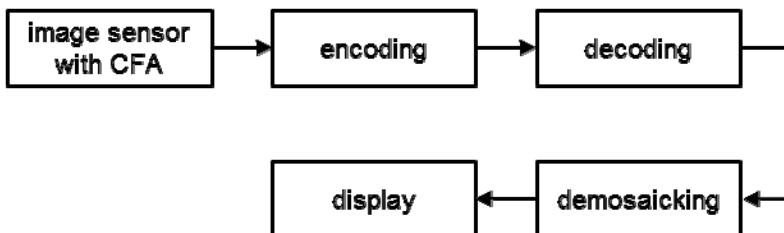


Fig. 3. Direct encoding for CFA without demosaicking

In the Bayer CFA, one may treat a 2 by 2 sub-block as a unit. In particular, Lee and Ortega proposed to generate two Y values, one u value and one v value from a 2×2 block with two green pixels, one red pixel and one blue pixel, as shown in Fig. 4 [3].



Fig. 4. RGB to Yuv conversion [3]

For example, the Yuv images can be generated from sampled RGB images using ITU-R Recommendation BT.709 [9] as follows:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0.7152 & 0 & 0.0722 & 0.2126 \\ 0 & 0.7152 & 0.0722 & 0.2126 \\ -0.168 & -0.168 & 0.432 & -0.1 \\ -0.279 & -0.279 & -0.056 & 0.615 \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ B \\ R \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 128 \\ 128 \end{bmatrix}$$

In addition to BT.709, we also tested the JPEG [10] and YCoCg [11] color coding methods.

This approach has been explored for still image coding [3] and video coding [8]. One problem with this RGB to Yuv conversion of Fig. 4 is that the vertical location of G2 is different from that of Y2. Consequently, this mismatching can cause errors in motion compensation, resulting in prediction problems and large residual errors. In this paper, we consider this location mismatching in intra-mode video coding. With such consideration, we were able to obtain consistent improvement in terms of coding efficiency.

2 Proposed Method

2.1 Intra-mode Coding

Fig. 5 illustrates the location mismatching problem of Fig. 4. Pixels on different lines are placed on the same lines during the RGB to Yuv conversion.

G1		G1		G1		G1	
	G2		G2		G2		G2
G1		G1		G1		G1	
	G2		G2		G2		G2
G1		G1		G1		G1	
	G2		G2		G2		G2
G1		G1		G1		G1	

Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2
Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2
Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2
Y1	Y2	Y1	Y2	Y1	Y2	Y1	Y2

Fig. 5. Location mismatching problem

The vertical location mismatching of Fig. 5 can cause coding inefficiencies. To address the mismatching problem, we took into account the actual G-pixel locations of the corresponding Y pixels. In an intra-mode prediction of H.264, the pixels were located as shown in Fig. 6(a). However, as previously explained, the actual locations are as shown in Fig. 6(b). Consequently, unnecessary artificial high frequency components sometimes appeared, which reduced coding efficiency. Fig. 7 illustrates such an example. When there was a boundary, an image produced by the coding method shown in Fig. 4 generated artificial high frequency components in the Y image (Fig. 7). Fig. 8 shows such an example when working with real images.

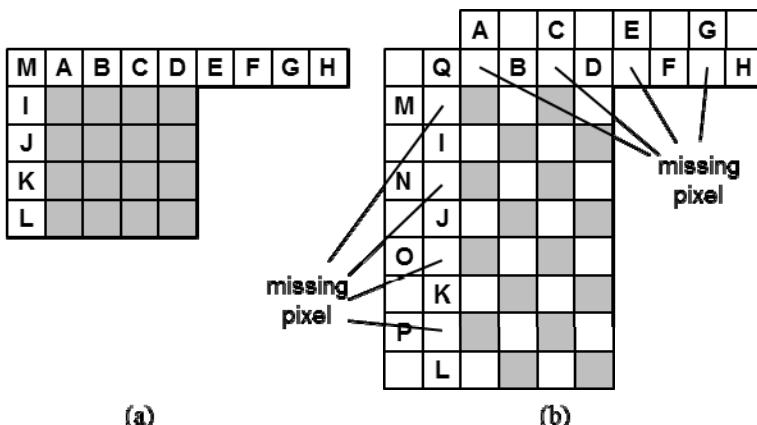


Fig. 6. (a) Conventional intra-mode structure of H.264. (b) Proposed modification to the intra-mode structure of H.264

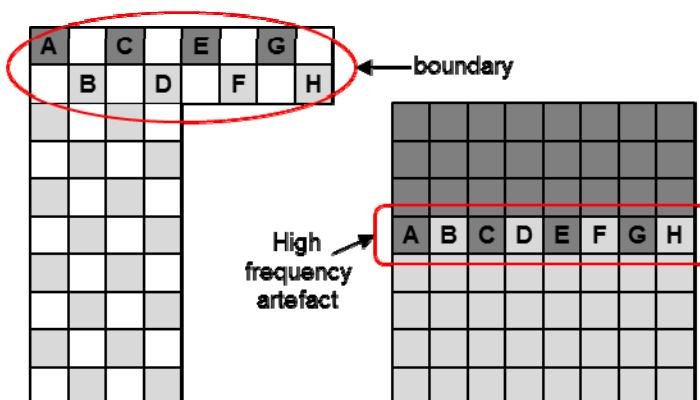


Fig. 7. Artificial high frequency artefacts caused by location mismatching



Fig. 8. High frequency artefacts caused by location mismatching

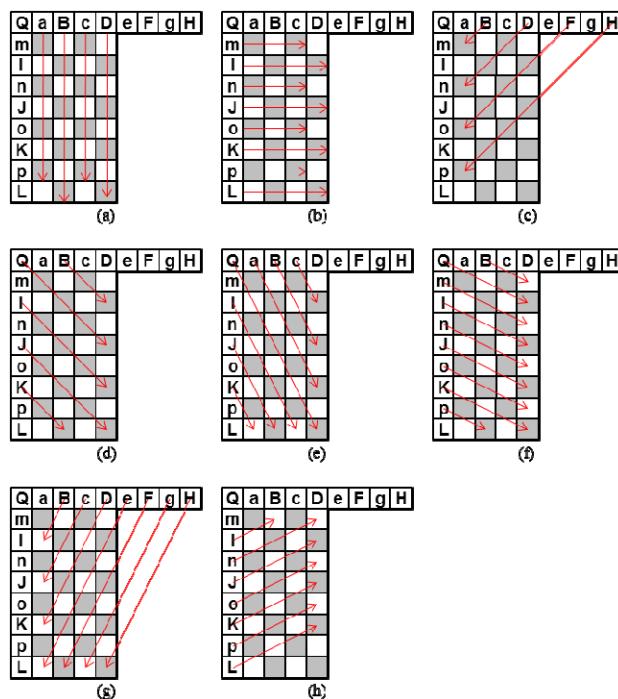


Fig. 9. Proposed intra-mode prediction. (a) vertical, (b) horizontal, (c) diagonal down left, (d) diagonal down right, (e) vertical right, (f) horizontal down, (g) vertical left, (h) horizontal up

In order to address this problem, we modified the intra-mode coding structure as shown in Fig. 6(b), where we placed Y pixels on the actual locations. Then, we performed the intra-mode predictions of H.264 as illustrated in Fig. 9. In Figs. 6 and 9, the missing pixels (marked with small characters in Fig. 9: a, c, e, g, m, n, o, p) were interpolated using the already decoded pixels (upper and left blocks). Various interpolation algorithms can be used to fill in these missing pixels. For example, interpolation can be performed using Y pixels only. Alternatively, they can be interpolated using Y, u and v pixels.

2.2 Inter-mode Coding

Once intra-mode coding was performed, we used the standard inter-mode coding method of H.264 without further changes. This minimized the codec modifications and can be easily applied to many devices in the field. In the experiments, we used an IBBPBBP structure, as shown in Fig. 10.

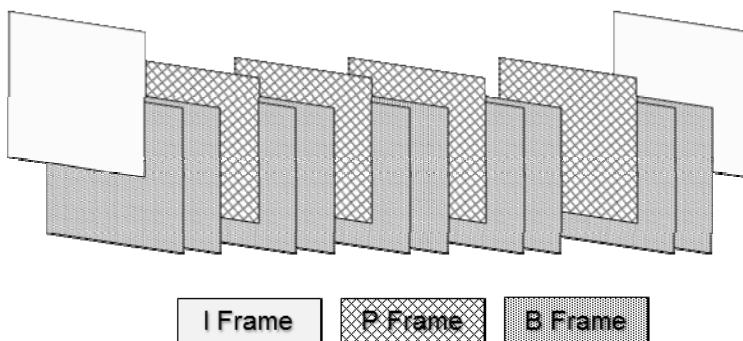


Fig. 10. Inter-mode prediction

3 Experimental Results

To generate sampled RGB video sequences, we sampled full-resolution RGB video sequences in accordance with the Bayer CFA. Then, we applied a demosaicking method to the sampled RGB video sequences, which produced interpolated full resolution RGB video sequences. One of these full resolution RGB video sequences was used as a reference video sequence. By applying color coding to the interpolated full resolution RGB video sequences, we generated conventional Yuv 420 video sequences. These sequences were encoded using a conventional encoding method and used as a baseline. Table 1 shows the test video sequence description used in the experiments.

Table 1. Test sequences (Viper test set)

Name	Format	Size	Frames
Plane(PL)	RGB	HD(1920x1080)	60
Waves(WV)	RGB	HD(1920x1080)	60
Freeway(FW)	RGB	HD(1920x1080)	60
Man in restaurant(MR)	RGB	HD(1920x896)	60
Playing Cards(PC)	RGB	HD(1920x1080)	60
Rolling Tomatoes (RT)	RGB	HD(1920x1080)	60
Table Setting(TC)	RGB	HD(1920x1080)	60

To generate signals for direct encoding, we generated Yuv 422 video sequences by directly applying the encoding method of [3] to the sampled RGB sequences. These sequences were encoded by using H.264 (Yuv 422 mode) with the proposed intra-mode coding process.

In order to compare the performance, we computed the RGB PSNR values. The conventional Yuv 420 coding method produced Yuv420 signals, from which full resolution RGB signals were produced by applying color format conversion. In the proposed method, the decoding procedure produced sampled RGB video sequences in the Bayer pattern. Then, we applied the demosaicking method to the decoded sampled RGB video sequences, which produced full resolution RGB video sequences. The RGB PSNRs were computed between the reconstructed RGB video sequences and the uncompressed interpolated RGB video sequences (Fig. 11) to avoid the influence of the demosaicking process.

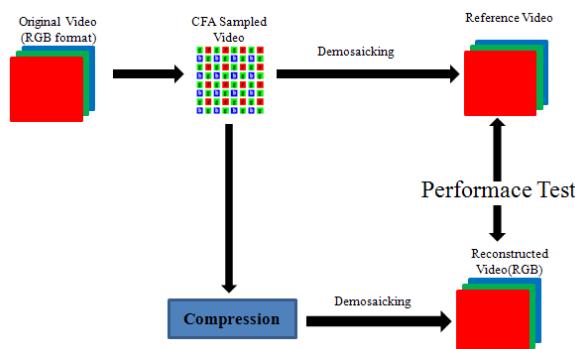
**Fig. 11.** Performance test (RGB PSNR computation)

Table 2 shows the performance comparison between the proposed method and the conventional method (Yuv 420) when all frames were encoded in the intra-mode. It can be seen that the proposed method showed consistent improved performance compared to the conventional Yuv420 coding method. The YCoCg color coding method showed the best performance. Fig. 11 shows the RD curve of *Rolling Tomatoes*. As can be seen, the proposed method showed noticeable improvements. Table 3 shows the performance comparison when inter-mode coding was used. Also, in this mode, the proposed method showed noticeable improvements.

Table 2. Performance Comparison

(BDRate: Bjontegaard Delta Rate, BDPSNR: Bjontegaard Delta PSNR).

Sequence	BT.709		YCoCg		JPEG	
	BDRate	BDPSNR	BDRate	BDPSNR	BDRate	BDPSNR
FR	-8.53%	0.434dB	-10.64%	0.499dB	-10.46%	0.461 dB
MR	-8.01%	0.432dB	-14.80%	0.713 dB	-9.49%	0.483 dB
PL	-7.39%	0.318dB	-10.26%	0.392 dB	-5.08%	0.189 dB
PC	-25.14%	1.044dB	-28.19%	0.928 dB	-29.97%	1.085 dB
RT	-22.02%	0.473dB	-29.36%	0.602 dB	-25.93%	0.538 dB
TS	-47.72%	1.212dB	-43.41%	1.097 dB	-51.33%	1.284 dB
WV	-12.80%	0.377dB	-25.86%	0.750 dB	-16.64%	0.462 dB
AVG	-18.80%	0.613dB	-23.21%	0.712 dB	-21.27%	0.643 dB

**Fig. 12.** RD curve of *Rolling tomatoes*

Table 3. Performance comparison (inter-mode)

Sequence	BDRate	BDPSNR
FR	-3.27%	0.167dB
MR	-0.90%	0.161dB
PL	13.64%	-0.301dB
PC	-28.92%	0.771dB
RT	-30.94%	0.473dB
TS	-55.77%	1.044dB
WV	-12.20%	0.360dB
AVG	-16.91%	0.382dB

4 Conclusions

In this paper, we proposed direct encoding for sampled RGB video sequences obtained using a single sensor with a color filter array. From a 2×2 subblock, two Y pixels, one u pixel and a v pixel were generated. During this conversion, pixels on different lines were mapped onto the same line and artificial high frequency components were produced. To address this problem, we considered the actual locations of the Y pixels during intra-mode coding. Experimental results show that the proposed method consistently produced improved performance compared to conventional Yuv420 coding.

References

1. Lee, C., Lee, J.: Efficient compression for sampled color images. *IEEE Trans. Electron Devices* 55(4), 2090–2096 (2009)
2. Toi, T., Ohita, M.: A subband coding technique for image compression in single CCD cameras with Bayer color filter arrays. *IEEE Trans. Consum. Electron.* 45(1), 176–180 (1999)
3. Lee, S.-Y., Ortega, A.: A novel approach of image compression in digital cameras with a Bayer color filter. In: *IEEE Intl. Conf. Image Processing*, vol. 3, pp. 482–485 (2001)
4. Koh, C.C., Mukherjee, J., Mitra, S.K.: New efficient methods of image compression in digital cameras with color filter array. *IEEE Trans. Consum. Electron.* 49(4), 1448–1456 (2003)
5. Lukac, R., Plataniotis, K.N.: Fast video demosaicing solution for mobile phone imaging application. *IEEE Trans. Consum. Electron.* 51(2), 675–681 (2005)

6. Zhang, L., Wu, X., Bao, P.: Real-time lossless compression for mosaic video sequences. *Real-Time Imag.* 11(5-6), 370–377 (2005)
7. Lian, N.-X., Chang, L., Zagorodnov, V., Tan, Y.-P.: Reversing Demosaicking and Compression in Color Filter Array Image Processing: Performance Analysis and Modeling. *IEEE Trans. Image Process.* 15(11), 3261–3278 (2006)
8. Lee, C., Lee, J.: Direct Video Encoding for CFA. In: Proceeding of the IEEE International Conference on Industrial Technology (2013)
9. ITU-R Recommendation BT.709, Parameter values for the HDTV standards for production and international programme exchange (2002)
10. ITU-R Recommandataion T.871, Information technology – Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF) (2011)
11. Doutre, C., Lin, B., Tzvetkov, V.: Compression of Colour Filter Array Video Sequences, The University of British Columbia, pp. 1–18 (2005)

Real-Time Hand Gesture Recognition for Uncontrolled Environments Using Adaptive SURF Tracking and Hidden Conditional Random Fields

Yi Yao and Chang-Tsun Li

Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK

Abstract. Challenges from the uncontrolled environments are the main difficulties in making hand gesture recognition methods robust in real-world scenarios. In this paper, we propose a real-time and purely vision-based method for hand gesture recognition in uncontrolled environments. A novel tracking method is introduced to track multiple hand candidates from the first frame. The movement directions of all hand candidates are extracted as trajectory features. A modified HCRF model is used to classify gestures. The proposed method can survive challenges including: *gesturing hand out of the scene, pause during gestures, complex background, skin-coloured regions moving in background, performers wearing short sleeve and face overlapping with hand*. The method has been tested on Palm Graffiti Digits database and Warwick Hand Gesture database. Experimental results show that the proposed method can perform well in uncontrolled environments.

1 Introduction

Hand gesture recognition is an intuitive way for facilitating Human Computer Interaction (HCI). However, its robustness against uncontrolled environments is widely questioned. Many challenges exist in real-world scenarios which can largely affect the performance of appearance based methods, including presence of cluttered background, moving objects in background, gesturing hand out of the scene, pause during the gesture, and presence of other people or skin-coloured regions, etc. This is the reason why the majority of works in hand gesture recognition are only applicable in controlled environments.

There have been few attempts for recognising hand gestures in different uncontrolled environments. Bao et al. [1] proposed an approach using SURF [2] as features to describe hand gestures. The matched SURF point pairs between adjacent frames are used to produce the hand movement direction. This method only works under the assumption that the gesture performer occupies a large proportion of the scene. If there are any other moving objects at the same scale of the gesture performer in the background, the method will fail. Elmezain et al. [3] proposed a method which segments hands from the complex background using 3D depth map and colour information. The gesturing hand is tracked by using Mean-Shift and Kalman filter. Fingertip detection is used for locating the target hand. However, this method can only deal with the cluttered background and is unable to cope with other challenges mentioned

earlier. Alon et al. [4] proposed a framework for spatiotemporal gesture segmentation. Their method is tested in uncontrolled environments with other people moving in the background. This method tracks a certain number of candidate hand regions. The number of candidate regions can largely affect the performance of the method, which must be specified beforehand, making it unrealistic in real-world scenarios. Two other works ([5], [6]) also tested their methods on the database of [4]. But none have outperformed [4] on their database.

In this paper, we propose a method for hand gesture recognition in uncontrolled environments. A novel tracking method called Adaptive SURF Tracking is introduced to extract hand trajectories. A model based on Hidden Conditional Random Fields (HCRF) [7] is trained to classify hand trajectories into 10 digits gesture classes.

2 Adaptive SURF Tracking

One of the key differentiating features of our proposed method from other existing methods is that the exact location of the gesturing hand is not required. Similar to [4], our method also keeps tracks of multiple candidates of hand regions. We call this tracking method Adaptive SURF Tracking. In the first frame of the video sequence, skin colour cues are used to detect possible skin colour regions as the initial regions of interests (ROI). After the first frame, key SURF points are extracted from every ROI and matched against their counterparts in the next frame. The dominant movement orientation of every ROI is then extracted from each frame to form the candidate hand trajectory vector, which is used as the input of the HCRF model.

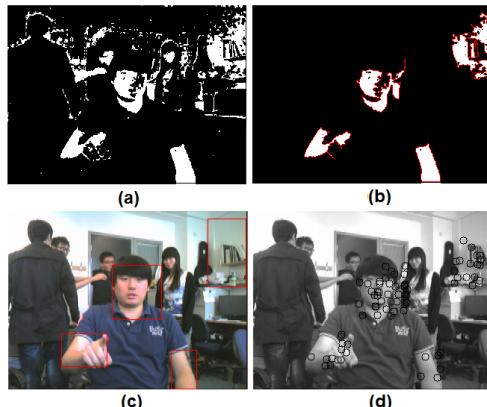


Fig. 1. Processing of the first frame, (a): skin colour binary image; (b) : result of denoising; (c) : initial ROIs; (d): SURF key points within initial ROIs

Figure 1 illustrates the mechanism for processing the first frame. The method detects faces in the frame, by using the Viola-Jones face detector [8]. Then thresholds of skin colour in the HSV colour space are estimated using pixels in the detected facial regions. Those thresholds are later used to produce a skin-colour binary image (Figure 1(a)) in the processing of every frame in the video. Hence the method can adapt to

different illumination conditions. If no faces are detected in the first frame, a Gaussian Mixture Model (GMM) in the RGB colour space, which is inferred from a large database of skin and non-skin pixels [9], will be used to calculate the skin-colour binary image, until at least one valid face region is detected in later frames. All closed contours are then detected in this binary image. A denoising process is performed on the skin-colour binary image by deleting all the interior contours and the contours of the areas smaller than a threshold T_{dsr} (Figure 1(b)), where

$$T_{dsr} = \bar{A}_f \times 0.25 \quad (1)$$

\bar{A}_f is the average area of all detected facial regions in the first frame. The minimum bounding rectangles from this binary image are taken as the initial ROIs of the first frame, as shown in Figure 1(c). Subsequently, SURF points are extracted from the first frame and those key points within the ROIs are kept (circles in Figure 1(d)).

Starting from the second frame, SURF features are extracted from the whole image of the current frame and matched against their counterparts in the previous frame, as shown in Figure 2(a). Once the matched pairs are calculated, a pruning process is performed on all matched pairs. Only those pairs with a displacement within a certain range between the matched key points in the current frame and the matching points in the previous frame are preserved. All the matched pairs which are located in stationary regions (e.g. in the face region) or regions that do not move beyond the lower bound of this displacement range are dropped. On the other hand, if a matched key point has displaced beyond the upper bound of the displacement range in the next frame, it most likely is a mismatch. This is a reasonable assumption because if an object moves too much within such a short period of time, it is unlikely to be the target hand. Various displacement ranges have been tested and we found that the range between 3 and 40 pixels is empirically feasible. An example of pruning is shown in Figure 2(b).

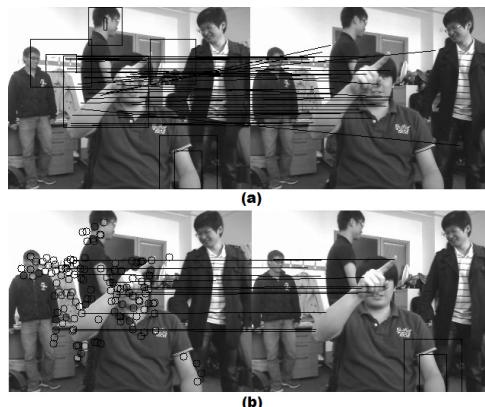


Fig. 2. Pruning process. (a): matched key point pairs from one of the ROIs, between previous frame (left) and current frame (right). (b): the remaining matched key point pairs after pruning.

After the pruning process, the ROIs in the current frame are drawn. For the SURF key points in one ROI in the previous frame, there are key SURF points in the current frame that matched to them. The corresponding ROI in the current frame is defined as the minimum bounding rectangle of these matched key points in the current frame. During every frame, according to the number of the remaining matched key points P after pruning and the area of the new ROIs A , the boundaries of the new ROIs may be extended by e pixels. The value of e is set as:

$$e = \begin{cases} 0, A \geq 3000 \\ 20, 500 \leq A < 3000 \\ 20, A < 500 \wedge P > 5 \\ 30, A < 500 \wedge 0 < P \leq 5 \\ 40, P = 0 \end{cases} \quad (2)$$

Instead of only keeping the matched key points in each of the new ROIs of the current frame, all key SURF points within these new ROIs are preserved for matching with those in the next frame. The reason for enlarging the ROIs is that they may not cover the entire area of the target hand candidates. Hence, in order to get as many tracking features as possible from the current ROIs of target hand candidates, the ROIs need to be enlarged to make sure that the new ROIs cover the corresponding hand candidates.

For every frame, the dominant movement direction of each ROI will be calculated as the hand trajectory feature of this hand candidate. Assume we have P matched SURF pairs between frames $t-1$ and t after pruning in a ROI, denoted by $M_t = \{\langle S_{t-1}^1, S_t^1 \rangle, \langle S_{t-1}^2, S_t^2 \rangle, \dots, \langle S_{t-1}^P, S_t^P \rangle\}$, where $\langle S_{t-1}^i, S_t^i \rangle$ is the i^{th} pair. The dominant movement direction of the r^{th} ROI in frame t is defined as:

$$\text{drt}(t, r) = \arg \max_d \{q_d\}_{d=1}^D \quad (3)$$

where, $\{q_d\}_{d=1}^D$ is the histogram of the movement direction of all matched SURF key point pairs in this ROI, and d indicates the index of directions. q_d is the d^{th} bin of the histogram. Each bin has an angle interval with range α , and $D = 360/\alpha$. We have tested various values for α and found that 20° produces best results. Definition of q_d is:

$$q_d = C \sum_{p=1}^P k\left(\|S_t^p\|^2\right) \delta(S_t^p, d) \quad (4)$$

where, $k(x)$ is a monotonic kernel function which assigns smaller weights to those key SURF points farther away from the centre of this ROI; $\delta(S_t^p, d)$ is the Kronecker delta function which has value 1 if the movement direction of $\langle S_{t-1}^p, S_t^p \rangle$ falls into the d^{th} bin; and the constant C is a normalisation coefficient defined as:

$$C = 1 \left/ \sum_{p=1}^P k \left(\|S_t^p\|^2 \right) \right. \quad (5)$$

Another desirable feature of our proposed method is that we only use hand movement direction as hand trajectory feature. Since speed and location of gestures are used as features in [4], to make their method less sensitive to the location and size of the gestures, face detection is used to estimate the location and scale of the gesture performers. Unlike [4], the location and speed of hand candidates are not used to describe hand gestures, hence our method does not need to estimate the location and scale of the gestures, the modified HCRF model is also not sensitive to the length of the gestures, which makes the proposed method invariant against the location, speed and size of the hand gestures.

3 Gesture Classification

After the tracking stage, once the movement direction vectors, namely the input sequences for HCRF model, of every hand candidates in the videos are extracted, they are put into a multi-class chain HCRF model as feature vectors, as shown in Figure 3. The videos are naturally segmented as one single frame is a single node in HCRF model. HCRF has been proven to be one of the strongest discriminative models with hidden states [7]. In this paper, since the task is recognising a set of hand-signed digits $Y = [y_0, y_1, \dots, y_9]$ (as shown in Figure 4), we define the hidden states to be the strokes of gestures. There are in total 13 states in the HCRF model for our own database, and 15 states in the Palm Graffiti Digits database [4]. Figure 3 shows 4 of the 13 states in our Warwick Hand Gesture Database, which form the gesture of digit 4. The optimisation scheme used in our HCRF model is Limited Memory Broyden–Fletcher–Goldfarb–Shanno method [10]. In our experiments, the weight vector θ is initialised with the mean value, and the regularisation factors are set to zero.

As one sequence of the movement direction represents the trajectory direction vector of one hand candidate, a video clip X with R ROIs can have multiple sequences: $X = [x_1, x_2, \dots, x_R]$. Hence we modified the original HCRF model to suit our special case of multiple sequences for one video. In the original HCRF model, the probability of gesture y , given the video clip X , hidden states h and weight vector θ , is calculated by,

$$P(y|X,\theta) = \sum_h P(y,h|X,\theta) = \frac{\sum_h \exp\{\Psi(y,h,X;\theta)\}}{\sum_{y',h} \exp\{\Psi(y',h,X;\theta)\}} \quad (6)$$

where $\Psi(y,h,X;\theta)$ is the potential function. Follow [7], we define the partition function :

$$Z(y|X,\theta) = \sum_h \exp\{\Psi(y,h,X;\theta)\} \quad (7)$$

For multiple sequences video $X = [x_1, x_2, \dots, x_R]$, the new partition function is defined:

$$Z'(y|X,\theta) = \arg \max_{x_r} \sum_h \exp \{ \Psi(y, h, x_r; \theta) \} \quad (8)$$

Hence the probability of gesture y , given the video clip X is:

$$P(y|X,\theta) = \frac{Z'(y|X,\theta)}{\sum_{y'} Z'(y'|X,\theta)} \quad (9)$$

and we take the final gesture to be $\arg \max_{y \in Y} P(y|X,\theta)$. Namely, the final gesture

label assigned to this video clip is the one with the highest partition value among all the partitions of all sequences during this gesture.

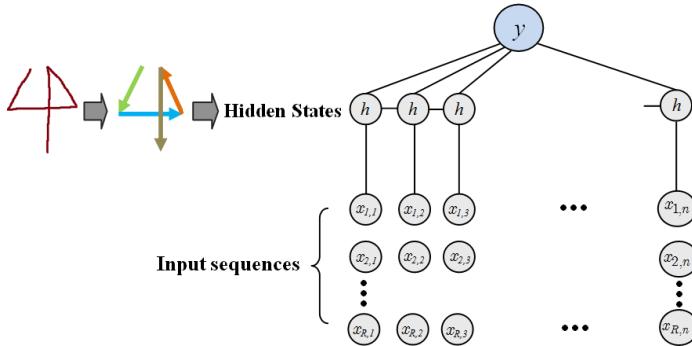


Fig. 3. HCRF model, the hidden states are defined as strokes of gestures, y is the gesture label, node $x_{R,n}$ means the movement direction of the R^{th} hand candidate in the n^{th} frame of the video

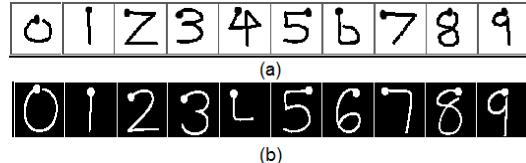


Fig. 4. The hand gesture sets, (a) is defined in our own database (Warwick Hand Gesture Database), (b) is from Palm Graffiti Digits Database[4]

4 Experiments

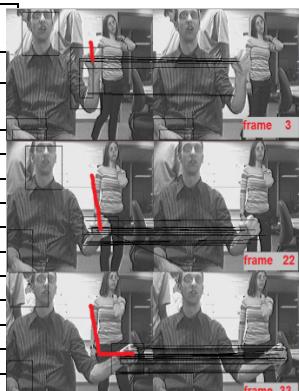
We conducted two experiments on two databases. First experiment is on the Palm Graffiti Digits database used in [4]. This database contains 30 video samples for training, 3 samples from each of 10 performers that wear gloves. Each sample captures the performer signing digits 0-9 each for once. There are two test sets, the "hard" and "easy" sets. There are 30 videos in the easy set, 3 from each of 10 performers, and 14 videos in the hard set, 2 from each of 7 performers. The contents of both test sets are the same as the training set, except that performers do not wear gloves, and there are 1

Table 1. Results of experiment on the easy set of the Palm Graffiti Digits database [4]

Gesture class	Easy Set				
	Training samples	Testing samples	Recognition Results		
			True	Detected	Accuracy (%)
0	30	30	30	34	100.00
1	30	30	30	31	100.00
2	30	30	28	28	93.33
3	30	30	28	28	93.33
4	30	30	30	30	100.00
5	30	30	30	32	100.00
6	30	30	25	26	83.33
7	30	30	29	30	96.67
8	30	30	30	30	100.00
9	30	30	27	31	90.00
Overall	300	300	287	300	95.67

Table 2. Results and sample (Gesture 4) of experiment on the hard set of the Palm Graffiti Digits database [4]

Gesture class	Hard Set				
	Training samples	Testing samples	Recognition Results		
			True	Detected	Accuracy (%)
0	30	14	11	15	78.57
1	30	14	13	13	92.86
2	30	14	13	15	92.86
3	30	14	13	14	92.86
4	30	14	13	14	92.86
5	30	14	14	16	100.00
6	30	14	6	6	42.86
7	30	14	12	13	85.71
8	30	14	13	16	92.86
9	30	14	13	18	92.86
Overall	300	140	121	140	86.43



to 3 people moving back and forth in the background (Table 2) in hard set. The specifications of the videos are: 30Hz, and resolution of 240×320 pixels. The results of the proposed method are shown in Table 1 and 2. Follow [4], the proposed method also does not aware the starting and ending frames of each gesture. A simple gesture spotting rule is applied. From last ending point up to the current frame, if the partition from all gesture classes are lower than a threshold, this part of the video will be treated as nonsign gesture. When at least one gesture class produces partition higher than the threshold, the proposed method will treat this frame as the starting frame of the gesture, until partitions from all gesture classes are lower than the threshold again. Compared with [4],[5],[6] and [1], the proposed method produced better accuracy on both easy and hard set, as shown in Table 3 and Figure 6.

The reasons the proposed method outperformed [4] are that: Firstly, in [4], the number of hand candidates must be specified beforehand, which is often unrealistic in real-world scenarios. The proposed method does not require the prior knowledge on the contents of the background. When processing the first frame, the eligible skin-coloured regions are taken as initial ROIs. Hence our method can adaptively detect

Table 3. All reported experimental results ([4],[5] and [6]) on Palm Graffiti Digits database, and results produced by this paper (the proposed method and method of [1])

10 Palm Graffiti Digits database [4]		
	Easy set	Hard set
Correa et al. RoboCup 2009 [5]	75.00%	N/A
Malgireddy et al. CIA 2011 [6]	93.33%	N/A
Alon et al. PAMI 2009 [4]	94.60%	85.00%
Bao et al. ICEICE 2011 [1]	52.00%	28.57%
The proposed method	95.67%	86.43%

number of hand candidates. Secondly, the people or other moving objects entering the scene after the first frame have no impact on the proposed method. Those objects will not be matched to the SURF features of the objects (including gesturing hand) that exist since the first frame. In [4], all the hand candidates in all frames have to be tracked, which makes the method inapplicable to the real-world scenarios.

We collected a more challenging database called Warwick Hand Gesture Database (see Figure 2 for example) to demonstrate the performance of the proposed method under new challenges. 10 gesture classes as in Figure 4(a) are defined for our database. This database consists of two testing sets, namely "easy" and "hard" sets. There are 300 video samples for training, 3 samples were captured from each of 10 performers for each gesture. There are 1000 video samples in total for testing. For each gesture, 10 samples were collected from each of 10 performers. The specifications of videos are the same as Palm Graffiti Digits database. Similar to the Palm Graffiti Digits database, the hard set of our database captures performers wearing short-sleeve tops with cluttered backgrounds. The differences are: No gloves in training set. Instead of 1-3 people, we had 2-4 people moving in the background, and there are new challenges in the clips, including: gesturing hand out of scene and pause during gesture. Since the work of [1] is the one most similar to the proposed method, we compared the performance between these two methods (Table 4 and Figure 6).

Table 4. The performances of method of [1] and the proposed method on Warwick Hand Gesture Database

Warwick hand gesture database		
	Easy set	Hard set
Bao et al. ICEICE 2011 [1]	71.00%	18.20%
The proposed method	93.80%	85.40%

As shown in the graph of movement direction vectors (Figure 5), the intra-class variance in our database is larger than the database of [4]. Our method still produced similar accuracy on both Warwick Hand Gesture Database and Palm Graffiti Digits Database. The reason our method can handle the new challenge of gesturing hand out of scene is that the ROI covers the arm section when the hand is out of the scene. The arm section is tracked until the frame in which the hand is back in the scene. Since, when the ROI is being redefined and enlarged in this frame, the hand section will be covered again. Therefore, the SURF features will be extracted in the new ROI

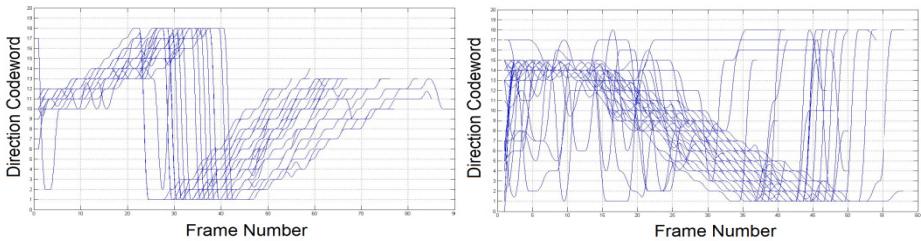


Fig. 5. Movement direction vectors for the gesture of digit 6 of the training set of: Palm Graffiti Digits database (left) and Warwick Hand Gesture database (right). The horizontal axis is the frame number while the vertical axis is the direction codeword (1-18).

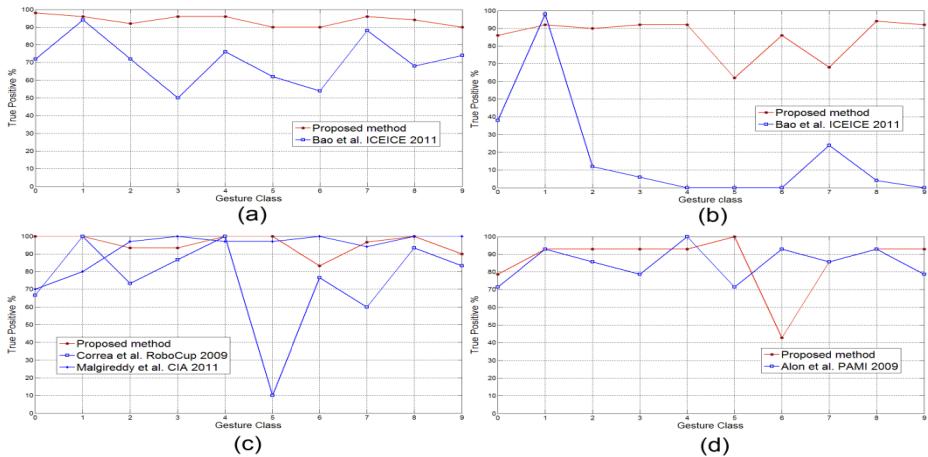


Fig. 6. The comparison of performances on (a): easy set of the Warwick hand gesture database; (b) hard set of the Warwick hand gesture database; (c): easy set of the Palm Graffiti Digits database; (d): hard set of the Palm Graffiti Digits database. The horizontal axis is the gesture label while the vertical axis is the recognition rate.

which covers the returning hand section. As for the pause during gesture challenge, the proposed method preserves the ROI when the number of moving matched SURF pairs in this ROI is 0, which means either the target is stationary or the method has lost track on this ROI. The ROI are enlarged in every frame until the number of matched SURF pairs is not 0. Hence the method can regain tracks on targets in most of the situations within several frames. As for speed, the proposed method performs on average in both experiments at: 53.00 ms/frame for easy sets, 53.75ms/frame for hard sets. That is 18.9 frames/sec and 18.6 frames/sec, respectively. Hence our method is able to perform comfortably in real time.

5 Conclusions

In this paper, we propose a real-time and purely vision-based method for hand gesture recognition in uncontrolled environments. The method can recognise hand gestures

against the complex background with 2 to 4 people moving in it. The method can handle challenges such as complex background, skin-coloured regions moving in background, performers wearing short-sleeve and face overlapping with hand. The method was tested on Palm Graffiti Digits Database [4], and achieved 95.67% on easy set, 86.43% on hard set. We also tested the proposed method on our own database with additional challenges of gesturing hand out of scene and pause during gesture. The method achieved 93.80% on easy set and 85.40% on hard set.

Acknowledgment. This work is included in the pending patent: UK patent application GB1305812.8, 28 March 2013, University of Warwick.

References

1. Bao, J., Song, A., Guo, Y., Tang, H.: Dynamic Hand Gesture Recognition Based on SURF Tracking. In: International Conference on Electric Information and Control Engineering, ICEICE (2011)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded-Up Robust Features. Computer Vision and Image Understanding (CVIU) 110(3), 346–359 (2008)
3. Elmezain, M., Al-Hamadi, A., Michaelis, B.: A Robust Method for Hand Gesture Segmentation and Recognition Using Forward Spotting Scheme in Conditional Random Fields. In: International Conference on Pattern Recognition, ICPR, pp. 3850–3853 (2010)
4. Alon, J., Athitsos, V., Yuan, Q., Sclaroff, S.: A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 1685–1699 (September 2009)
5. Correa, M., Ruiz-del-Solar, J., Verschae, R., Lee-Ferng, J., Castillo, N.: Real-Time Hand Gesture Recognition for Human Robot Interaction. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 46–57. Springer, Heidelberg (2010)
6. Malgireddy, M.R., Nwogu, I., Ghosh, S., Govindaraju, V.: A Shared Parameter Model for Gesture and Sub-gesture Analysis. In: Aggarwal, J.K., Barneva, R.P., Brimkov, V.E., Koroutchev, K.N., Korutcheva, E.R. (eds.) IWCIA 2011. LNCS, vol. 6636, pp. 483–493. Springer, Heidelberg (2011)
7. Quattoni, A., Wang, S., Morency, L.P., Collins, M., Darrell, T.: Hidden-state Conditional Random Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 1848–1852 (October 2007)
8. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57, 137–154 (2004)
9. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Skin Detection. International Journal of Computer Vision 46(1), 81–96 (2002)
10. Liu, D.C., Nocedal, J.: On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming 45(1-3), 503–528 (1989)

Examination of Hybrid Image Feature Trackers

Peter Abeles

Robotic Inception

pabeles@roboticinception.com

Abstract. Typical image feature trackers employ a detect-describe-associate (DDA) or detect-track (DT) paradigm. Intuitively, a hybrid of the two approaches inherits the benefits of each approach and possibly their defects, however this has never been demonstrated formally in a more general setting. In this paper, the stability and speed of DDA, DT, and hybrid trackers are compared and discussed using a diverse set of real-world video sequences.

1 Introduction

Image point feature tracking in video sequences is a key component in many computer vision problems, such as structure from motion and object detection/recognition. Specific applications include: scene recognition for loop-closure, face recognition, video stabilization, visual odometry, and augmented reality. The most popular feature trackers can be categorized as employing detect-describe-associate (DDA) or detect-track¹ (DT) paradigms.

DDA-based trackers operate by 1) detecting feature locations and characteristics, 2) describing each feature with local image information, and 3) associating features between images. Numerous feature detectors (Harris [1], Shi-Tomasi [2], FAST [3], MSER [4]) and local feature descriptors (SIFT [5], SURF [6], BRIEF [7]) are available for DDA trackers. Association is done by selecting pairs of feature descriptors between two images that minimize an error metric. DDA trackers are robust to temporary obstructions and abrupt changes in view, but are computationally expensive.

DT trackers operate by first detecting features, then updating each feature track location by performing a local search initialized using the last known location. Several template based DT trackers have been proposed in the past, with Kanade-Lucas-Tomasi (KLT) [8,2] being the most popular. Under nominal conditions, DT trackers are faster and more stable than DDA trackers, but if the scene changes abruptly, a DT tracker will fail and cannot recover.

Despite DT trackers having superior nominal performance, their usage is limited to a subset of problems that DDA is applied to. The reason for DDA's wider usage is its resilience and ability to recover tracks. To overcome this issue, hybrid approaches have been proposed in the past in an attempt to improve runtime speed and/or stability. Past hybrid approaches were application specific, making their results difficult to generalize.

Most practitioners in the field accept or assume these qualitative statements about the relative merits of each type of tracker. However, a direct comparison of these three

¹ More commonly referred to as “detect then track”.

approaches, which validates and quantifies this knowledge, does not exist in the literature. In this work, a performance study is presented that compares the three types of trackers. To facilitate this comparison a generic hybrid tracker is described. Real-world video sequences are used to evaluate performance, each of which is designed to stress the trackers in different ways.

2 Related Work and Background

The first step for both DDA and DT trackers is feature detection. Corner (e.g. Harris and Shi-Tomasi) and blob (e.g. Hessian Determinant [9]) detectors are commonly used to identify salient features within images for tracking purposes. After a feature has been detected, a description of that feature is extracted from its local region. DT trackers can use simple templates as descriptors due to their assumption that the scene is slowly changing. DDA make a weaker assumptions and require descriptors that can handle greater changes in appearance.

The major philosophical difference between the two approaches involves how tracks are updated. DDA perform an expensive global search for features, followed by association. DT performs a local search for each track independently.

Feature association is an expensive operation for DDA trackers. The simplest solution has a complexity of $O(N^2)$, although faster approximate alternatives are available (e.g. k-d trees [10]). An optimal solution is defined as the two features that minimize a distance metric, often the L_2 (Euclidean) norm:

$$a_i = \min_j ||F_i^0 - F_j^1|| \quad (1)$$

where F^0 and F^1 are sets of feature descriptors from two different images.

All DT trackers work by minimizing the difference between a template and the image with a local search around feature's previous location. Besides KLT, several others have been proposed [11,12]. It was pointed out by Baker and Matthews [13] that these approaches are equivalent to a first order approximation. The primary difference between each approaches lies in the types of motion models used and if the template and/or image is warped. While mathematically similar, these differences significantly affect computation performance.

KLT updates track locations using a translational motion model and minimizes the difference between the track's description and the image using the following cost function:

$$\epsilon = \int_W [I(x - d) - J(x)]^2 w \cdot dx \quad (2)$$

where W is a local region around location x , w is an optional weight, d is the translation parameter being optimized, and I and J are the first and second images in the sequence. Pyramidal approaches are used to handle larger displacements.

Unlike DDA trackers, DT trackers' runtime speed is dependent on the number of tracks and not image size. Track stability is often improved by updating the description after each frame. The downside to updating the descriptor is that image noise will cause tracks to perform a random walk.

Several hybrid trackers that employ both DT and a DDA tracker have been proposed in the past. In Uemura and Mikolajczyk [14] diverged KLT tracks are detected using a SIFT descriptor when detecting human actions. In Pilet and Saito [15] robustness is added to a region based normalized-cross-correlation (NCC) tracker by switching to KLT when association fails to find a match. To reduce image processing overhead for visual loop closing in SLAM, Pradeep et. al [16] proposed to attach SIFT descriptions to KLT tracks with periodically associating all tracks to newly detected features. Ladikos et. al [17] track the pose of known objects using a combination of template (DT type) and feature (DDA type) trackers.

In all of the just mentioned works, a direct evaluation of track performance is lacking. In those works, performance is measured indirectly as a function of the target application, e.g. localization accuracy or detection rate. Test environments are limited and focused on specific applications. They also lack a more generalized discussion of design issues and failure conditions unique to hybrid tracking.

3 Generalized Hybrid Tracker

For purposes of comparison, a generalized hybrid DDA-DT tracker with a modular design is described below. For sake of brevity this specific tracker will be referred to as the hybrid tracker. The previously mentioned hybrid trackers were designed for specific applications. The generalized hybrid tracker only specifies a high-level design and requires that specific implementations for feature detection, feature description, and DT tracking be provided to it.

In the hybrid tracker, tracking starts by spawning new tracks from detected image features. A feature track is defined by a 2D location and has two types of descriptions, namely, DDA description and DT description. The DDA description is immutable and the DT description is updated after each image is processed.

When the next image in the sequence arrives, tracks are first updated using the DT tracker. If a track fails a DT update, the track is placed in storage. Tracks in storage are not updated as new images arrive. When triggered, new tracks can be spawned and old tracks in storage reactivated. A trigger can be the number of active tracks dropping below a threshold and/or periodically after N frames are processed. The purpose of using a trigger is for computational efficiency alone.

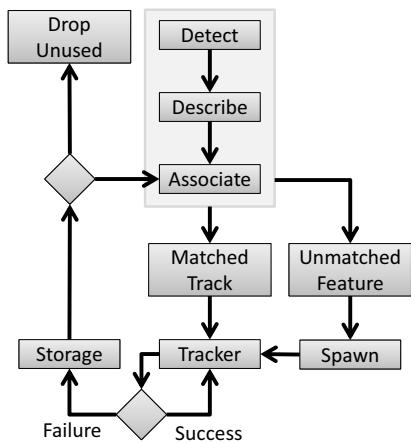


Fig. 1. Flow chart showing track life cycle. Track locations are updated using DT tracker. Tracks in storage are periodically reactivated from detected features.

The first step in spawning new tracks or reactivating tracks in storage is detecting image features. After image features have been detected they are associated against all active tracks and all tracks in storage. If an image feature is unassociated, then a new track is spawned from it. If a track in storage is associated with an image feature then it is reactivated by removing it from storage and setting its location to the feature and updating its DT description. A flow diagram of the track life cycle is shown in Figure 1. A greedy nearest-neighbor algorithm is used for association in this implementation.

3.1 Design Issues

When selecting algorithms to use inside the hybrid tracker, there are several issues that need to be considered. Can the DT tracker track features found by the detector? Can the DT tracker detect track faults?

Texture is used by DT trackers to track features. Unlike corner features, blob features contain all their texture information along the blob's outside edges, with little information inside. In theory, if a large blob was detected, the DT tracker's template could lack texture and fail, even with a pyramidal approach. In practice, this was found not to be an issue with real-world data due to sufficient texture across scales.

In the hybrid algorithm, a feature only switches to using the DDA approach when the DT tracker detects a fault. DT trackers typically employ several methods for detecting faults, but are susceptible to gradual drift. As will be shown below, it is possible for a slowly changing video sequence to produce worse performance than one with abrupt changes. This issue of drift can be mitigated in applications where geometric information is available. For example, when estimating the camera's ego-motion, features which drift are outliers and are dropped.

4 Experimental Setup

Performance is evaluated using video sequences for which a homography describes the relationship between each video frame. This approach is similar in spirit to the approach taken by Mikolajczyk et. al [18], where sequences of still images are provided along with corresponding homographies. Videos were collected using a consumer grade hand held point-and-shoot camera. Specifically, a Sony Cyber-Shot DSC-Hx5V camera capturing 640x480 MP4 video.

A homography provides a unique mapping between pixels in two image, $x' \propto H \cdot x$ where $H \in \mathbb{R}^{3 \times 3}$ is the homography and x is a homogeneous 2D pixel coordinate. A homographic relationship comes about when the scene is planar, or the camera's motion is purely rotational. Thus, the evaluated video sequences are either of planar objects, panoramic, or taken with a stationary camera.

An automated algorithm is used to reconstruct the "true" homography between the first image and each subsequent image. Procedure: 1) Remove lens distortion. 2) Track point features. 3) Estimate homography using points. 4) Non-linear refinement using inlier point set. 5) Non-linear refinement that minimizes average squared difference of pixel intensity.

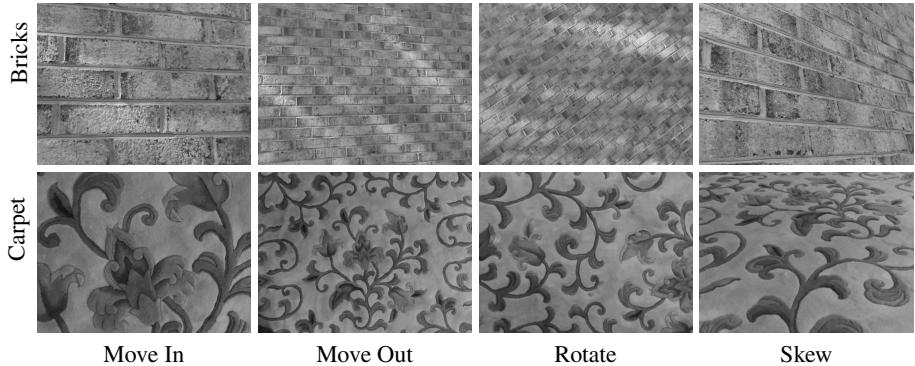


Fig. 2. Four different 6-DOF camera motions are evaluated for planar scenes. *Bricks* is more uniform in appearance and level of texture, while *carpet* has clearly defined objects with less interior texture.



Fig. 3. Changes in ambient light are evaluated in *illumination*. In *panoramic* the camera is approximately rotated about its focal point in an urban environment. For *compressed*, the bricks skewed sequence is highly compressed using JPEG. Compression artifacts are difficult to see in this figure due to the reduced image size.

Track stability is measured using F-measure Eq. 3, which is a function of precision and recall.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

Precision is defined as the number of true positive tracks divided by the total number of active tracks. Recall is defined as the number of true positives divided by the initial number of tracks. A true positive track is defined as a track that is within tolerance of the true feature location. The choice of tolerance for defining true positive tracks is somewhat arbitrary, and a value 5 pixels was selected. The same conclusions are reached when other tolerances are used.

Four scenes are used to evaluate performance (see Figures 2 and 3). Two scenes of planar objects with four different 6-DOF camera motions, one stationary camera view of a bookshelf with variable illumination, and one panoramic of an urban environment. The two 6-DOF scenes, *bricks* and *carpet*, are views of planar objects with different

textures intended to stress trackers differently. The effect of frame rate/object speed on tracker performance is examined by reprocessing a sequence with axial camera rotation at different frame rates.

Table 1. Tracker Summary

	Type	Detector	Descriptor
KLT	DT	Shi-Tomasi	5x5 Pyramid
FAST-BRIEF	DDA	FAST	BRIEF
FH-BRIEF	DDA	FH	BRIEF
FH-SURF	DDA	FH	SURF-64
FH-BRIEF-KLT	Hybrid	FH	Multiple
FH-SURF-KLT	Hybrid	FH	Multiple

Evaluated trackers are summarized in Table 1. Specific implementations of the hybrid tracker are named using a three word pattern, (DETECTOR) - (DESCRIPTOR) - (DT). DDA trackers are named using two words (DETECTOR) - (DESCRIPTOR). Each word signifies an internal algorithm. SURF is a popular state-of-the-art descriptor designed for speed and stability. BRIEF is a more recently proposed descriptor that uses binary encoding. Fast Hessian (FH) is the scale-space blob detector proposed with SURF. Both Shi-Tomasi and FAST are corner detectors, with the former using the image gradient and the latter using pixel intensity values. FAST is popular in applications with constrained computation resources due to its speed, but less stable than other detectors.

If provided, all descriptors, detectors, and trackers use recommended parameters from the original authors. Algorithms are tuned for stability across all scenarios. Runtime performance is evaluated on an Intel Quad Core 2 Q6600 at 2.4 GHz running Ubuntu Linux with kernel 2.6.35. Algorithms are provided by BoofCV, see project home page (<http://boofcv.org>) for a discussion on correctness and performance.

All trackers use the first frame as their key-frame and never spawn new tracks.

5 Results

Stability as an average across each sequence is shown in Figure 4. The complementary nature of KLT and the DDA trackers is clearly evident. During move out scenarios KLT excels while DDA trackers perform poorly and the reverse is true for rotation scenarios. Ability of feature detectors to estimate scale is limited by the camera's resolution, while KLT continuously updates each feature's description. KLT has more problems tracking sequences with heavy compression than DDA, likely caused by the lack of smoothness between frames due to compression artifacts. The FAST detector is not invariant to illumination or rotation, causing FAST-BRIEF to perform relatively poor for scenarios with changes in illumination and rotation.

Other behaviors are more evident when examining performance as a function of frame number, Figures 6 and 7. When the camera is rotating, DDA's stability is cyclical because of its ability to recover tracks, while KLT gradually decays. Hybrid trackers

Average Track Stability For Each Scenario

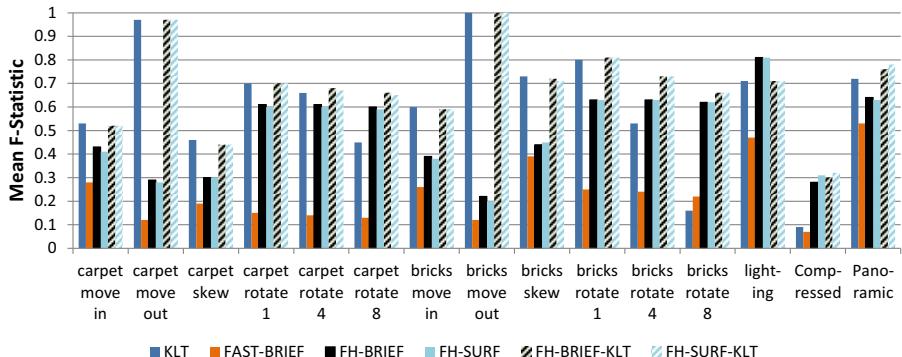


Fig. 4. Average F-statistic across each scenario, higher is better. Hybrid trackers are top performer in almost every scenario. Other trackers exhibit greater variability, with poor performance in one or more scenarios.

Runtime Performance

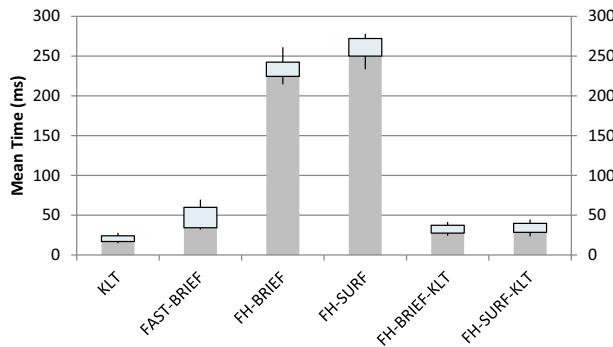


Fig. 5. Runtime performance box and whisker plot for all scenarios, lower is better. Y-axis is the average time to process a frame in milliseconds. Statistics shown are minimum, 25%, median, 75%, and maximum. Hybrid trackers are about 7.5 times faster than comparable DDA trackers and 1.5 times slower than KLT on average.

switching between the two techniques can also be seen. When tracking first begins it tends to behave like a DT tracker, but if tracks are dropped it recovers them by switching over to DDA.

The hybrid tracker's stability has a non-linear relationship with frame rate. This is illustrated by Figure 6 where stability can either improve or degrade as more frames are dropped. The hybrid tracker relies on KLT detecting its own faults, which can be unreliable at certain frame rates. Once a fault is detected, it switches to become more DDA tracker like and its performance recovers.

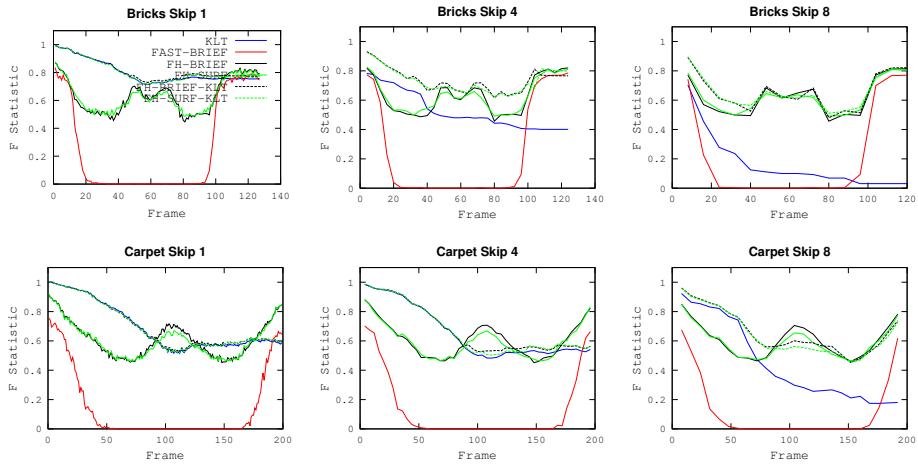


Fig. 6. Track stability by video frame for axial camera rotation when N frames are skipped, higher is better

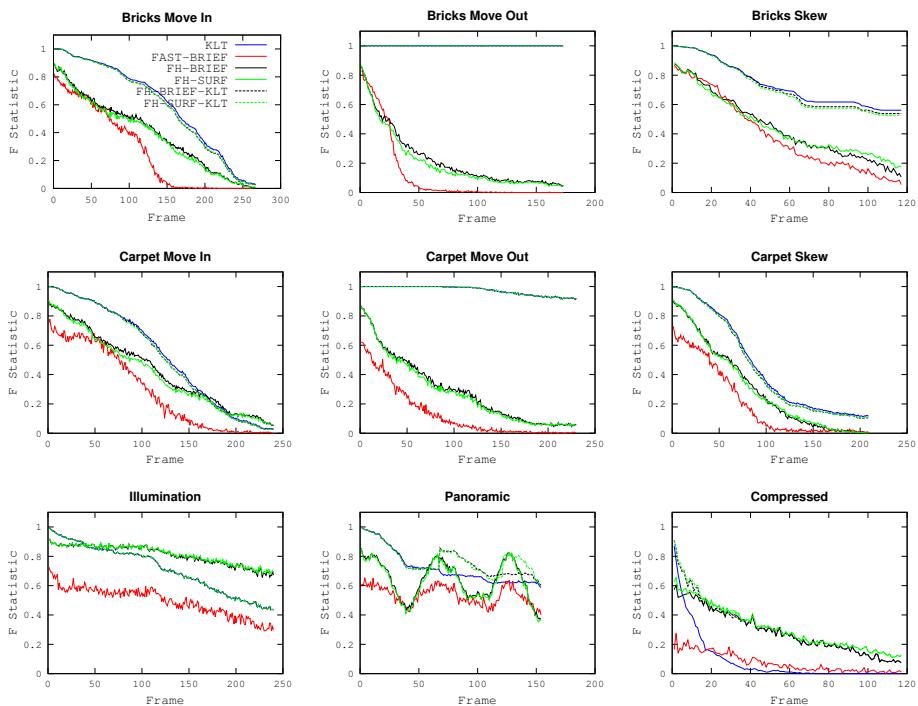


Fig. 7. Track stability by video frame for several scenarios, higher is better

Runtime performance across all scenarios is shown in Figure 5. KLT has the fastest performance followed by the two hybrid trackers. FAST-BRIEF is a close fourth place, but has poor stability. FH-BRIEF and FH-SURF are many times slower than other trackers.

6 Future Work

While a popular choice for evaluation in the literature, using homographic scenes to evaluate performance has significant disadvantages. The use of homographies is primary out of necessity, it provides a way to uniquely map pixels between two images. It does not capture the complexities found in arbitrary 3D motion. One such complexity involves objects overlapping each other and moving at different speeds. Evaluating trackers under arbitrary 3D motion requires complete knowledge of the scene's 3D structure and the camera's pose. Recently, data sets have been made available [19] that claim high accuracy vehicle pose and capture the scene's structure using LADAR. These could be used to evaluate tracking performance under a less restricted environment.

7 Conclusions

For the first time, track stability of DDA, DT, and hybrid trackers have been compared against each other for track stability and runtime performance. This comparison allows practitioners to make more informed decisions about the relative merits of each strategy in different scenarios. Hybrid trackers exhibit many of the advantages of DDA and DT tracking and to a lesser extent their disadvantages. The primary weakness arises from the need to detect faults in the DT tracker, which is unreliable in specific situations.

While hybrid trackers have received little attention in the literature, they were among the top performers across all scenarios and never experienced catastrophic failures. In some cases, they out performed both DDA and DT trackers. Performance of DDA and DT trackers fell along predictable lines and with each experiencing catastrophic failures. The results of this study suggest that a larger emphasis should be placed on the development of hybrid trackers.

References

1. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the 4th Alvey Vision Conference (1988)
2. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1994 (1994)
3. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision, vol. 1, pp. 430–443 (2006)
4. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22, 761–767 (2004)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints, cascade filtering approach. International Journal of Computer Vision (IJCV) 60, 91–110 (2004)

6. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. Computer Vision and Image Understanding (CVIU) 110, 356–359 (2008)
7. Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., Fua, P.: Brief: Computing a local binary descriptor very fast. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 1281–1298 (2012)
8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence (1981)
9. Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30, 79–116 (1998)
10. Silpa-Anan, C., Hartley, R.: Optimised kd-trees for fast image descriptor matching. In: Conference on Computer Vision and Pattern Recognition (2008)
11. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Trans. Pattern Anal. Mach. Intell. 20, 1025–1039 (1998)
12. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework: Part 1. Technical report, Robotics Institute (2002)
13. Baker, S., Matthews, I.: Equivalence and efficiency of image alignment algorithms. In: Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 1090–1097 (2001)
14. Uemura, H., Mikolajczyk, S.I.K.: Feature tracking and motion compensation for action recognition. In: British Machine Vision Conference, BMVC (2008)
15. Pilet, J., Saito, H.: Virtual augmenting hundreds of real pictures: An approach based on learning, retrieval, and tracking. In: IEEE Virtual Reality 2010 (2010)
16. Pradeep, V., Medioni, G., Weiland, J.: Visual loop closing using multi-resolution sift grids in metric-topological slam. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
17. Ladikos, A., Benhimane, S., Navab, N.: A realtime tracking system combining template-based and feature-based approaches. In: VISAPP 2007 (2007)
18. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1615–1630 (2005)
19. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition, CVPR (2012)

3D Shape Estimation Based on Sparsity in Stereo Matching

Naoto Hirose, Tatsuki Yasunobe, and Akira Kawanaka

Department of Information and Communication Sciences,
Faculty of Science and Technology,
Sophia University, 7-1, Kioicho, Chiyoda-ku Tokyo 102-8554, Japan

Abstract. Estimation error in stereo matching affects a generated 3D shape seriously. To correct for the disparity error caused by the estimation error, a disparity correction method that is based on the sparsity of the signal is proposed. First, disparity values for all pixels of the basis image are obtained by stereo matching using ASWA (Adaptive Support-Weight Approach). The focus is on each pixel, and the disparity values are replaced in a block that is centered on the pixel. The disparity values are wavelet-transformed and are calculated as the norm of the wavelet coefficients. The norm is added to an evaluation function for disparity estimation, and the updated disparity value is obtained. Through qualitative and quantitative evaluations, the proposed method is compared with the conventional method in this paper, and the results show that the proposed method is able to correct the disparity error.

1 Introduction

Recently the use of 3D models that can render images from every viewpoint has increased in various fields such as broadcasting, games, and virtual shopping. Three-dimensional geometry and texture data for stationary objects can be obtained by a 3D range finder, whose measurements can generate a precise model. However, the measurement has problems in that it is not able to measure dynamic objects and also requires large-scale equipment. A dynamic 3D model of a moving object can be generated from images taken from multiple viewpoints with multiple cameras. Various methods have been proposed to generate dynamic 3D models from a multi-viewpoint image. The Volume Intersection [1],[2] and stereo matching[3]-[6] methods are well known. The former can obtain robust 3D shapes with short processing time. But 3D generated models are limited to convex shapes; the concave portions of the object are not expressible because the shape is estimated from the silhouettes. Three-dimensional shape estimation using stereo matching can express a shape regardless of its shape properties.

The stereo-matching scheme determines the disparity of stereo images and estimates the distance to the target object surface from the viewpoint using the disparity. This method has a problem in that it tends to cause estimation errors in the portions of the images with less texture variation because it uses small blocks in the estimation of disparity. To solve the problem, ASWA (Adaptive Support-Weight Approach) [7],[8]

assigns higher weight to a pixel having similar intensity in the search window between the stereo images, but the disparity search is not accurate enough for estimating 3D shapes. Then, the proposed stereo matching is based on sparsity [9]-[11] from an idea that 3D objects have sparsity, and so disparity.

ASWA is described as a conventional stereo matching method in Section 2. The novel method of correcting disparity data based on sparsity is described in Section 3. The experiment that evaluates the performance of the proposed method was performed, and the results are shown in Section 4. The conclusions are described in Section 5.

2 Estimation of Disparity Using ASWA

ASWA which assigns higher weight to a pixel having similar intensity in the search window between the stereo images is often used as a method for locating corresponding points in stereo images, and the evaluation function is expressed as the following equation:

$$E = \frac{\sum w(p, q)w(\overline{p}_d, \overline{q}_d)e(q, \overline{q}_d)}{\sum w(p, q)w(\overline{p}_d, \overline{q}_d)}, \quad (1)$$

where p is an objective pixel position, q is the pixel position neighboring pixel p , and d is a disparity. \overline{p}_d is a pixel position in the reference image and is d pixels away from the objective pixel position. \overline{q}_d is the position of a pixel d pixels away from the pixel \overline{p}_d . $w(p, q)$ and $w(\overline{p}_d, \overline{q}_d)$ are weights for the pixels in the search windows in the basis image and the reference image, respectively. $e(q, \overline{q}_d)$ is the sum of the absolute differences in pixel intensity in the window of the basis image and the reference image. An example of the disparity image estimated based on Equation 1 is shown in Fig.1.



Fig. 1. Stereo image and disparity image. (a) Basis image, (b) Reference image, (c) Ground truth, (d) Disparity image by ASWA.

Fig. 1(a) and (b) are stereo images, (c) is ground truth, and (d) shows the disparity image obtained by ASWA. Fig. 1 (c) and (d) are expressed as gray scale images that show that as brightness approaches white, the disparity is larger. Also, the normalized maximum value for disparity as a function brightness is 255; and the minimum at a brightness of 0.

3 Estimation of Disparity Based on Sparsity

3.1 Sparsity of Signal

When a signal has only a few large amplitude components and the other components have values close to zero, the signal is said to be sparse. The idea of signal restoration based on sparsity is applied to the correction method for the disparity values obtained by stereo matching. This correction method is based on the assumption that the object surface often changes smoothly. Inpainting for the damaged image is well known as a method that utilizes sparsity well [12],[13]. An example is shown in Fig. 2.

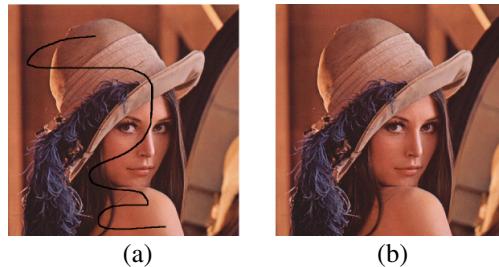


Fig. 2. An example of the inpainting method. (a) Input image, (b) Output image.

The inpainting method corrects a 2D image which lacks pixel values such as moth-eaten holes or scribbles. The processing of inpainting is performed by the following optimization with threshold processing:

$$X_i^{(k+1)} = WDen_\lambda \left(X_i^{(k)} + M_i \left(X_i^{(0)} - X_i^{(k)} \right) \right), \quad (2)$$

where k is the number of repetitions, $X_i^{(k)}$ is a vector expressing the pixel value after k updates. The input image is substituted for $X_i^{(0)}$ as the initial value. M_i is a mask that indicates the portion to be corrected. $WDen_\lambda$ is a dictionary composed of the basis of the frequency transform and performs thresholding with threshold λ . Here, multiple resolution decomposition with wavelet transform is used as the dictionary. The inpainting method performs thresholding for the wavelet coefficients after multiple resolution decomposition and reconstructs image based on the sparsity signal.

3.2 Introducing Sparsity to Disparity Estimation

Some hard estimation errors of disparity were found at positions that should take similar disparity value as their neighbors in Fig. 1(d). We intend to correct such estimation error based on the sparsity of disparity. First, frequency decomposition of disparities obtained by stereo matching is performed. Then, the norm of the wavelet

coefficients are calculated, and the norm is added to the evaluation function of ASWA. The evaluations are expressed in the following equations:

$$H_0 = \frac{\sum w(p, q)w(\overline{p_d}, \overline{q_d})e(q, \overline{q_d})}{\sum w(p, q)w(p_d, q_d)} + \lambda |\mathbf{c}|_0 \quad (3)$$

$|\mathbf{c}|_0$: the number of coefficients satisfying $|c_{ij}| > T$

$$H_1 = \frac{\sum w(p, q)w(\overline{p_d}, \overline{q_d})e(q, \overline{q_d})}{\sum w(p, q)w(p_d, q_d)} + \lambda |\mathbf{c}|_1 \quad (4)$$

$$|\mathbf{c}|_1 = \sum_i \sum_j |c_{ij}|,$$

where c_{ij} are the wavelet coefficients obtained by wavelet transform for the disparity in the small block with which disparity is estimated, $|\mathbf{c}|_0$ is the 0th order norm which is the number of the wavelet coefficients beyond a threshold T , and $|\mathbf{c}|_1$ is the first order norm which is the sum of absolute value of wavelet coefficients without a DC component. λ is a control parameter.

In both Equations (3) and (4), the first term is the evaluation function of ASWA, and the second term evaluates the sparsity. When the evaluation function is minimized, the disparity value is replaced by a new value. The flow of procedure is summarized as follows:

- Step 1: Estimate the disparity distribution between two images using stereo matching with the evaluation function of ASWA.
- Step 2: Focus on a top-left point in the disparity distribution and pick a block whose size is same as the block size $m \times m$ of stereo matching with that point as its center.
- Step 3: Moreover, the disparity of the objective pixel position is replaced with a top-left point in a small region whose size is denoted by $s \times s$ with the objective pixel position as its center. Call this region the *disparity trade candidate region*. When the disparity value is replaced, replace not only the central disparity value but also its neighbors with the same value. Call this region the *disparity uniformization region* which is denoted by $t \times t$.
- Step 4: Perform multiple resolution decomposition using a wavelet transform for the disparity in the block $m \times m$ of step2.
- Step 5: Calculate the norm $|\mathbf{c}|$ of the wavelet coefficients and then evaluation function H . Where, the 0th or first order norm is used .

Step 6: Steps 3-5 are repeated for all disparity value in the *disparity trade candidates*.
 Step 7: Disparity d that minimizes the evaluation function becomes the new disparity.
 Step 8: Steps 2-7 are repeated for all positions in the disparity distribution.

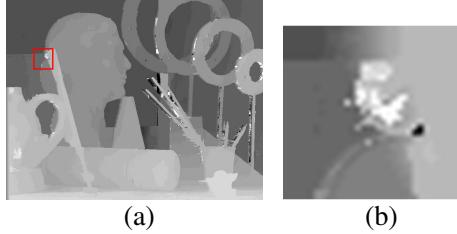


Fig. 3. An example of choosing a point in the disparity distribution in Step 2. (a) Disparity distribution which is obtained by ASWA, (b) Enlarged region for which the wavelet coefficients are calculated.

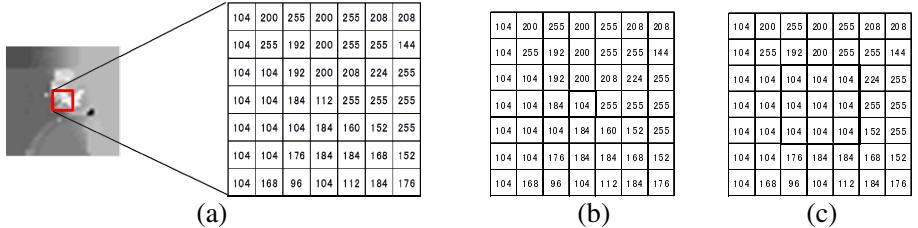


Fig. 4. An example of uniformization for the disparity distribution in Step 3. (a) *Disparity trade candidate region* for $s=7$, (b) *Disparity uniformization region* for $t=1$, (c) *Disparity uniformization region* for $t=3$.

4 Experiments

A comparison of the 0th and first order norms for the stereo images in Fig. 1 was performed. Then a comparison the proposed method and the conventional method was performed. The size of input image “art” is 695×555 , “laundry” and “reindeer” is 671×555 . The size of block for stereo matching is 35×35 . The decomposition level of the wavelet transform is 3, the size of disparity trade candidate region is 7×7 . When the 0th order norm is calculated, the control parameter is 2.0. When the first order norm is calculated, the control parameter is 0.03. To compare with the proposed method, the disparity corrected using inpainting method is shown. The correcting portion of inpainting method M_i is determined using median filter whose size is 7×7 .

$$M_i = \begin{cases} 0 & |D_i - MedianD| < 1 \\ 1 & |D_i - MedianD| \geq 1 \end{cases}, \quad (5)$$

where D_i are disparity values estimated by using ASWA and $MedianD_i$ are disparity values which is median-filter outputs for D_i .

Peak Signal to Noise Ratio which is a function for evaluating the degree of distortion compared to the ground truth, was used as an objective measure.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} [dB], \quad (6)$$

where MSE indicates Mean Square Error between the disparity distribution and the ground truth.



Fig. 5. Input images. (a) Left image of “laundry”, (b) Right image of “laundry”, (c) Left image of “reindeer”, (d) Right image of “reindeer”.

4.1 Determining the Optimal Parameter in the Method Based on the 0th Order Norm

Table 1 shows the results due to the size of *disparity uniformization region* in the method based on the 0th order norm. When the *disparity uniformization region* is 3×3 , a white mass is found in part of the ring in the top center of the disparity image in Fig. 6 (a). When the *disparity uniformization regions* are 5×5 and 7×7 , the disparity values between the ring and the forehead of the stone statue have similar values, but the white mass is corrected. In the objective evaluation shown in Table 1, the PSNR of 5×5 is 0.11dB higher than 7×7 . Therefore, the size of the *disparity uniformization region* of 5×5 was used for the value in the method based on the 0th order norm.

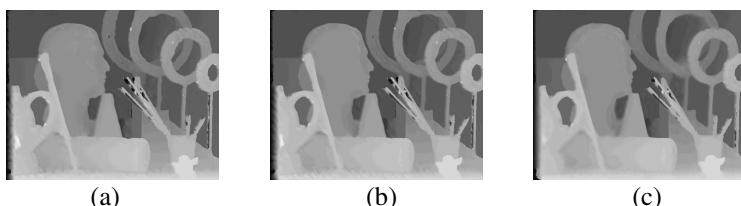


Fig. 6. Performance evaluation due to the size of *disparity uniformization region* in the 0th order norm. (a) $t=3$,(b) $t=5$,(c) $t=7$.

Table 1. PSNR of the disparity obtained by the method based on 0th order norm

<i>disparity uniformization region [pixels]</i>	PSNR[dB]
3×3	24.75
5×5	25.01
7×7	24.90

Table 2. PSNR of the disparity obtained by the method based on first order norm

<i>disparity uniformization region [pixels]</i>	PSNR[dB]
3×3	25.02
5×5	25.10
7×7	25.15

4.2 Determining the Optimal Parameter in the Method Based on the First Order Norm

The result of changing the size of *disparity uniformization region* in the method based on the first order norm is shown in Table 2. When the disparity uniformization region is 3×3, a white mass is found in part of the ring near the top center of the disparity image in Fig 7 (a). When the size of *disparity uniformization region* is 5×5 or 7×7, the white mass remains partially, but the expansion of the outline of the stone statue is suppressed. In the objective evaluation shown in Table 2, the PSNR of 7×7 is 0.05dB higher than that of 5×5. Therefore, the size of *disparity uniformization region* is set to 7×7 in the method based on the first order norm.

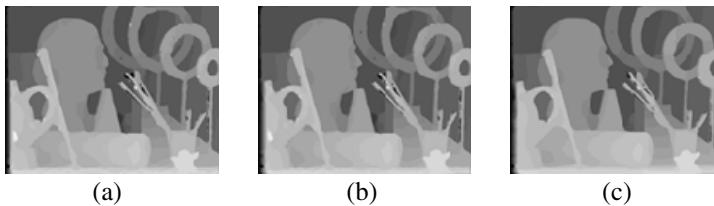
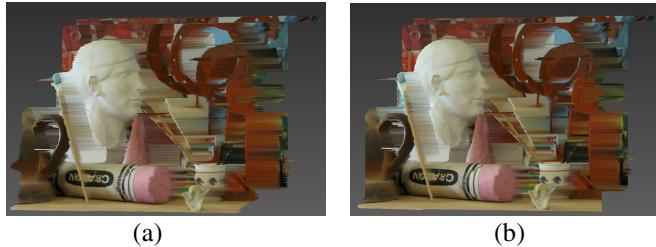


Fig. 7. Performance evaluation due to the size of *disparity uniformization region* in the first order norm. (a) $t=3$, (b) $t=5$, (c) $t=7$.

4.3 Performance Comparison between Two Norms

The 3D shapes generated by the method based on the 0th order norm and the first order norm using optimal parameters, which have been obtained in experiments 4.1 and 4.2, are shown in Fig. 8. The distortion of the shapes arises in some spots in the method based on the 0th order norm, but there is little distortion of shape in the method based on the first order norm. From PSNRs shown in Table 1 and 2, the method based on the first order norm is shown to be better than the method based on the 0th order norm.



(a)

(b)

Fig. 8. 3D shapes using optimal parameters. (a)The 3D shape generated by the disparity based on the 0th order norm with the *disparity uniformization region* 5×5 ,(b)The 3D shape generated by the disparity based on the first order norm with the *disparity uniformization region* 7×7 .

4.4 Comparison with Conventional Method

The disparity images obtained by ASWA, inpainting method and the proposed method with first norm are shown in Fig. 9 and 10 (a) - (c), the 3D shape generated by disparity value obtained by each method are shown in Fig.9 and 10 (e) - (g). (d) and (h) shown the Ground truth.

In Fig. 9 and 10, the disparity image and 3D shape obtained by ASWA are corrected by the proposed method. From the objective evaluation, when the proposed method is used, the disparity image corrects closer to the ground truth because PSNR becomes higher for all stereo images. The proposed method is shown to correct the disparity obtained by the conventional method in these evaluation experiments.

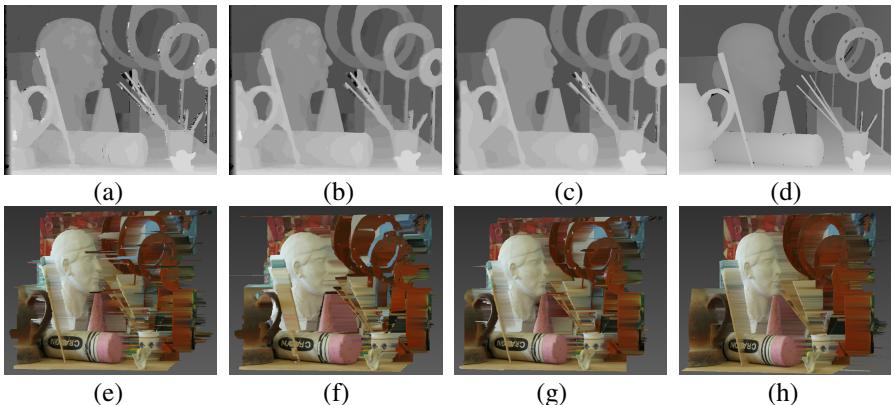


Fig. 9. The visual results of “art” obtained by different methods (a) Disparity image obtained by ASWA, (b) Disparity image corrected by inpainting, (c) Disparity image obtained by the method based on the first order norm, (d) Ground truth, (e) The 3D shape generated by the disparity obtained by ASWA, (f) The 3D shape generated by the disparity corrected by inpainting, (g) The 3D shape generated by the disparity obtained by the method based on the first order norm, (h) Ground truth.

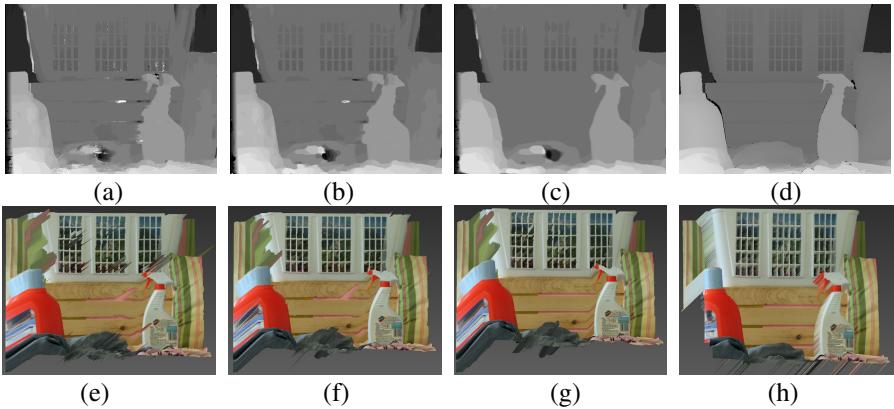


Fig. 10. The visual results of “laundry” obtained by different methods. (a) - (h) are the same as Fig.9.

Table 3. Comparison of PSNR[dB] obtained by different methods

The kind of method	art	laundry	reindeer
ASWA[7][8]	20.77	21.10	22.19
Inpainting[12][13]	21.64	21.77	22.70
Proposed	25.15	25.13	23.15

5 Conclusion

In this paper, a novel method for correcting the disparity obtained by stereo matching based on the sparsity of signal is proposed. Based on the results of some experiments, the proposed method is shown to be able to correct the distorted shape from the conventional stereo-matching algorithm to a robust 3D shape.

In addition, the comparison between the method based on the 0th order norm and the first order norm, found the latter is more suitable for the disparity correction method.

Although a sparsity of the wavelet coefficients of the disparity distribution was introduced, the depth distribution is still important for 3D shape expression. Therefore, we will utilize the wavelet coefficients of depth distribution which are the reciprocals of disparity for the evaluation of depth distribution sparsity.

References

- [1] Schreer, O., Kauff, P., Sikora, T.: 3D video communicaion, pp. 135–136. John Wiley & Sons, Ltd. (2005)
- [2] Tomiyama, K., Katayama, M., Iwadate, Y., Imaizumi, H.: A Dynamic 3D, Object-Generating Algorithm from Multiple Viewpoints Using the Volume Intersection and Stereo Matching Methods. Image Information and Television Engineers 58(6), 797–806 (2004)

- [3] Rander, P.: A Multi-Camera Method for 3D Digitization of Dynamic Real-World Events, CMU-RI-TR-98-12 (March 1998)
- [4] Min, D., Sohn, K.: Cost aggregation and occlusion handling with wls in stereo matching. TIP 17(8), 1431–1442 (2008)
- [5] Gerrits, M., Bekaert, P.: Local stereo matching with segmentation-based outlier rejection. In: CRV (2006)
- [6] Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. PAMI 30(2), 328–341 (2008)
- [7] Yoon, K.-J., Kweon, I.S.: Adaptive Support Weight Approach for Correspondence Search. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 650–656 (2006)
- [8] Yoon, K.-J., Kweon, I.-S.: Locally Adaptive Support-Weight Approach for Visual Correspondence Search. Robotics and Computer Vision Laboratory, Dept. of Electrical Engineering and Computer Science, KAIST, 373-1 Gus-eong-dong Yuseong-gu Daejeon, 305-701, Korea (2005)
- [9] Starck, J.-L., Fadili, M.J.: An Overview of Inverse Problem Regularization Using Sparsity. In: ICIP, pp. 1453–1456 (2009)
- [10] Martin, A., Fuchs, J.-J., Guillemot, C., Thoreau, D.: Sparse Representation for Image Prediction. In: EUSIPCO, pp. 1255–1259 (2007)
- [11] Sarkis, M., Diepold, K.: Towards real-time stereo using non-uniform image sampling and sparse dynamic programming. In: International Symposium on 3D Data Processing, Visualization and Transmission (June 2008)
- [12] Fadili, M.J., Starck, J.-L.: EM Algorithm for Sparse Representation-Based on Image In-painting. In: ICIP, pp. II61–II64 (2005)
- [13] Ghoniem, M., Chahir, Y., Elmoataz, A.: Geometric and texture inpainting based on discrete regularization on graphs. In: ICIP, pp. 1349–1352 (2009)

Color Image Compression by Riemannian B-Tree Triangular Coding

Olfa Triki^{1,2} and Mourad Zéraï^{2,3}

¹ GRIFT, ENSI, Manouba University, Tunisia

² Ecole Supérieure Privée d'Ingénierie et de Technologies, Tunisia

³ LAMSIN-ENIT, Tunis Al Manar University, Tunisia

Abstract. We propose an enhancement of the B-Tree-Triangulation coding technique in a Riemannian framework . Our method focuses on the subdivision criteria in the triangulation process, and compares results obtained by means of two different metrics: the Euclidean and the Riemannian one. Comparison between input and compressed/decompressed images, in terms of a set of image quality measures, shows better performance of Riemannian triangulation in low intensity levels, while using Euclidean distance works better in high intensities. We therefore propose two combined decision criteria and compare image quality measures. Results show an enhancement of image quality at similar compression rates, and are promising for further adjustments to specific applications.

1 Introduction

The Binary Tree Triangular Coding (BTTC) scheme, introduced by [1] in 1997, is a well-known compression technique that proved to be more efficient than spectral-based methods. Many authors tried to enhance this method [2–4] and their work shows that the weaknesses of BTTC are the subdivision criterion in coding and linear interpolation for decoding. In this paper, we address this issue. However, rather than proposing another algorithm, we show that using the same method, but considering it in a Riemannian context of color geometry instead of the classic Euclidian one, can enhance BTTC quality performances for the same compression rates.

The article is organized as follows: First we draw a state of the art and recall BTTC-triangulation algorithm in section 2. We then introduce in section 3 the Riemannian color geometry, upon which is based our method. Section 4 presents the proposed Riemannian BTTC (R-BTTC) algorithms, and compares their results in terms of a set of image quality measures.

2 Image Compression: State of the Art

Image compression aim is to reduce the amount of data needed to represent the same image. We distinguish mainly unlossy and lossy compression, where the information can be entirely recovered or not, respectively. Lossy compression

techniques, which are the most used, can be classified into two families: spectral-based coding [5–7] and geometry-based coding [1, 2, 8, 9]. Spectral-based techniques rewrite the image by projecting it onto a basis. The compressed data is then simply projection coefficients. In geometry-based techniques, the approach is generally to use some geometrical segmentation of the image, and then to reduce the data to uniform segmented regions. Interpolation can afterwards be used in order to get a smooth result rather than clear pattern contours.

BTTC [1] is proved to perform better than spectral-based methods, but has a higher complexity. Being interested mainly in compression performances regardless of computation time, we focus here on BTTC-triangulation, and more precisely on the subdivision process, in order to enhance its compression rate for the same image quality. Let us first give a brief view of the BTTC coding/decoding scheme.

2.1 B-Tree Triangular Coding (BTTC)

The Binary Tree Triangular Coding (BTTC) compression method is based on the iterative construction of a triangular mesh which decomposes the image into triangles where pixel gray-scales are similar [1]. Once a maximum error rate is fulfilled, the subdivision stops and the triangles have zero children.

The image is then coded as a binary tree stored in a string s representing leaf and intermediate nodes. Finally, the decoded image is obtained by filling the reconstructed triangles. This filling can be performed in different manners, but linear interpolation proved to give acceptable image quality with a convenient computational cost, which made BTTC faster than the standard techniques based on transforms [1].

The algorithm of BTTC presents a weakness in the subdivision process, where the error criterion is relatively simplistic. In fact, the algorithm presented by *Distasi & al.* fixes a maximum tolerated error ϵ inside a triangle. Subdivision is iteratively performed until this condition is fulfilled. In order to achieve this error minimization, a threshold is therefore applied on the maximum distance between gray-levels (or colors) inside each triangle to decide whether to subdivide it or not.

This error measure in BTTC is calculated as an Euclidean distance d_E , defined by:

$$d_E = \left[((maxR - minR)^2 + (maxG - minG)^2 + (maxB - minB))^2 \right]^{\frac{1}{2}}, \quad (1)$$

where $max = (maxR, maxG, maxB)$ and $min = (minR, minG, minB)$ are respectively the maximum and minimum values of color in the considered triangle.

In [2], the authors study the effect of using some different error criteria: average difference, entropy, mean square error, and fuzzy compactness. As predicted, using more informative (average instead of maximum) quality measures enhances the compression quality. Results show a better performance when using fuzzy compactness, followed by average difference. However, these results also mean a certain computational cost, due to the alghorithm complexity added by the

calculus of such criteria. In [4], the authors address both the triangular mesh construction and the interpolation steps. They modified B-tree structures by S-tree subdivision, and enhanced image quality after decompression by means of the gouraud shading model instead of linear interpolation. In [3], anisotropic diffusion is used to get again a better interpolated image. As we can see, although linear interpolation made BTTC faster than usual compression techniques, it still needs to be enhanced.

In this paper, we propose to use a different distance metric, which should lead to a better triangulation. Moreover, we propose to perform BTTC within the Riemannian color framework, allowing us not only to get a better triangulation for the same compression rate, but also to enhance interpolation results. In fact, previous work [10, 11] on color perception and metrics proved that the color space is not flat, i.e. Euclidian, it is rather Riemannian.

3 Riemannian Color Framework

We limit this section to distance definitions under two well-known Riemannian metrics: Helmholtz and Stile's ones. Fundamental properties and notations for Riemannian manifolds, that will be applied in this sections, can be encountered, in [12]. Besides, Theoretical background can be found in [13], where distance definition is explained in detail.

The Helmholtz distance [14] from the color \mathbf{x} to the color \mathbf{y} , is given by:

$$d_H(\mathbf{x}, \mathbf{y}) = \left\{ \sum_{i=1}^3 [\log(y_i) - \log(x_i)]^2 \right\}^{\frac{1}{2}}, \quad (2)$$

where the subscript H in d_H stands for "with respect to the Helmholtz metric".

Walter W. Stiles modified the Helmholtz's proposal in order to better account for observations of threshold values (see [15] p. 660). The Stiles distance from the color \mathbf{x} to the color \mathbf{y} , is then given by:

$$d_S(\mathbf{x}, \mathbf{y}) = \left\{ \left[\frac{1}{\rho} \log \left(\frac{1+9y_1}{1+9x_1} \right) \right]^2 + \left[\frac{1}{\gamma} \log \left(\frac{1+9y_2}{1+9x_2} \right) \right]^2 + \left[\frac{1}{\beta} \log \left(\frac{1+9y_3}{1+9x_3} \right) \right]^2 \right\}^{\frac{1}{2}}, \quad (3)$$

where the subscript S stands for "with respect to the Stile's metric". Here, ρ , γ and β are constants fixed in a heuristic way, in order to get the closest metric to the human perception.

3.1 Decoding with Riemannian Interpolation

In order to reconstruct the compressed image, BTTC scheme procedes by filling each triangle. This filling can be performed in different manners, but linear interpolation proved to give acceptable image quality with a convenient computational cost. Let us recall that in the framework of Euclidean geometry, the interpolation formula is given by:

$$c = \alpha * c_1 + \beta * c_2 + \gamma * c_3. \quad (4)$$

**Fig. 1.** Test images

In order to be compliant with the human vision system and the Riemannian color geometry, we propose here a Riemannian interpolation, where the colour c in a point inside a triangle $\langle c_1, c_2, c_3 \rangle$ is given by:

$$c = c_1^\alpha * c_2^\beta * c_3^\gamma, \quad (5)$$

where α, β and γ are barycentric coordinates of point P inside the triangle.

Notice that Riemannian interpolation has the advantage of resulting in colour values within the range [0, 255], while Euclidean formula can give values bigger than 255, and needs a truncation, which means a loss of information.

4 Riemannian-BTTC

Applying Riemannian color geometry to the BTTC method will infer a change in two steps of the algorithm:

- The subdivision criterion, based on a measure of the maximum deviation inside a triangle, will be calculated according to one of the known Riemannian distances $d_H(\max, \min)$ or $d_S(\max, \min)$, instead of Euclidean distance d_E .
- During decoding, the interpolation performed to fill missing data in the triangles coded will be calculated according to the Riemannian color interpolation formula.

4.1 Helmholtz and Stile's BTTC

In order to estimate the effect of changing triangulation procedure on the coding/decoding scheme, we performed BTTC and R-BTTC on a set of images. (some examples are shown Figure 1), and compared the coded/decoded images and the triangulations obtained.

In the case of Riemannian deviation criterion, introducing the *log* enhances the error for dark intensities. As a consequence, the triangulation process makes more emphasis this time on low intensity regions. In fact, using an Euclidian measure of distance leads to small distances in low gray levels, especially in

colored images where each channel difference is squared. Since the Human perception can distinguish such small differences, this kind of color distance measure is therefore not well suited to image processing. On the other hand, the Euclidean distance defines the minimum and maximum values in a separate manner for each color channel. The measured distance does not reflect a color distance between two pixels, and therefore is not a measure of the observed color deviation.

Figure 2 shows results obtained for image 'wheel', with Euclidean and Riemannian BTTC schemes for the same compression ratio, ie the same number of triangles. Note that, for a mesh of 10000 triangles, Euclidean-BTTC concentrates more on high intensity regions, while R-BTTC behaves the opposite way. Obviously, for low compression rates, both methods give similar triangulations.

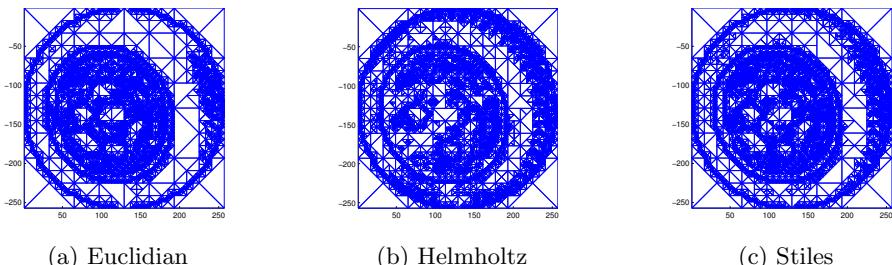


Fig. 2. BTTC results for Wheel (10000 triangles) with different color distance metrics

Figure 3 shows another example of triangulations obtained with Euclidean, Helmholtz and Stile's distances, for the Rubens painting of King Ahmed. Figure 3.b in particular, shows that using Riemannian distance in the triangulation process significantly increases the SNR for low intensities. Experimental results performed suggest that this result is generally true for intensities ranging from 0 to 120 approximately. For higher gray-levels, Euclidean distance performs better in terms of SNR. Starting from this observation, we propose a combined criteria subdivision scheme.

4.2 ER-BTTC: A Combined Euclidian-Riemannian BTTC

Here, at a given leaf of the subdivision tree, in order to decide whether to subdivide or not, we first perform a test: if the mean intensity value of the considered triangle is in [0, 128], we threshold according to the Riemannian Helmholtz distance, else we use Euclidean distance. Figure 3.b shows already an enhancement of SNR per gray level, in the case of King Ahmed painting.

A battery of tests is further performed, on another image(palm-tree), with bad lighting conditions. Figure 4 shows un example of coded/decoded image with Euclidian and Riemannian frameworks. Figure 5 shows a comparison of results obtained with Euclidian, Helmholtz, Stiles and combined schemes, in terms of a set of well-known Image Quality Measures(IQMs) [13, 16, 17] (see

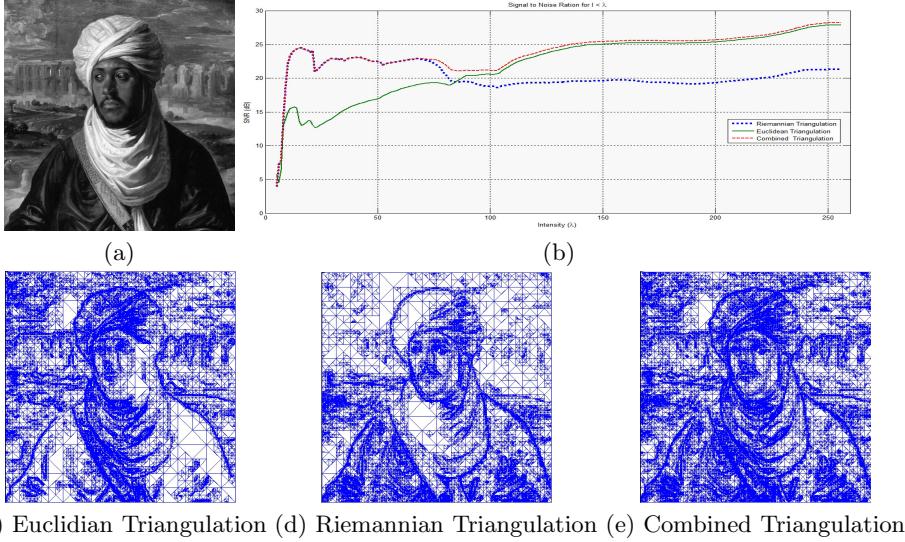


Fig. 3. E-BTTC and R-BTTC results for Rubens painting of King Ahmed

Table 1. Some Euclidean and Riemannian quality measure formulæ[13]

Evaluation Technique	Euclidean Form	Riemannian Form
Average Difference	$\frac{\sum_{i,j}^{M,N} [F(i,j) - \hat{F}(i,j)]}{MN}$	$\frac{\sum_{i,j}^{M,N} d_R(F(i,j), \hat{F}(i,j))}{MN}$
Structural Content	$\frac{\sum_{i,j}^{M,N} [F(i,j)]^2}{[\hat{F}(i,j)]^2}$	$\frac{\sum_{i,j}^{M,N} d_R(F(i,j), \mathbf{1})^2}{d_R(\hat{F}(i,j), \mathbf{1})^2}$
N. Mean Squared Error	$\frac{\sum_{i,j}^{M,N} [F(i,j) - \hat{F}(i,j)]^2}{[F(i,j)]^2}$	$\frac{\sum_{i,j}^{M,N} d_R(F(i,j), \hat{F}(i,j))^2}{d_R(F(i,j), \mathbf{1})^2}$
Maximum Difference	$\max_{i,j} (\ F(i,j) - \hat{F}(i,j)\)$	$\max_{i,j} d_R(F(i,j), \hat{F}(i,j))$

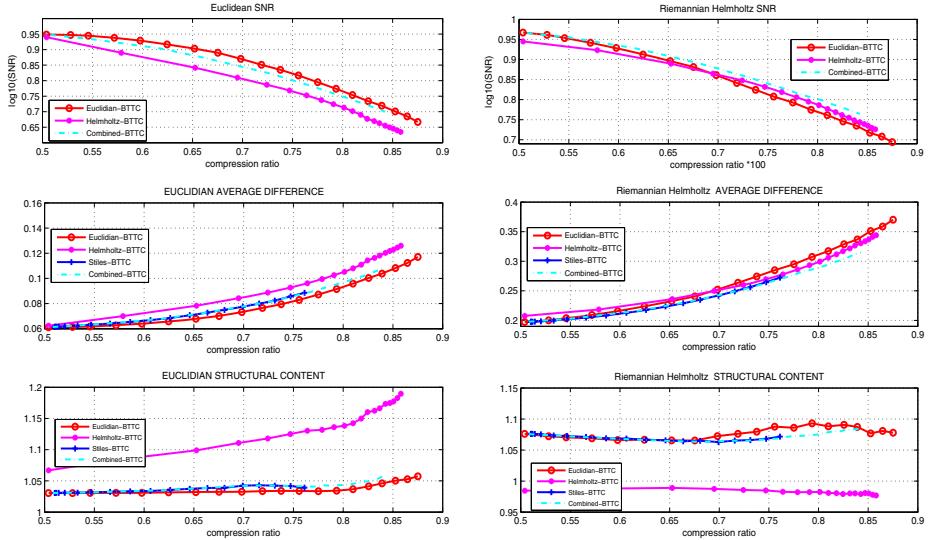
Table 1). At first sight, classic global IQMs such as SNR and Average Difference (AD) vote for the corresponding metric, i.e. Euclidian IQM shows better performances for Euclidian BTTC, and vice-versa. A deeper insight is given by two kinds of IQM indicators: (i) Structural Content (SC) shows, even with the Euclidian SC, a much better performance of the Helmholtz scheme, (ii) IQMs evaluated on gradients of original and compressed images all vote for the combined scheme, or show a very similar performance of combined and stiles BTTCs (Figure 6).

The observed quality measures are completely coherent with the theoretical background: the human eye being more sensitive to contours, performing IQMs on the gradient image therefore concentrates on the raising structures in the image, rather than summing small differences over the hole image, which is the case for global indicators such as the SNR. We conclude therfore that the proposed combined-BTTC is a good alternative that enhances BTTC compression



(a) E-BTTC

(b) R-BTTC

Fig. 4. Euclidian and Riemannian BTTC decompressed results for image 'palm tree'**Fig. 5.** IQMs for image 'palmier'

ratios for the same image quality. However, we would like to define a mathematically coherent scheme, entirely within the Riemannian framework, instead of a simple threshold on gray levels. In order to achieve this goal, we define Upper and Lower Helmholtz metrics, and settle a new combined BTTC algorithm: Bilateral-Helmholtz-BTTC. Note that, with these new definitions, Lower Hwlmholtz is simply the formerly defined Helmholtz metric.

4.3 Bilateral Helmholtz BTTC: BH-BTTC

For a more general use of BTTC, we introduce a combined metric, trying to get the best image quality for the same compression ratio. This goal should be obtained when triangle distribution is less emphasising on a particular gray level range. Instead, we need a symmetric behaviour in dark and bright colors. In order

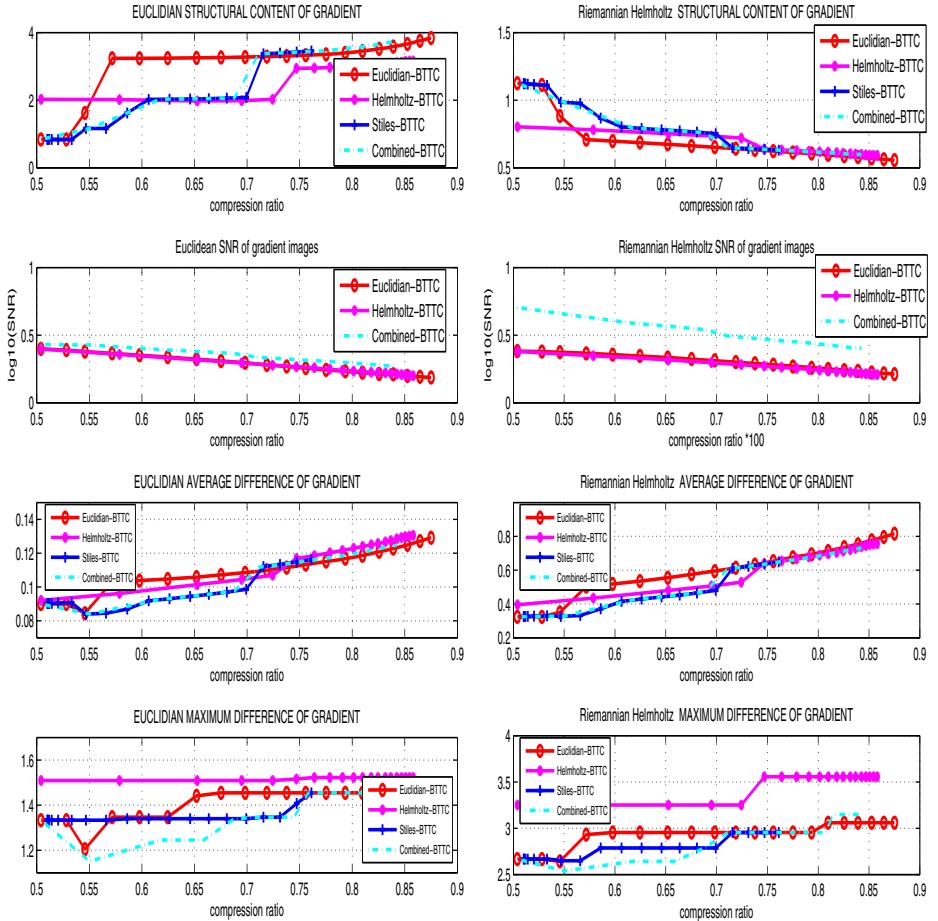


Fig. 6. IQMs calculated on gradient images for image 'palmier'

to get a symmetric behaviour of BTTC in upper and lower intensity levels, we hence define an Upper Hwelmholtz Distance d_{UH} , given by:

$$d_{UH}(p, q) = \left\{ \sum_{i=1}^3 \left[\log(1 - p^i) - \log(1 - q^i) \right]^2 \right\}^{\frac{1}{2}},$$

where the emphasis is on diffrences in high intensities.

The proposed Bilateral Helmholtz Distance d_{BH} is obtained by combining Upper and Lower Helmholtz metrics within the same distance measure, given by:

$$d_{BH}(p, q) = \left\{ \sum_{i=1}^3 \left[\log\left(\frac{1 - p^i}{p^i}\right) - \log\left(\frac{1 - q^i}{q^i}\right) \right]^2 \right\}^{\frac{1}{2}}.$$

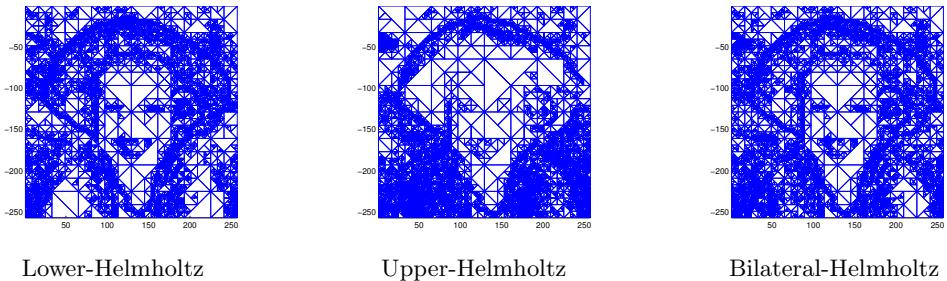


Fig. 7. BTTC triangulation results for image *Child*

Figure 7 shows some results obtained for image *child*, with Euclidian, Lower, Upper and Bilateral Helmholtz BTTC, where the triangle subdivision criterion is performed according to the corresponding distance measures. We can easily notice a better distribution of triangles over image details with the Bilateral Helmholtz BTTC. Note for example cloth, hat and face details.

4.4 Adaptive Helmholtz BTTC: α -BTTC

Finally, we propose an adaptative BTTC triangulation scheme, in order to adapt triangulation error criterion to the input image. We introduce a weighted Helmholtz distance, where the weight α can be manually set or automatically calculated from image histogram properties. Distance definition becomes:

$$d_{BH\alpha}(p, q) = \left\{ \sum_{i=1}^3 \left[\alpha * \log \left(\frac{p^i}{q^i} \right) - (1 - \alpha) * \log \left(\frac{1 - p^i}{1 - q^i} \right) \right]^2 \right\}^{\frac{1}{2}}. \quad (6)$$

5 Conclusion

We proposed in this paper a Riemannian-BTTC framework, where both subdivision criteria for triangulation and color interpolation for decoding are performed under the Riemannian color geometry framework. Experimental results show an interesting improvement of the SNR in coded/decoded images within the Riemannian framework, for a similar number of mesh triangles. This means getting a better quality of the coding/decoding process without requiring more transmission pass-band. Note that these distance measures can be integrated within other improvements of the BTTC algorithm, such as using image quality measures like fuzzy compactness in [2] in order to get even better results.

In order to adapt the algorithm performances to specific application needs, a set of Riemannian BTTC schemes is proposed in this paper: Helmholtz-BTTC, Stiles-BTTC, combined RE-BTTC, and finally Bilateral Riemannian BTTC.

Although the combined BTTC seems to be a good solution for general purpose images, we also propose specific tuning of the range of colors in bilateral R-BTTC in order to adapt triangulation to each image color content. Finally, these results could be confirmed in many other applications such as medical image segmentation, where small intesity differences in low intensity levels occur and are relatd to important tissue density information.

References

1. Distasi, R., Nappi, M., Vitulano, S.: Image compression by b-tree triangular coding. *IEEE Transactions on Communications*, 1095–1100 (1997)
2. Prasad, M.V.N.K., Mishra, V.N., Shukla, K.K.: Space partitioning based image compression using quality measures for subdivision decision. *Appl. Soft Comput.* 3, 273–282 (2003)
3. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. *Journal of Mathematical Imaging and Vision* 31, 255–269 (2008)
4. Chung, K.L., Wu, J.G.: Improved image compression using s-tree and shading approach. *IEEE Transactions on Communications* 48, 748–751 (2000)
5. Zemcik, P., Vorcek, J., Frydrych, M., Klviinen, H., Toivanen, P.: Multispectral image color encoding. In: International Conference on Pattern Recognition, vol. 3, p. 3609 (2000)
6. Santa-Cruz, D., Ebrahimi, T.: A study of jpeg 2000 still image coding versus other standards. In: Proc. of the X European Signal Processing Conference, vol. 2, pp. 673–676 (2000)
7. Sudhakar, R., Karthiga, R., Jayaraman, S.: Image compression using coding of wavelet coefficients: A survey. In: CVIP 2005, pp. 25–38 (2005)
8. Demaret, L., Dyn, N., Iske, A.: Image compression by linear splines over adaptive triangulations. *Signal Processing* 86, 1604–1616 (2006)
9. Davoine, F., Antonini, M., Chassery, J.M., Barlaud, M.: Fractal image compression based on delaunay triangulation and vector quantization. *International Conference on Pattern Recognition* 5, 3609 (2000)
10. Vos, J.: From lower to higher colour metrics: a historical account. *Clinical and Experimental Optometry* 89, 348–360 (2006)
11. Frese, T., Bouman, C.A., Allebach, J.P.: A methodology for designing image similarity metrics based on human visual system models. In: Proceedings of SPIE/IS & T Conference on Human Vision and Electronic Imaging II, pp. 472–483 (1997)
12. Jost, J.: Riemannian Geometry and Geometric Analysis, 5th edn. Springer, Heidelberg (2008)
13. Zéraï, M., Triki, O.: A differential-geometrical framework for color image quality measures. In: Bebis, G., et al. (eds.) ISVC 2010, Part III. LNCS, vol. 6455, pp. 544–553. Springer, Heidelberg (2010)
14. Von Helmholtz, V.: Handbuch der Physiologischen Optik. Voss, Hamburg (1896)
15. Stiles, W.S., Wyszecki, G.: Color Science Concepts and Methods, Quantitative Data and Formulae. John Wiley and Sons, Inc., Chichester (2000)
16. Eskicioglu, A.M.: Quality measurement for monochrome compressed images in the past 25 years. In: Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2000, vol. 6, pp. 1907–1910 (2000)
17. Sakuldee, R., Udomhunsakul, S.: Objective performance of compressed image quality assessments. *International Journal of Computer Science* 2, 258–267 (2007)

Human Tracking and Counting Using the KINECT Range Sensor Based on Adaboost and Kalman Filter

Lei Zhu and Kin-Hong Wong

Dept. of Computer Science & Engineering, The Chinese University of Hong Kong
`{lzhu,khwong}@cse.cuhk.edu.hk`

Abstract. Conventional methods for human tracking and counting are based on images captured by 2-D frontal cameras, which have a major problem of occlusion among the people to be counted. In our paper, we use a 3-D sensor (Kinect) to capture the top-down view of the flow of people at the entrance of a premise for human counting purposes. In particular we use the Head and Shoulder Profile (HASP) of a human as the input feature. Then we use an Adaboost algorithm built from weak classifiers sensitive to certain spatial input features for detecting human objects from the input. Therefore, our system can detect a human facing all directions correctly. After detection, a Kalman based tracker is used to track the detected human object and filter false detection, which improves the false positive detection rate significantly. Our experiment result shows that the system can detect and track human motion accurately in real time at about 20 Frames per second.

Keywords: Kinect, Human tracking, human detection, Kalman filter, Adaboost.

1 Introduction

Human tracking is a challenging and crucial problem in computer vision. It can be applied to surveillance, logistics and crowd control applications etc. Usually, it is achieved by using the 2-D image features captured by a camera. The detection systems will decide if a human image is found if the input features fit a predefined whole human model. For example, some upright pedestrian detection systems utilizing local features (HOG [1] and Shapelet [2]) are proposed. Some approaches focus on detecting a collection of partial body parts to describe a human model, for example, Pedro et al. [3] uses 10 body parts to model a human.

Another research direction relies on the depth map captured from 3-D sensors such as the Kinect sensor for detection. So it can avoid the problems due to illumination, color and texture diversity [4] etc. For example, J. Shotton et al. [5] extract human body pose from depth images. However, they all suffer the same problem of occlusion among people because their sensors are horizontally placed.

Motivated by the fact that there is no occlusion among humans when viewed from above, we use the Head and Shoulder Profile (HASP) in the depth map viewed from the top as the input. Another advantage is that HASP can also tell us at what direction the human subject is facing, which is useful in human motion tracking.

In this work, there are two main contributions:

Firstly, it is the first time top-down 3-D views based on the HASP model is used for human detection. Therefore, lighting conditions, color of the human subject have little effect on the outcome. For the processing, the HASP 3-D range data obtained by the Kinect sensor viewing from above is first transform into a 2-D image with the range (depth) data as the intensity. We now can operate on the 2-D depth image for human detection. We found that many 2-D human detection algorithms are applicable in our system using the 2-D dept image as the input. For example, Li et al. [6] uses the HOG feature to train a human detection by the Adaboost algorithm. Theoretically, a lot of human detection methods using the RGB camera can be used to detect the HASP in depth map. We will discuss how to apply these methods in our system later.

Secondly, our tracking framework is based on a combination of Adaboost detection algorithm with spatial features and Kalman filter. Because our Adaboost detection system cannot yield 100% detection rate, therefore, Kalman filter is used to improve the tracking accuracy to improve false detection rate. For example, if the human head is rotated 90 degrees from the middle horizontally, it may occlude the shoulder, the detection by Adaboost may fail temporary. Kalman filter is good at parameter tracking even input data is occasionally lost, so our tracking can still be maintained in these problematic situations. We have demonstrated successfully that the Adaboost and Kalman system work fine in the real environment to solve the problem we mentioned above in real time.

The reminder of this paper is organized as follows. In section 2, we describe the overview of our system. A human detection by the Adaboost algorithm is depicted in section 3. The human tracking method by Kalman filter is discussed in section 4. The experimental result is shown in section 5. We present the conclusion and discuss the future work in section 6.

2 Overview of System

The target of our system is to track humans who come into a predefined area. An example is shown in Fig. 1 which illustrates a 3D sensor Kinect is mounted above an entrance to capture a 640*480 depth image. We understand that the Kinect can only work within a certain range in depth, however, correction lens can be used and are available in the market if the working distance is outside the typical Kinect working range. Fig. 3a shows such an image and the grey level pixels represent the depth values measured by the 3D sensor and the object. We will use this depth map as the input for human tracking and counting. As for the output, when a human object is detected, a rectangle is drawn automatically to present the detection and tracking result. Besides, the number of human who

has come into the environment will be recorded and the number of human who has come out will also be computed and recorded. Thus, we can also count the number of humans who remain in the area.

The system work flow is shown in Fig. 2a. The input, a depth map from the Kinect sensor, is used for our Adaboost classification system for human detection. After classification, the detected human positions are regarded as the measurement to update the automating predication model by Kalman filter. Detailed processing of the Adaboost classifier and Kalman filter will be discussed below.

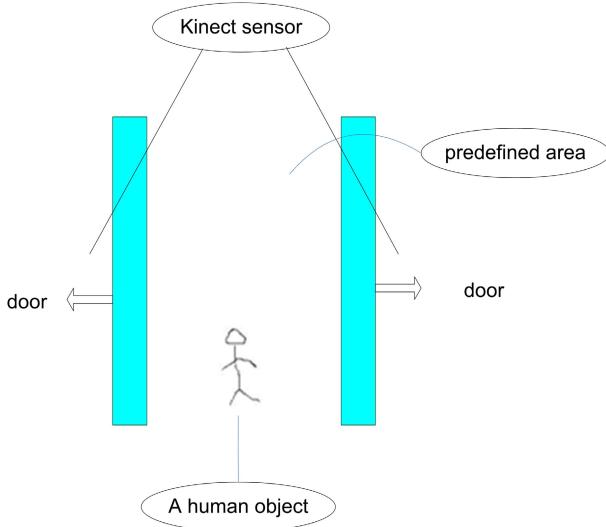


Fig. 1. The setup of the Kinect sensor and the predefined area

3 Human Detection by Adaboost

The first step of our tracking system is detection of HASP for a human for each frame generated by the Kinect sensor using the Adaboost algorithm. To detect where all humans are in a specific frame, we first build a processing window. We slide the processing window across the 3D range image of the predefined area and see if a human object is detected in a specified processing window. The procedure is described in Fig. 2b.

3.1 Spatial Features

Our intuitive detection criteria are inspired by the following observations. (Criterion-1) From the HASP (head and shoulder profile) image of a typical human object we see that there is an empty space in the front and back of the

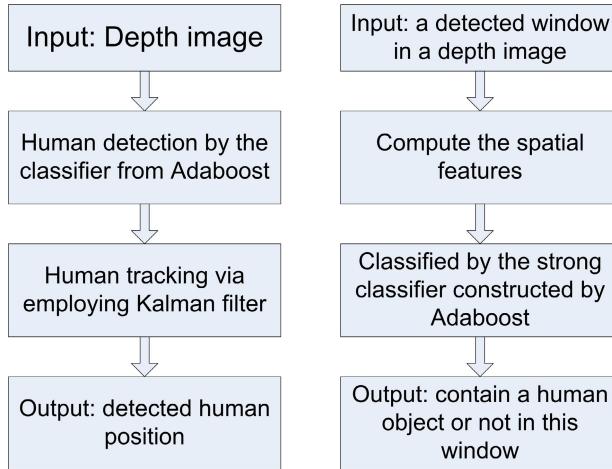


Fig. 2. a) The work flow of our system. b) The detection procedure for a specific window.

head. (Criterion-2) Besides, the right side of right shoulder and the left side of left shoulder are also empty. (Criterion-3) Moreover, there is a height difference between the head and the two shoulders. Those criteria depict the spatial information of HASP, thus, we call those features as spatial feature. And those criteria can be expressed by the difference of two pixel areas in the depth map, which is a Haar-like feature.

To compute the spatial feature, we formed 20 redefined sub-windows indexed by n ($W_s(n)$, $n=0,..,19$) inside the processing window. The processing window and sub-windows with indexes are shown in Fig. 3a. We use four Haar-like operators to assist us to measure how well the image satisfies certain detection criteria. The four Haar-like features are shown in Fig. 3b. Similar to the Viola-Jones [7] face detection method using Adaboost, a depth integral image is used to compute all pixel values in a rectangle area to rapidly generate the Haar-features.

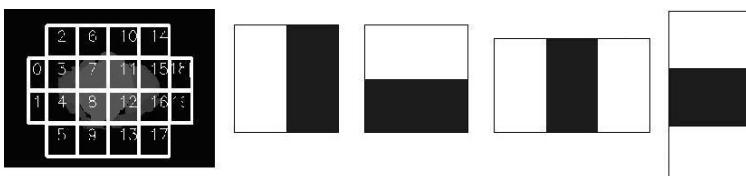


Fig. 3. a) The processing widow (PW) with 20 predefined sub-windows ($W_s(n)$). The numbers show indexes (n) of the sub-windows. b) Haar-like features 1,2,3,4 from left to right used in our system.

3.2 Adaboost Algorithm

Adaboost (Adaptive Boost) is a boosting algorithm introduced by Freund et al [8], which constructs a strong classifier by a linear combination of weak classifiers with a weight for each classifier. However, a challenge of our top view human object detection is that the human can stand and face all directions and with many postures. Therefore, a horizontal window (shown in Fig. 4a) is first employed to scan the depth image, and we define the final strong classifier as follows:

$$F_{hor}(win) = \begin{cases} 1; & \text{if } \sum(\alpha_j * H(j)) \geq \theta_{hor} * \sum(\alpha_j) \\ 0; & \text{otherwise} \end{cases} \quad (1)$$

where α_j is the weight of weak classifier $H(j)$; θ_{hor} is the threshold of classifier F_{hor} . This horizontal processing window is designed for the person facing toward the exit of the gate (see Fig. 1a). However, we find that this classifier is not able to find human objects whose shoulders are almost parallel to the entrance line. Therefore, a vertical window in Fig. 4b is used to detect human objects, and the strong classifier is:

$$F_{ver}(win) = \begin{cases} 1; & \text{if } \sum(\alpha_k * H(k)) \geq \theta_{ver} * \sum(\alpha_k) \\ 0; & \text{otherwise} \end{cases} \quad (2)$$

where α_k is the weight of weak classifier $H(k)$; θ_{ver} is the threshold of vertical strong classifier F_{ver} . Obviously, this vertical processing window is designed for the person facing perpendicular to the direction towards to exit of the gate and the human object with an almost horizontal shoulder is hard to be detected using this classifier. Thus, in our system, we create a window [Fig. 4c] named union window to detect the human objects in the depth map, which can be regarded as a combination of the horizontal window and the vertical window and the classifier function is:

$$F_C(win) = F_{hor}(win) \parallel F_{ver}(win) \quad (3)$$

The union window contains both horizontal strong classifier and vertical strong classifier, which can save the computation time for deciding whether a human is in the window.

4 Kalman Filter Tracking

Kalman filter[9] is a linear prediction method of state estimation from a large number of measurements over time. In our system, a Kalman filter will be created when a new human object is coming into the Kinect viewing area. The state for the Kalman filter to model the human dynamic motion is defined as follow:

$$w = (x, y, \dot{x}, \dot{y}) \quad (4)$$

Where x is the horizontal coordinate and y is the vertical coordinate of a pixel in the depth map. \dot{x} is the velocity of the motion of x and \dot{y} is the velocity

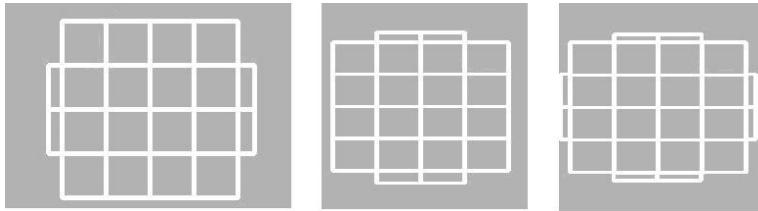


Fig. 4. Three processing windows. (a) Horizontal window. (b) Vertical window. (c) Union window.

of the motion of y . The velocity is assumed to be constant in the sampling period. Therefore, the state transition equation and the observation equation are respectively:

$$w_t = Aw_{t-1} + \gamma_t, A = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5)$$

$$\varepsilon_t = g_t(w_t) + v_t, g_t(w_t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \quad (6)$$

Where γ_t and v_t are zero-mean Gaussian noise for model prediction and correction by measurement. ε_t is the detected position by the strong classifier in the depth map. $g_t(w_t)$ is a 2×4 matrix called the observation matrix, which is to correct the state estimated values in the prediction model by the measurement ε_t .

5 Experiments and Results

The depth map from the Kinect sensor, which is hung at the ceiling of our lab, should also be pre-processed, so that the pixel value represents the distance from the floor to the object surface. Note that the depth values of some pixels are 0. But our method is robust to those noise, as the spatial features are the average of the differences of several windows's pixel values. Thus, even though the depth values of some pixels(noise) in the window are 0, the spatial features defined on the several windows are almost unchanged.

5.1 Training Datasets

To construct a strong classifier by Adaboost, we build a dataset for the training process with positive samples and negative samples. For each specific object, we took many depth maps of this object by rotating a certain degree. Our positive training set consists of 720 depth pictures and the negative training dataset has 288 pictures. Fig. 5 shows some samples from a person and a chair.



Fig. 5. a) Positive samples (a person) with 0, 45, 90, 135. b) Negative samples (a chair) with 0, 45, 90, 135.

5.2 Human Detection Evaluation by Adaboost

Dataset 1 contains 50 depth images with only one human standing in different directions and postures, and dataset 2 is composed of 50 depth images with two humans, and the two humans are standing separately. 70 depth images with 3 or more human objects from Kinect sensor consist of Dataset 3, where human objects in some images stand closely with each other.

Note that if the window size is smaller, the human standing with a big Head and Shoulder Profile (HASP) can not be detected. If the window size is bigger, two humans standing closely can not be detected, which also decrease the detection rate. Thus, we just change the size of sub-window. The right detection rates with different sub-window sizes ($40 * 40$ pixels, $45 * 45$ pixels, $50 * 50$ pixels) are shown in Fig. 6. Fig. 7 illustrates the detection results with $45 * 45$ sub-windows. On the other hand, detection rate in Dataset 3 is lower than other datasets' detection rates, as there are many samples in Dataset 3 containing human objects standing too closely to each other so that the two HASPs are emerged together. Therefore, we can conclude that our human detection algorithm can detect humans who are not too close to each other, but the more closely two humans are standing next to each other, the lower detection rate that can be achieved, which obey what we predicted. However, usually the time when two humans are standing very close to each other is very short in real tracking, and they tend to be separated after a short while. This situation requires that we can accurately track the human positions after the two emerged HASPs, which can be solved by Kalman filter.

5.3 Human Tracking Evaluation

We first detect the human object in the depth image manually, which will serve as the ground truth. The detection rate is measured by the overlapping rate of the manual detected window (the ground truth) and the machine detected window. Two videos (video 1 and video 2) are used for the tracking test in our system. The first video (video 1) records a human walking under the viewing area of the Kinect with different standing directions and postures. The tracking results of some frames in the video 1 are shown in Fig. 8, and Table 1 shows the overlapping rate for each 10 frames in video 1. The average computation time of each frame in first video is about 33ms.

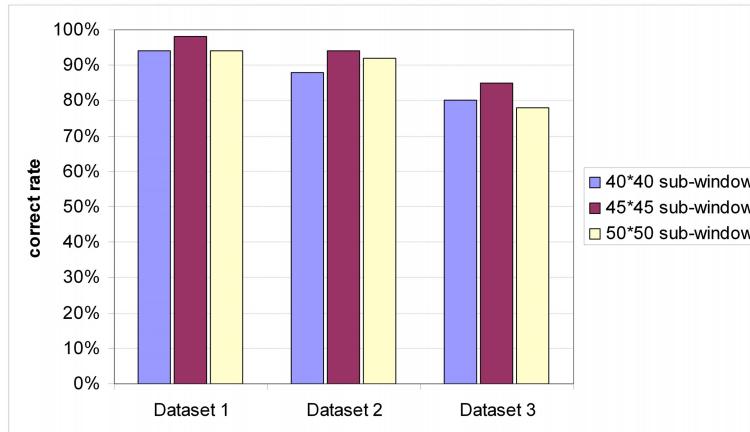


Fig. 6. Correct detection rate of different datasets with different window sizes

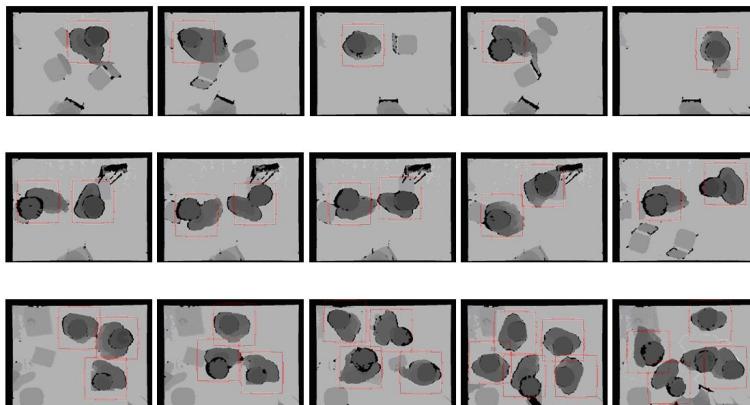


Fig. 7. (First row) Detection result of 5 images from Dataset 1. (Second row) Detection result of 5 images from Dataset 2. (Third row) Detection result of 5 images from Dataset 3.

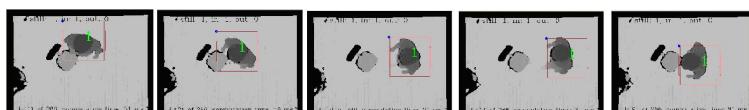


Fig. 8. Tracking result of each 20 frames in video 1 starting from frame 11

Table 1. The overlapping rate for the specific frames in video 1

Frame index	11	21	31	41	51	61	71	81	91
Overlapping rate(%)	91.8	88.5	94.5	98	97.2	89	84	97.2	85.5

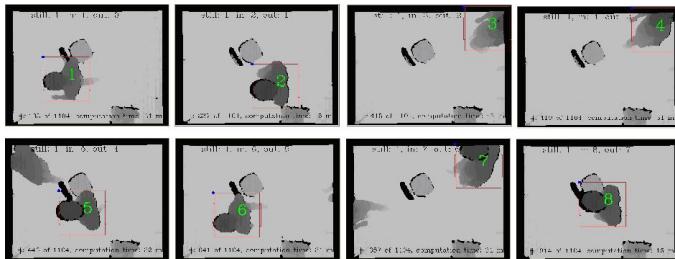


Fig. 9. Tracking result of video 2 with 8 humans going through the view area

Table 2. The overlapping rate for the specific frames in video 2

Frame index	138	237	415	419	446	641	657	914
Overlapping rate(%)	94.4	91.8	91.8	91.7	84.0	89.1	89.1	89.2

The second video records multiple humans coming into the viewing area and coming out from this area with different directions one by one. The tracking results of some frames in video 2 are shown in Fig. 9, and Table 2 illustrates the overlapping rates corresponding with the frames shown in Fig. 8. The average computation time of each frame in video 2 is about 32ms.

6 Conclusion

We constructed a real-time human detection and tracking based on the depth map from a commercially available 3D range sensor (Kinect sensor). The Kinect sensor is hung 2-5 meters above an area for detection, so any human passing through this area can be sampled and detected. The 3D map of the human object is described by a featured data structure called the Head and Shoulder Profile (HASP). The system is able to count the number of people passing through the Kinect viewing area. There are two techniques used in our system. The first is human detection by a strong classifier constructed by the Adaboost algorithm with sample spatial features. The second is the use of Kalman filter to track the moving objects. We found that Kalman filter can make sure that the same person is tracked correctly. In the future, we will focus on improving the detection result by using different feature descriptors and different learning methods.

Acknowledgement. This work is supported by a direct grant (Project Code: 2050455) from the Faculty of Engineering of the Chinese University of Hong Kong.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
2. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8. IEEE (2007)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision 61, 55–79 (2005)
4. Hernández-Vela, A., Zlateva, N., Marinov, A., Reyes, M., Radeva, P., Dimov, D., Escalera, S.: Graph cuts optimization for multi-limb human segmentation in depth maps. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 726–732. IEEE (2012)
5. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM 56, 116–124 (2013)
6. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4. IEEE (2008)
7. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
9. Kalman, R.E., et al.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82, 35–45 (1960)

Hand Pose Estimation from a Single RGB-D Image^{*}

Alina Kuznetsova and Bodo Rosenhahn

Institute for Information Processing (TNT),

Leibniz University Hanover, Germany

{kuznetso,rosenhahn}@tnt.uni-hannover.de

Abstract. Hand pose estimation is an important task in areas such as human computer interaction (HCI), sign language recognition and robotics. Due to the high variability in hand appearance and many degrees of freedom (DoFs) of the hand, hand pose estimation and tracking is very challenging, and different sources of data and methods are used to solve this problem. In the paper, we propose a method for model-based full DoF hand pose estimation from a single RGB-D image. The main advantage of the proposed method is that no prior manual initialization is required and only very general assumptions about the hand pose are made. Therefore, this method can be used for hand pose estimation from a single RGB-D image, as an initialization step for subsequent tracking, or for tracking recovery.

1 Introduction

Precise hand pose estimation and tracking is an important task in areas such as HCI, sign language recognition and robotics. Different data sources and methods are used to solve this task. One of the possible settings is to use recently introduced consumer range cameras that produce both color(RGB) and depth(D) data streams. Although the use of such devices makes the hand pose estimation task much more feasible, it is still unsolved, both because of general problems of the hand pose estimation and drawbacks of consumer RGB-D cameras.

In general, hand pose estimation is very challenging due to the many degrees of freedom (DoFs) of the hand as an articulated object, which leads to great variability in hand appearance and self-occlusions.

The main drawbacks of the available RGB-D cameras are low resolution and missing range data.

There are three main groups of methods used to estimate the hand pose: template-based methods, model-based methods and machine learning methods, as well as different combinations.

* This work has been partially funded by the ERC within the starting grant Dynamic MinVIP.

In our work, we concentrate on a model-based hand pose estimation method combined with feature detection. We use both color and depth images to overcome the problem of missing depth data. As a result, our method is able to correctly determine the hand pose from a single RGB-D image.

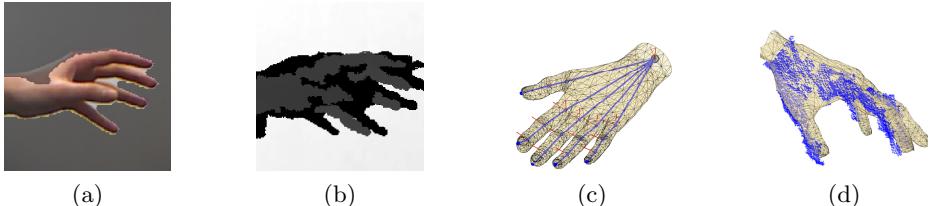


Fig. 1. Initial data (1a — color image, overlaid with hand silhouette, 1b—depth data), hand model (1c), fitted hand and point cloud in 3D(1d).

2 Related Works

All existing methods for hand pose estimation can be divided into three rough groups: template-based methods, model-based methods and machine learning methods [1]. There exist, of course, different combinations of these methods. Each group of methods has its own strengths and weaknesses.

Template-based methods. Template-based methods are usually applied in a single or multiple image setting [2]. To apply these methods, one firstly creates a database of possible hand poses and then uses this database to find a pose closest to the input image. The work for shape matching was recently extended to incorporate depth data in [3]. There are several limitations of this approach: firstly, this approach can only be used to distinguish between a limited number of poses and the discrimination power decreases with the increasing number of poses; and secondly, it requires the creation of a database that strongly depends on experimental setup.

Model-based methods. Recently, more works appeared on model-based approach for hand pose estimation and tracking. In these methods, a general model of a hand, parametrized by continuous parameters, is fitted to an image, based on the evaluation of similarity between the model and the image. In case of a single image, one often constraints the number of parameters to a subset of the 26 full DoF [4], or strong assumptions about the image are made [5,6]. Several types of clues are used to evaluate the model, such as edges, optical flow [7], silhouettes, shading [8], color gloves [9], etc. The main disadvantages of model-based methods are that they are not real-time, not stable due to the high number of degrees of freedom and occlusions, and require prior initialization. It is also challenging to apply these methods in case of missing data.

Machine learning methods. Machine learning methods have been rarely applied to hand tracking due to the high variability in hand appearance and difficulty to distinguish different hand parts and find stable features [10]. However,

due to the appearance of the RGB-D sensors, machine learning methods can be used for hand pose estimation more efficiently [11]. It is still difficult to apply them though, due to the shortcomings of the data delivered by depth sensors.

In our work, we concentrate on model-based approach for hand pose estimation and propose to overcome several disadvantages mentioned above with a method that:

- determines the hand pose from a single image;
- does not require any initialization or model fitting;
- partially overcomes the problem of missing data;
- can be easily extended to the tracking setting.

This paper has the following structure. In Section 3, we present the data and explain the main challenges of hand pose estimation. In Section 4, we explain the main steps of the algorithm. In Section 5, we evaluate our approach both on real and synthetic data. We conclude our work in Section 6 with a discussion of the advantages and the disadvantages of our approach and the possible extensions.

3 Experimental Setting and Data

For our work, we use the Microsoft Kinect sensor. This sensor delivers depth and color data at VGA resolution. Depth data is available at a distance of 0.6m-4m.

The data itself is hard to work with due to noise and missing depth data (see Fig. 2). Missing depth data occurs because of the two reasons: self occlusions prevent Kinect to acquire the correct depth values for some parts of the hand and the fingers themselves are hard to capture, because they are thin objects.

Because of the problems mentioned above, only a single depth frame alone can not be used for hand pose estimation. Therefore, we use RGB data to create hand silhouettes using background subtraction, so that we are able to partially recover from missing depth data. On the other hand, as mentioned in Section 2,

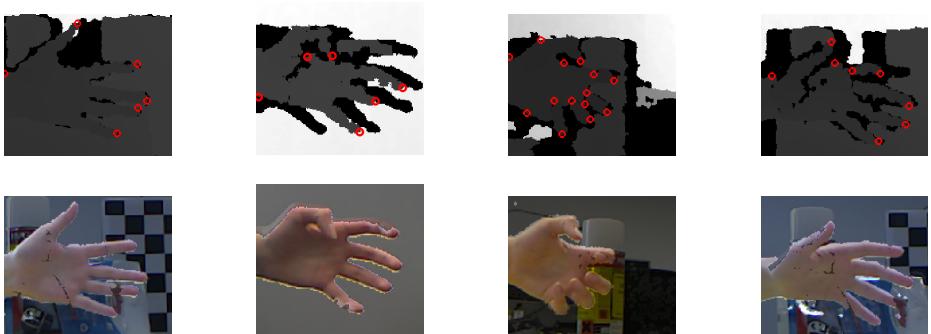


Fig. 2. Several examples of the RGB-D data; black areas of the depth images denote missing depth data; color images are overlaid with the extracted silhouette; initial finger tips detections are shown as red points.

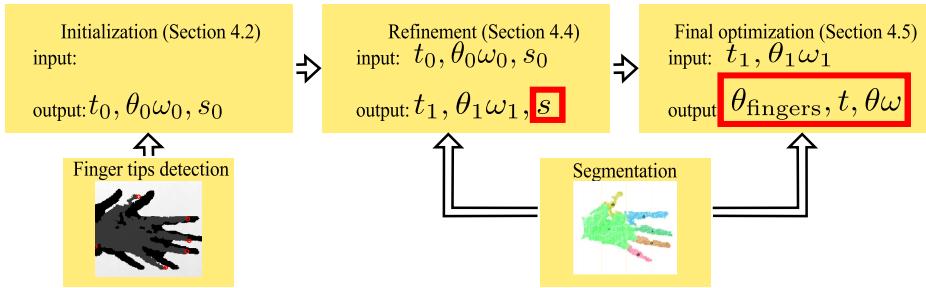


Fig. 3. The work-flow of the algorithm; the final output parameters are marked in red

single color images are usually not enough for hand pose estimation, and many ambiguities can be resolved by using depth data.

We make general assumptions about the hand in the image — we assume that it is a left hand, rotated so that the palm is at least partially visible.

4 Hand Pose Estimation Algorithm

Our algorithm consists of a number of steps (see Fig 3). We describe each step of the algorithm in more details in the following sections.

Hand model. We use a 26 DoF hand model, that consists of a 3D mesh of medium resolution of $N_m = 1317$ vertices and a skeleton connected to the model using the standard skinning technique [12]. The hand pose is defined by 20 joint angles θ_{fingers} between the bones. The bone transformation is parametrized using exponential mapping [13], which is typically used for model-based human pose estimation [14]. Additionally, 6 parameters $t, \theta\omega \in \mathbb{R}^3$ encode position and rotation of the hand. Here $\theta\omega$ is the rotation by the angle θ around the axes ω .

4.1 Initialization Step

For initialization, we detect finger tips using geodesic extrema (see [15]). Geodesic extrema are computed by propagating distance from one point on the point cloud to the other points and then taking the maximum. Note, that the detected points are most likely not the positions of the actual finger tips (see Fig 2). We use these points to determine an approximate hand rotation and scaling. Since we do not know correspondences between the detected finger tips and those of the model fingers, we check all possible matches. For each match, we find the corresponding rotation and translation of the model, and evaluate the error between the model and the detections as the average Euclidean distance between corresponding points. We then select the match that delivers the smallest cost. An example of initialization result is given in Fig 5.

4.2 Point Cloud Segmentation Using Region Growing

For pose estimation, we use an ICP-based optimization technique [16]. It is well known that such technique is prone to false matches. Therefore, we pre-segment our point cloud to differentiate between fingers and palm parts.

We use region growing based segmentation on the depth images: a pixel is included into a current region if the two distance measures, (1) and (2), are smaller than the corresponding thresholds. We define empirical thresholds at the 10mm for the Euclidean distance (1) and 0.8750 for the scalar product (2) between normals.

$$d_{eucl}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^3 (x_i - y_i)^2 \right)^{\frac{1}{2}}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^3, \quad (1)$$

$$d_{norm}(\mathbf{x}, \mathbf{y}) = \langle \mathbf{n}_x, \mathbf{n}_y \rangle. \quad (2)$$

Here \mathbf{n}_x and \mathbf{n}_y are the normals to the point cloud at the corresponding points \mathbf{x} and \mathbf{y} . To compute the normals we use principal component analysis (PCA) as in [17].

After initial segmentation we merge small regions and divide large regions. The segmentation results are presented in Fig 4.

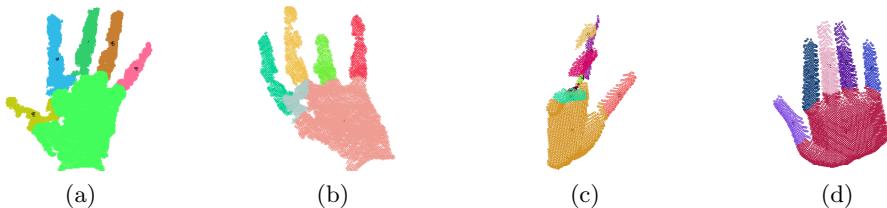


Fig. 4. Segmentation results for the real data produced by Kinect (4a and 4b) and the artificial dataset [18] (4c and 4d)

After the segmentation, we determine the palm region as the largest region, and the rest of the points belong to the fingers. The point indices of the point cloud, corresponding to the palm, are denoted by I_{palm} , whereas the indices, corresponding to the i -th finger, are denoted by I_{finger}^i ; the union of all non-palm points is denoted by I_{fingers} .

4.3 Hand Position and Size Refinement

As mentioned in Subsection 4.1, the initial hand pose and scaling estimation can be inaccurate. Therefore, we firstly refine these parameters using an ICP technique. The parameter vector has seven dimensions: $(\mathbf{t}, \theta\omega, s)$, where \mathbf{t} and $\theta\omega$ define hand position and rotation, and s defines scaling.

We then optimize the functional:

$$E(\mathbf{t}, \theta\omega, s) = E_{\text{palm}} + \alpha E_{\text{fingers}} + \beta E_{2D} . \quad (3)$$

Here E_{palm} is the error term responsible for the distance between the point cloud region, identified as palm, and the palm of the model.

$$E_{\text{palm}} = \frac{1}{|I_{\text{palm}}|} \sum_{i \in I_{\text{palm}}} (\mathbf{p}_i^{\text{pc}} - \mathbf{p}_{n(i)}^m)^2, \quad \mathbf{p}_i^{\text{pc}}, \mathbf{p}_{n(i)}^m \in \mathbb{R}^3 . \quad (4)$$

Here, for each point \mathbf{p}_i^{pc} of the point cloud we search for the nearest palm point $\mathbf{p}_{n(i)}^m$ in the hand model. We pre-build a kd-tree [19] on the data points to accelerate point search.

The E_{fingers} term is responsible for the distance between non-palm points in the point cloud and non-palm points of the model:

$$E_{\text{fingers}} = \frac{1}{|I_{\text{fingers}}|} \sum_{i \in I_{\text{fingers}}} (\mathbf{p}_i^{\text{pc}} - \mathbf{p}_{n(i)}^m)^2, \quad \mathbf{p}_i^{\text{pc}}, \mathbf{p}_{n(i)}^m \in \mathbb{R}^3 . \quad (5)$$

Here $\mathbf{p}_{n(i)}^m$ is the nearest non-palm point from the model. Finally, the E_{2D} term defines the distance between the 2D silhouette of the hand, and the projection of the model:

$$E_{2D} = \frac{1}{N_{\text{sill}}} \sum_i (\mathbf{p}_i^{\text{sill}} - \text{Pr}(\mathbf{p}^m)_{n(i)})^2 + \frac{1}{N_m} \sum_j (\mathbf{p}_{n(j)}^{\text{sill}} - \text{Pr}(\mathbf{p}^m)_j)^2, \quad (6)$$

$$\mathbf{p}_i^{\text{sill}}, \text{Pr}(\mathbf{p}^m)_j \in \mathbb{R}^2 . \quad (7)$$

Here $\mathbf{p}_i^{\text{sill}}$ is a 2D point of the silhouette with N_{sill} points, $\text{Pr}(\mathbf{p}^m)_{n(i)}$ is the nearest projected model point, where $\text{Pr}(\mathbf{p})$ is the perspective projection operator with the default Kinect calibration parameters.

Since the whole functional can be represented as a sum of squares, we use the trust region approach for non-linear optimization (see, for example, [20]). Of course, the minimum found is not guaranteed to be the global minimum, and therefore the fit is not perfect. We fix the weight from (3) $\alpha = 0.1$ and $\beta = 1.3$ for the real data. After this stage, we get a much better fit between the hand model and the point cloud (see Fig 5).

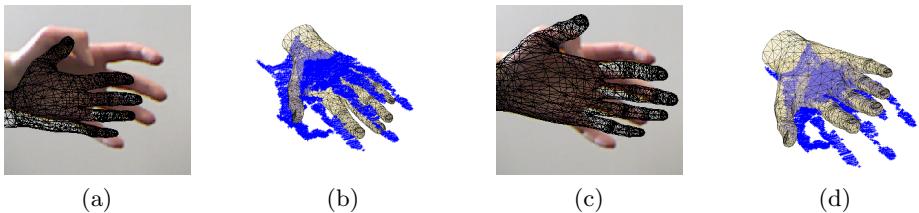


Fig. 5. Hand fitting initialization (5a and 5b) and refined hand pose and size (5c and 5d). The final hand pose estimation follows in the last step.

4.4 Final Hand Pose Estimation

In this stage, we do the final hand pose estimation. On the previous step, we estimated hand pose and size, but the fingers parameters θ_{fingers} are still unknown. The two steps are separated because otherwise dependencies between the parameters and the error functional become more complex. Moreover, initialization described in Subsection 4.1 is not good enough to avoid false matches. We now use a variation of a non-rigid ICP algorithm to fit the pose of our model to the point cloud and the silhouette.

As mentioned above, every ICP algorithm is prone to false matching. It is especially hard problem for such a highly articulated object as the hand, since fingers tend to be matched incorrectly. Therefore, prior to optimization, we match different parts of the model to different parts of the point cloud. For matching each part of the point cloud to a part of the model, we define a cost for each match as Euclidean distance between the centers of two segments. Each model segment can only be matched to zero or one point cloud segment. The problem can be mathematically written as follows:

$$\mu = \operatorname{argmin}_{\mu} \sum_i d_{\text{eucl}}(\mathbf{c}_i^{\text{pc}}, \mathbf{c}_{\mu(i)}^m), \quad \mu(i) \in \{1, \dots, 5\} . \quad (8)$$

where \mathbf{c}_i^{pc} is the center of the i -th region of the point cloud and $\mathbf{c}_{\mu(i)}^m$ is the center of the $\mu(i)$ -th region of the model. Note, that palm region is already known, so it is excluded from the matching procedure. The solution μ is found using brute force search, since the number of possible combinations is low.

The full pose of the hand is parametrized by 26 parameters: $(\theta_{\text{fingers}}, \mathbf{t}, \theta\omega)$, where $\mathbf{t}, \theta\omega$ are defined in the previous section, and θ_{fingers} is a 20-dimensional vector of joint angles defining fingers' position. We also add natural constraints on the θ_{fingers} parameters.

To find the parameter values, we optimize the following functional:

$$E(\theta_{\text{fingers}}, \mathbf{t}, \theta\omega) = E_{\text{palm}} + \left(\sum_i E_{\text{fingers}}^i \right) + \zeta E_{2D} . \quad (9)$$

The first and the last term have exactly the same meaning as in the previous step. The additional term can be described as follows:

$$E_{\text{fingers}}^i = \frac{1}{|I_{\text{finger}}^i|} \sum_{j \in I_{\text{finger}}^i} (\mathbf{p}_j^{\text{pc}} - \mathbf{p}_{n(j, \mu(i))}^m)^2 . \quad (10)$$

Here $n(j, \mu(i))$ is the closest model point to the point j , coming from $\mu(i)$ region of the model and $|I_{\text{finger}}^i|$ is the number of indices in the set I_{finger}^i . In this way, smaller finger regions gain more weight and have more influence in the optimization.

The last 2D term is weighted with the weight $\zeta = 1$ for the real data, which determines the importance of the silhouette.

We use the same algorithm for constrained non-linear optimization, as in the previous section. The solution of the optimization problem delivers the full pose and size estimation of the hand.

5 Evaluation

We evaluate our results both on the real data and the synthetic data (consisting of depth data and the corresponding silhouette).

Experiments with the synthetic data. We evaluated our algorithm on the synthetic data used for evaluation in [18]. Since we are not doing tracking, it would be unfair to compare the results we achieved directly with the numbers provided in [18]. Note, that the model used to produce synthetic data in this case differs significantly from our model in shape and proportions.

Since for synthetic data joint positions are known, we measured the distance between the joint positions of the fitted model and the ground truth joint positions. We evaluate the error after each step of our method to show its impact.

In Fig 6, mean error for each hand part is represented. One can see, that after the hand position and scale refinement, fingers error becomes larger. The explanation for this is simple — the position refinement gives more weight to the palm, and therefore finger error can actually become greater. As expected, after the final fit the error is the smallest then on the two previous steps.

Note, that our estimation accuracy is close to the one reported in [20] (in 74% the estimated pose deviation 40mm or less from the ground truth), even though we estimate the pose from a single frame instead of tracking and the model we use differs from the one used for creating the artificial data significantly.

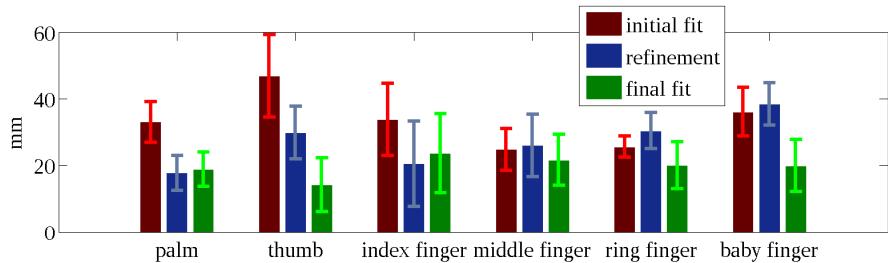


Fig. 6. Mean error of the six hand parts: palm, thumb, and four fingers; error bars show standard deviation; $\alpha = 0.1$, $\beta = 1.3$, $\zeta = 2$; the palm error is measured as the mean error of metacarpophalangeal joints and palm origin; the finger error is measured as the mean error of all joints of the finger

In general, we could observe that the changes of the parameters $\alpha \in [0.1, 2]$ and $\zeta \in [0.1, 2]$ in the corresponding intervals do not affect the average results significantly. However, changes in β have significant influence on the results. For example, in case of $\beta = 5$ for the artificial data, there is almost no difference in

error between the initial and the final fit. We attribute it to the fact, that the β parameter is crucial for determining the correct hand scaling.

Experiments with the real data. For these experiments, we used the data recorded by the Kinect sensor. In Fig 7, the results of the full fitting procedure are shown. In general, the pose of the hand is fitted correctly. Such estimation is enough for gesture recognition and also for tracking initialization. But one can see that the match is not perfect. We attribute it to several factors. Firstly, there is a mismatch between hand form and proportions, and the real hand. Secondly, the scaling parameter is critical for the following pose estimations, so small mistakes in the scaling parameter directly lead to errors during fitting.

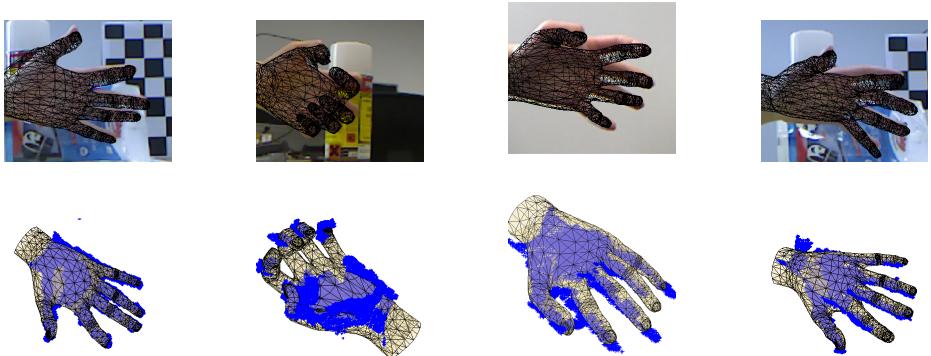


Fig. 7. Several model fitting results on the images from the database; the upper row shows color image with fitting results; the bottom row shows point cloud to model fit in 3D; the weights used are $\alpha = 0.1, \beta = 5, \zeta = 1$

Our algorithm performs good in case of an open hand and a partially closed hand (see Fig 7). However, we observed that in case of a fully closed hand our algorithm fails to estimate hand pose correctly, unless provided with initial pose, that is close to the pose on the image. We believe, that this problem can be potentially solved by learning an approximate pose from an image prior to optimization.

6 Conclusion and Future Work

We presented an approach for model-based hand tracking, based on the classical non-rigid ICP algorithm for combined RGB-D data. We evaluated our approach both on the synthetic and the real-world data and showed, that it is capable of producing plausible hand pose estimations. Our approach can be used both for one-shot hand pose estimation and extended for tracking. Our approach allows to partially overcome the problem of missing data.

In future, we will focus on improving the speed and also on providing a more stable initialization. As we use local optimization methods, our approach is prone

to get stuck in a local minima, so we would like to extend it to usage of global optimization methods. Another possible direction could be to couple it with machine learning or template-based matching, since they are proven to be helpful for hand pose recognition.

References

1. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst.* 108, 52–73 (2007)
2. Stenger, B.: Template-based hand pose recognition using multiple cues. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 551–560. Springer, Heidelberg (2006)
3. Doliotis, P., Athitsos, V., Kosmopoulos, D., Perantonis, S.: Hand shape and 3D pose estimation using depth data from a single cluttered frame. In: Bebis, G., et al. (eds.) ISVC 2012, Part I. LNCS, vol. 7431, pp. 148–158. Springer, Heidelberg (2012)
4. Stenger, B., Mendonça, P.R.S., Cipolla, R.: Model-based 3d tracking of an articulated hand. In: CVPR (2), pp. 310–315 (2001)
5. Bray, M., Koller-Meier, E., Mueller, P., Gool, L.V., Schraudolph, N.N.: 3d hand tracking by rapid stochastic gradient descent using a skinning model. In: 1st European Conference on Visual Media Production, CVMP, pp. 59–68 (2004)
6. de La Gorce, M., Fleet, D.J., Paragios, N.: Model-based 3d hand pose estimation from monocular video. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1793–1805 (2011)
7. Ballan, L., Taneja, A., Gall, J., Van Gool, L., Pollefeys, M.: Motion capture of hands in action using discriminative salient points. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 640–653. Springer, Heidelberg (2012)
8. de La Gorce, M., Paragios, N., Fleet, D.J.: Model-based hand tracking with texture, shading and self-occlusions. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2008, Anchorage, Alaska, USA, June 24–26. IEEE Computer Society (2008)
9. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. *ACM Trans. Graph.* 28, 63:1–63:8 (2009)
10. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Hand pose estimation using hierarchical detection. In: Sebe, N., Lew, M., Huang, T.S. (eds.) ECCV/HCI 2004. LNCS, vol. 3058, pp. 102–112. Springer, Heidelberg (2004)
11. Keskin, C., Kiraç, F., Kara, Y.E., Akarun, L.: Real time hand pose estimation using depth sensors. In: ICCV Workshops, pp. 1228–1234 (2011)
12. Jacka, D., Merry, B., Reid, A.: A comparison of linear skinning techniques for character animation. In: Afrigraph, pp. 177–186. ACM (2007)
13. Murray, R.M., Sastry, S.S., Zexiang, L.: A Mathematical Introduction to Robotic Manipulation, 1st edn. CRC Press, Inc., Boca Raton (1994)
14. Pons-Moll, G., Rosenhahn, B.: Model-Based Pose Estimation. Springer (2011)
15. Plagemann, C., Ganapathi, V., Koller, D., Thrun, S.: Real-time identification and localization of body parts from depth images. In: 2010 IEEE International Conference on Robotics and Automation, ICRA, pp. 3108–3113 (2010)
16. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 239–256 (1992)

17. Rusu, R.B.: Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany (2009)
18. Iason Oikonomidis, N.K., Argyros, A.: Efficient model-based 3d tracking of hand articulations using kinect. In: Proceedings of the British Machine Vision Conference. BMVA Press (2011)
19. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM 18, 509–517 (1975)
20. Coleman, T.F., Li, Y.: An interior trust region approach for nonlinear minimization subject to bounds (1996)

3D Human Tracking in a Top View Using Depth Information Recorded by the Xtion Pro-Live Camera

Cyrille Migniot and Fakhreddine Ababsa

IBISC - team IRA2 - University of Evry val d'Essonne, France
{Cyrille.Migniot,Fakhr-Eddine.Ababsa}@ibisc.fr

Abstract. This paper addresses the problem of the tracking of 3D human body pose from depth image sequences given by a Xtion Pro-Live camera. Human body poses could be estimated through model fitting using dense correspondences between depth data and an articulated human model. Although, most of the time for the video surveillance, the camera is placed above the persons, all the tracking methods use the front view. Indeed the human shape is more discriminative in this view. We propose a new model to be fitted to the top view in a particle filter framework for a real-time markerless tracking. The model is composed of two parts: a 2D model providing the human localization and a 3D model providing its pose. There are few wrong estimations and they are efficiently detected by a confidence measure. . . .

1 Introduction

In recent years, there is a body of research on the problem of human parts detection, pose estimation and tracking from 3D data. The image is fitted to an articulated model that embodies the possible human movements. The great majority of the methods in the literature use a model adapted to a front view of the person because the shape of a person is much more discriminative on this orientation. Furthermore the color of the skin and the elements of the face is seldom available in a top view. Nevertheless, in the application of the video-surveillance, the camera is frequently installed above the persons. The tracking on a top view need to use depth feature. One of the more popular devices used to provide it is kinect, which has sensors that capture both rgb and depth data. Here, we simulate the installation of a Xtion Pro-Live cameras (depth-color camera launched by Assus in 2013) in the ceiling of a supermarket. The goal is to analyze the behaviors of the customers during their buying acts within the shelves. The first step is tracking the pose of a customer. Then it will be recognized by an other treatment.

The main challenge in articulated body motion tracking is the large number of degrees of freedom to be recovered. For non-linear or non-Gaussian problems, the particle filter algorithm [1] has became very popular. Based on the Monte-Carlo simulation, it provides a suitable framework for state estimation in a non-linear,

non-Gaussian system. At moment k , let x_k be the state of the model and y_k be the observation. Particle filter recursively approximates the posterior probability density $p(x_k|y_k)$ of the current state x_k evaluating the observation likelihood based on a weighted particle sample set $\{x_k^i, \omega_k^i\}$. Each of the N particles x_k^i corresponds to a random state propagated by the dynamic model of the system and weighted by ω_k^i . There are 4 basic steps:

- **resampling:** N particles $\{x_k^i, \frac{1}{N}\} \sim p(x_k|y_k)$ from sample $\{x_k^i, \omega_k^i\}$ are resampled. Particles are selected by their weight: large weight particles are duplicated while low weight particles are deleted.
- **propagation:** particles are propagated using the dynamic model of the system $p(x_{k+1}|x_k)$ to obtain $\{x_{k+1}^i, \frac{1}{N}\} \sim p(x_{k+1}|y_k)$.
- **weighting:** particles are weighted by a likelihood function related to the correspondence from the model to the new observation. The new weights ω_{k+1}^i are normalized so that : $\sum_{i=1}^N \omega_{k+1}^i = 1$. It provides the new sample $\{x_{k+1}^i, \omega_{k+1}^i\} \sim p(x_{k+1}|y_{k+1})$.
- **estimation:** the new pose is approximated by:

$$x_{k+1} = \sum_{i=1}^N \omega_{k+1}^i x_{k+1}^i \quad (1)$$

The articulation of the person is often taken into consideration and modeled by a skeleton whose the rigid segments represent the body parts. To represent the volume occupied by the person, the skeleton comprises of a set of appropriately assembled geometric primitives [2–4]. The most usefull features chosen to describe the human class are the skin color [5], contour [6] and results of classifiers [7].

To reduce the computing time, Gonzales [5] splits its tracking to each sub-part of the body, Yang [8] simplifies the likelihood function with a hierarchical particle filter and Deutscher [2] reduces the required number of particles with an annealed particle filtering.

The movements of the skeleton can be constrained by interaction with objects in the environment [9, 10]. All poses of the skeleton are not possibles in practice. For example the head can not rotate over 360° . The sampling can be constrained by a projection on the feasible configuration space [3].

The human tracking in the top view is a seldom explored issue that has numerous applications in the video surveillance. Heath [11] estimates the 3D trajectories of salient feature points (primarily at the shoulders level) that he uses as the observation for the particle filtering. Canton-Ferrer [12] defines the exclusion zone for the blocking by an ellipsoide. For the tracking, he separates 3D blobs and used a particle filter where each particle represents a voxel of the blob. That estimates the centroid of the blob that model the person. These methods realize the tracking of the position of the person and not the gesture.

We propose a method that well-follows the pose of the arms that defines the gesture of the person.

In this paper, we describe a human gesture tracking from a top view by particle filtering. Relevant information is given by the depth provided by a Xtion Pro-Live camera. Rincón [13] realizes a first tracking to estimate the global location of the person and a second to recover the relative pose of the limbs. Similarly, but for the upper part of the body, we broke it the model up a 2D model representing the head and the shoulders and an 3D model representing the whole body (figure 1). The first one is relatively easy to obtain from the depth array. For the second one, the computational cost is reduced by constraining the space of possible poses with prior information given by the head and shoulders location.

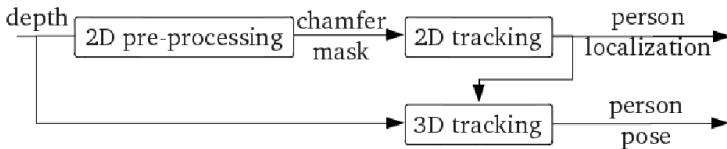


Fig. 1. Overview of the method: the 2D tracking estimates the location of the person while the 3D tracking estimates the pose

Our main contributions are first using a specially top view adapted model, secondly taking advantage of the depth signal to define likelihood function, thirdly decomposing the model in two parts so as to reduce the complexity of filtering and use simultaneously 2D and 3D models and finally introducing a confidence measure so as to detect the wrong pose estimations.

2 Particle Filter Implementation

To acquire the depth image, the Xtion Pro-live camera produced by Asus is installed at 2,9 m of the ground. This depth image provides a set of points in the 3D space (in grey in the figure 4) that represent the visible part of the surface of the person on a top view. Moreover, using 3D data allows introducing spatial constraints: a same object has various sizes in the 2D space according to its distance to the camera while it has always the same size in the 3D space. Two trackers using particle filter are presented in the following. For the initialisation step, the shoulders location is given by a detection process while the pose of the arms is fixed to be parallel to the body. A small number of frames are sufficient to find the right arm pose. We use a simple constant-vitesse dynamic model with a gaussian dispersion for the propagation.

2.1 The Head-Shoulders Model

As Micilotta [14], we use the Ω -like shape produced by the head and the shoulders. The top view of the head and the shoulders is modeled by two ellipses. Each ellipse defines 3 degrees of freedom.

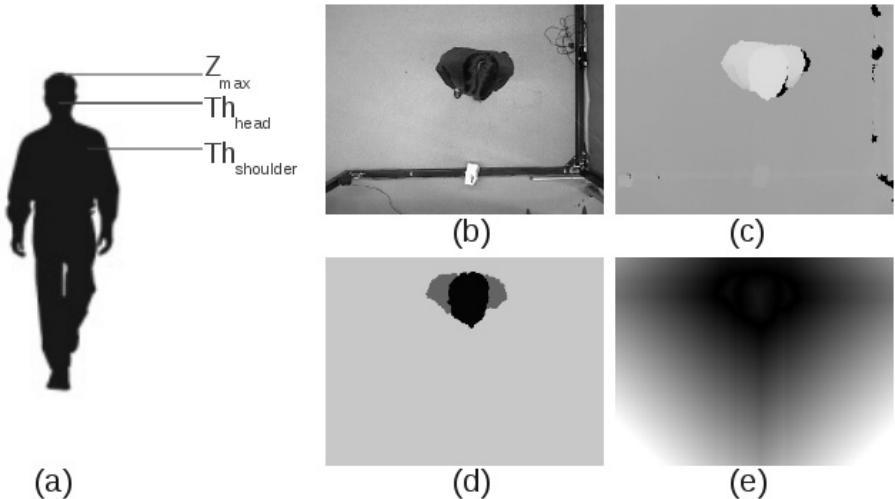


Fig. 2. The camera records simultaneously the color (b) and the depth (c) images. The depth image is thresholded (d) in the head and the shoulders levels (a). The ellipses of the model are fitted to the chamfer distance of this map (e).

The depth cue gives the distance of an element to the camera, thus it gives its distance to the ground. We threshold the depth image in two levels corresponding to the middle of the head and the end of the shoulders (figure 2(d)). The likelihood function is computed by the matching of the ellipses given by the particle with the chamfer distance of the thresholded depth image (figure 2(e)). The chamfer distance is robust to the person stoutness variation. Constraints are inserted in the propagation step in order to link the two ellipses.

2.2 The Complete 3D Model

Modelisation. This model is hardly constrained by the previous one. The 2D position of the shoulders gives the location and the orientation of the person in the scene. The 3D model determines the arms movement. We make the hypothesis that an arm has 5 degrees of freedom: 3 for the shoulder and 2 for the elbow. The skeleton of the model is also build and defined by a 5-dimensions state vector for each arm. The state space is limited by biomechanical knowledge about human motion. To represent the volume, geometrical primitives are added. Arms and forearms are modeled by truncated cylinders, torso by an elliptic cylinder and finally the hands by rectangular planes (figure 3).

Likelihood Function. As for the previous model, the depth cue is used. Nevertheless, not only the thresholded 2D representation is exploited. Indeed the depth variation is well-descriptive of arm. Thus a 3D chamfer distance is chosen.

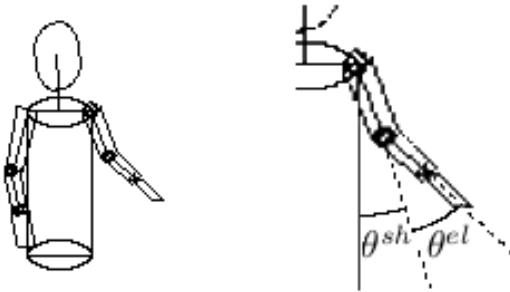


Fig. 3. The complete 3D model (left) is defined from the angles of the state vector (right)

Let Δ be the pixels of the foreground of the depth image excluding the head and the shoulders detected previously and \mathcal{M} be the 3D model given by a particle. The likelihood function related to particle i is defined by:

$$\omega_i = \text{average}_{p \in \Delta}(d_{3D}(p, \mathcal{M}^i)) \quad (2)$$

where d_{3D} is the shortest distance from a point to a 3D model.

3 Performances

We have simulated the behavior of customers in experimental conditions to evaluate the ability of our method to track a person. We have recorded two sequences S_1 et S_2 that are made of 450 frames ($>1\text{min}$) and 300 frames ($\approx 43\text{s}$) with various arm movements.

To qualitatively evaluate our results, we display in figure 4 the 3D model given by the tracking and projected on the color image and in the XZ and YZ planes. The pixels that can be seen in the depth image are represented in gray in the XZ and the YZ planes. We can also notice that the model is well-fitted to the person.

We use the depth cue as a ground truth to provide a quantitative evaluation. The pixels of the foreground of the depth image excluding the head and the shoulders (given by the 2D tracking) are shared between the two arms through the shoulders orientation. The evaluation measure we used is the average of the distance between the transposition in the 3D space of these pixels and the 3D model estimated by the tracking.

A high number of particles increases the accuracy of the tracking but it increases the time processing. A compromise may be found. The processing times are here obtained with a non-optimized C++ implementation running on a 3.1GHz processor. The figure 5 shows that there are no more meaningful improvement over 75 particles. Under this limit, it provides a real time processing.

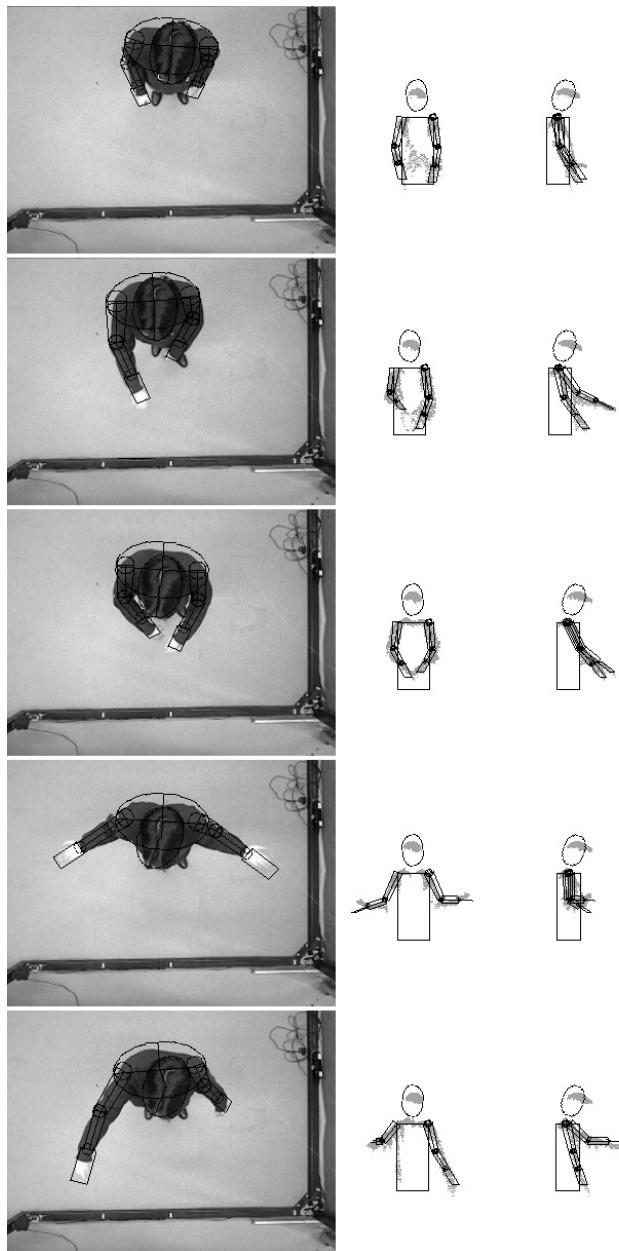


Fig. 4. On the left the model on the color image and on the right the model in the 3D space projected on XZ and YZ planes (the pixels in gray correspond to the points given by the depth image)

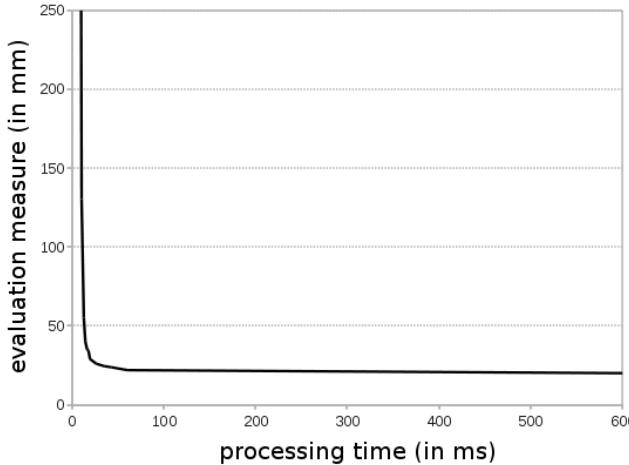


Fig. 5. Evolution of the tracking quality as a function of the processing time for the two sequences: when the number of particles increases, the accuracy of the tracking increases but the processing frequency decreases. Our process is real time.

The tracking is significantly degraded under 25 particles. With 75 particles on 912 evaluations of the arm in the sequence, we obtain an evaluation measure with an average of 25mm and a standard deviation of 5mm.

Sometimes, some particular poses are difficult to estimate. It is the case when the fist is held high (figure 6). To detect these false estimations we introduce a confidence measure. The skeleton of the estimation is projected on the 2D plane of the depth image. The confidence measure is the average distance from the pixels of the projected arms to the foreground. It is equal to 0 when the model is well-fitted to the observation and up otherwise. We have computed this confidence measure on the sequences \mathcal{S}_1 , \mathcal{S}_2 et \mathcal{S}_3 (figure 6). In case of wrong estimation, this criterium provides higher values that are easy to detect. The number of wrong estimations is small (1,52% for the three sequences) and they mostly persist for only one frame.

To evaluate the trajectories of the parts of the arm, we follow the 3D positions of the shoulders, the elbows and the wrists on a third sequence \mathcal{S}_3 (≈ 55 s) with large movement of the right arm. We use two ARTTRACK1 cameras and the software DTRACK to follow reflecting balls (figure 7(a)). The position of these captors are recorded simultaneously with the Xtion Pro-Live acquisition. We can see in figure 7(bcd) that the movement recorded by the captor is relatively closed to the ones computed by our method from the Xtion Pro-Live acquisition. The ART movements are most extensive because the captors can't be placed precisely on the articulation.

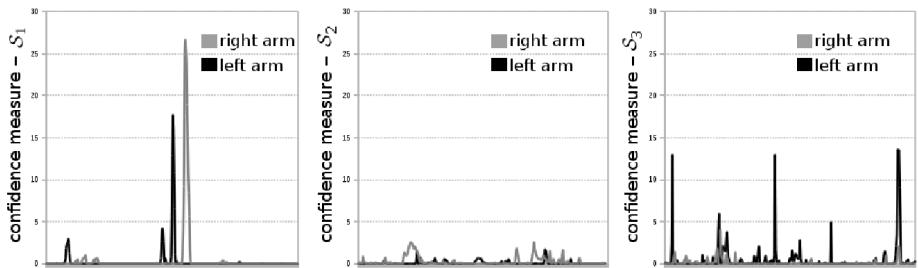


Fig. 6. The confidence measure shown for the two arms: the local maxima correspond to false estimations of the pose. They are relatively scarce and easy to detect.

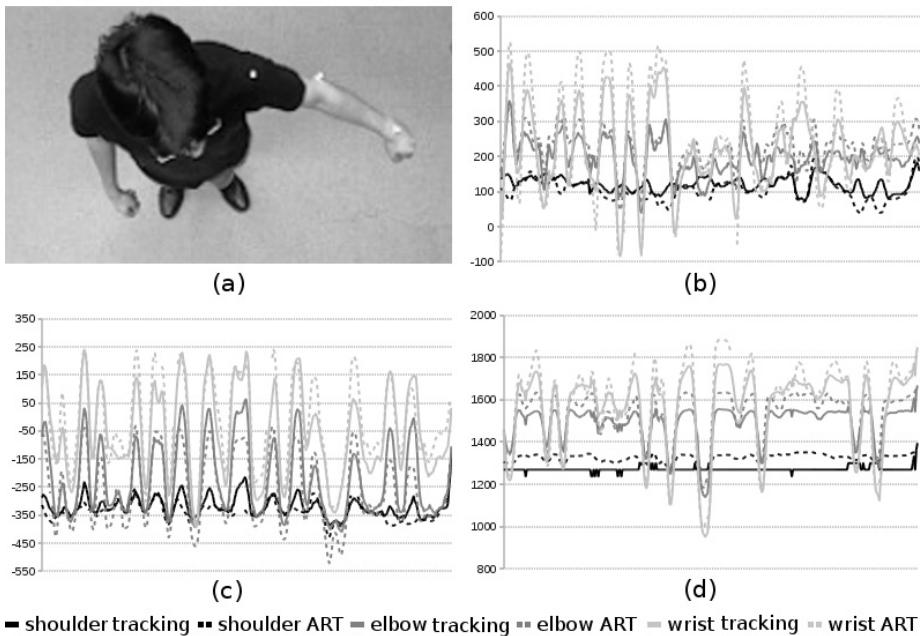


Fig. 7. Captors are used to follow the 3D positions of the shoulder, the elbow and the wrist of an arm (a). The trajectories of these captors (ART) are closed to the ones provided by our method (tracking) for the x (b), y (c) and z (d) coordinates.

4 Conclusion

In this paper we have proposed a new 3D tracking method that uses the particle filter to the particular case of the top view. A new Asus camera is used to take advantage of the depth cue. To do this, we have introduced a top view model that simultaneously used 2D and 3D fitting based on a chamfer distance. Moreover, for the behavior recognition context, a confidence measure is associated to each

frame so as to detect the possible wrong estimations. The process is efficient and real-time.

The tracking is the first step of the behavior analysis. Future works would use our tracking for action recognition. A camera pose estimation [15–17] could insert our work in a Augmented Reality context with a moving camera. Finally a coupled tracking and segmentation method would give more information for the following of the process.

References

1. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision* XXIX, 5–28 (1998)
2. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: *Computer Vision and Pattern Recognition* (2000)
3. Hauberg, S., Sommer, S., Pedersen, K.S.: Gaussian-like spatial priors for articulated tracking. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*, Part I. LNCS, vol. 6311, pp. 425–437. Springer, Heidelberg (2010)
4. Horaud, R., Niskanen, M., Dewaele, G., Boyer, E.: Human motion tracking by registering an articulated surface to 3d points and normals. *IEEE Transaction on Pattern Analysis and Machine Intelligence* XXXI, 158–163 (2009)
5. Gonzalez, M., Collet, C.: Robust body parts tracking using particle filter and dynamic template. In: *IEEE International Conference on Image Processing*, pp. 529–532 (2011)
6. Xia, L., Chen, C., Aggarwal, J.K.: Human detection using depth information by kinect. In: *International Workshop on Human Activity Understanding from 3D Data* (2011)
7. Kobayashi, Y., Sugimura, D., Sato, Y., Hirasawa, K., Suzuki, N., Kage, H., Sugimoto, A.: 3d head tracking using the particle filter with cascaded classifiers. In: *British Machine Vision Conference*, pp. 37–46 (2006)
8. Yang, C., Duraiswami, R., Davis, L.: Fast multiple object tracking via a hierarchical particle filter. In: *International Conference on Computer Vision*, pp. 212–219 (2005)
9. Kjellström, H., Krägic, D., Black, M.J.: Tracking people interacting with objects. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2010)
10. Oikonomidis, I., Kyriazis, N., Argyros, A.: Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: *IEEE International Conference on Computer Vision*, pp. 2088–2095 (2011)
11. Heath, K., Guibas, L.: Heath, k., and guibas, l.j. In: *ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–9 (2008)
12. Canton-Ferrer, C., Salvador, J., Casas, J.R., Pardàs, M.: Multi-person tracking strategies based on voxel analysis. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) *RT 2007 and CLEAR 2007*. LNCS, vol. 4625, pp. 91–103. Springer, Heidelberg (2008)
13. Del Rincón, J., Makris, D., Nebel, J.: Tracking human position and lower body parts using kalman and particle filters constrained by human biomechanics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 62, 26–37 (2011)

14. Micilotta, A., Bowden, R.: View-based location and tracking of body parts for visual interaction. In: British Machine Vision Conference, pp. 849–858 (2004)
15. Didier, J., Ababsa, F., Mallem, M.: Hybrid camera pose estimation combining square fiducials localisation technique and orthogonal iteration algorithm. International Journal of Image and Graphics 8, 169–188 (2008)
16. Ababsa, F., Mallem, M.: A robust circular fiducial detection technique and real-time 3d camera tracking. International Journal of Multimedia 3, 34–41 (2008)
17. Ababsa, F.: Robust extended kalman filtering for camera pose tracking using 2d to 3d lines correspondences. In: IEEE/ASME Conference on Advanced Intelligent Mechatronics, pp. 1834–1838 (2009)

Determination of Object Directions Using Optical Flow for Crowd Monitoring

Aravinda S. Rao¹, Jayavaradhan Gubbi¹, Slaven Marusic¹,
Andrew Maher², and Marimuthu Palaniswami¹

¹ ISSNIP, Department of Electrical and Electronic Engineering,
The University of Melbourne, Parkville, VIC - 3010, Australia
aravinda@student.unimelb.edu.au, {jgl,slaven,palani}@unimelb.edu.au
² ARUP, Melbourne, VIC - 3000, Australia
Andrew.Maher@arup.com

Abstract. Determination of object direction in a multi-camera tracking system is critical. The absence of object direction from other cameras pose challenges if the object is along the optical axis. The problem of determining object direction worsens further if the cameras in the existing infrastructure are improperly placed and are uncontrollable. To determine the direction of an object in such situations, three methods based on optical flow (OF) are presented. The first method uses centroids of optical flow vector magnitudes and Kalman filter for tracking and is suitable for less crowded scenarios. The second method uses geometric moments to evaluate the flow vector distribution and to ascertain the direction in case of crowded scenarios by partitioning the scene and then applying moments to individual partitions independently. The third method is appropriate for small-sized objects near vanishing points where global object motion is less. During surveillance, whether multi-object, single-object or crowded scenarios, the aforementioned methods are applicable accordingly. The results show that the object directions can be accurately inferred from three methods for different scenarios.

1 Introduction

Crowd tracking is an important application in computer vision. In a networked camera setting, within a camera sensor network, three scenarios of object identification and tracking are encountered: (1) overlapping field of view, (2) partially overlapping field of view, and (3) non-overlapping field of view. In case of partially overlapping and non-overlapping field of view, object tracking and determining directions are significantly critical for co-operative tracking. Determining the correct direction of motion is important and challenging, specifically, when fixed existing camera infrastructure is used. For instance, if we have an object approaching towards the camera or moving way from the camera along the optical axis, it is essential for us to determine the direction of the object. However, obtaining direction information in the presence of multiple objects and when the size of the objects are small becomes challenging.

Background subtraction [1] and optical flow (OF) analyses are the two main methods popularly used for extracting motion information from a region of interest [2]. While the background subtraction models use the variations of the pixels, OF uses irradiance constancy and smoothness in determining the pixel displacement [3]. Subtracting background model would help to provide region of movements, but not direction when object is small and along optical axis. The subtracted model would still highlight the same region without any additional directional information. The optical vectors provide magnitude and direction information along the X and Y axes of the image plane, but meager information along the Z (depth of field of view) axis. For instance, when a camera is installed along the corridors of a large venue, we often see people movement along the optical axis and near vanishing point. Moreover, the placement of the camera (with respect to height) is variable due to varying ceiling heights causing mismatched size of the objects as seen in different cameras.

Using optical flow, in [4], the apparent motion of the observer and the actual optical flow vectors were separated and were mapped using a rotating observer. In [5], one-dimensional optical flow vectors were queued for each direction and the queue that had the maximum positive value was considered as the moving direction. Shibata *et.al.* [6] have used the prominent direction indicated by the feasible vectors. In determining the direction of the object, all of these works inherently depend on the dominant vector directions. Others have proposed head-and-face detection [7], walking direction [8] and gait action [9], where the primary aim was to distinguish among different positional body angles.

Most of the CCTV systems will have vertical FoV (VFoV) of up to 45° from the ceiling such as overhead cameras [10] [11] and tilted [12]. The data that is being used in our work has VFoV up to 30° . Because of this vertical FoV, the objects at the far end of the perspective projection appear along the optical axis. In this paper, we focus on resolving the issues that arise when using motion information obtained from OF. We aim to determine direction of objects only from motion information. Three methods have been proposed to address the direction issue primarily using optical flow. The first method is applicable for situations where objects are clearly separated. The method uses magnitudes of the flow vectors, and their corresponding centroids to track the object direction. The second method uses geometric moments of optical flow distributions in a smaller search space and when the scene is cluttered to determine the collective direction of objects. The third method analyzes the directions obtained by flow vectors of the neighboring pixels of identified object region. The third method is suited for crowded and small-sized objects when the objects appear to be moving along optical axis near vanishing points where motion along X and Y axes are limited.

2 Methodology

Horn-Schunck OF method [3] based on brightness constancy is used in this work. The OF vector matrix O consisting of horizontal (x) and vertical (y) velocities

(Eq. 1) is used to calculate the magnitude (mag) and direction (dir) of the vectors as given by Eq. 2 and Eq. 3 respectively. Considering most of the surveillance cameras come with short focal length, and consequently wide angle of view, the imaging of the scene falls under the category of perspective projection. The magnification factor $m = \frac{f}{z}$, where z is the distance from the camera to the object point in the scene and f is the focal length of the camera [13]. As the distance between the object and the camera decreases (i.e. z decreases) along the optical axis, the magnification of the object increases and also the area (m^2), associated with it [13]. The OF pattern for a 3×3 object region moving along the optical axis is as shown in Fig. 1.

$$O := \{x + iy : x, y \in \mathbb{R}\} \in \mathbb{C}^{m \times n} \quad (1)$$

where $m, n \in \mathbb{R}$ and $i = \sqrt{-1}$.

$$\text{mag} := \{(x^2 + y^2)^{\frac{1}{2}}\} \in \mathbb{R}^{m \times n} \quad (2)$$

$$\text{dir} := \{\tan^{-1}(\frac{y}{x})\} \in \mathbb{R}^{m \times n} \quad (3)$$

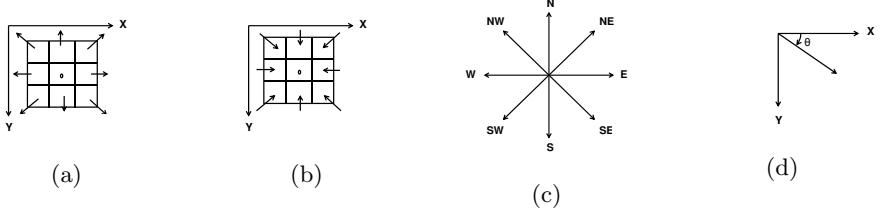


Fig. 1. Optical flow pattern for an object along the optical axis (a) approaching the camera, (b) moving away from the camera, (c) and (d) depict eight directions juxtaposed against X,Y directions.

2.1 Direction of an Object Using Flow Vector Magnitudes

Here we consider less-dense case for inferring directions from multiple objects. In case of less dense scenarios, background subtraction approach provides rich information about the scene by subtracting the background of a scene. In order to reduce the noise present in the video and to handle the crowded scenes, pre-processing, segmentation and morphological operations were applied to the raw video (24-bit RGB) by frame differencing combined with RGB channel operations $G^2 - B$ and $(G^2 - B)^{-1}$. All the pixels with OF magnitude greater than zero are labeled as 1 and others as 0. Next, the binarized matrix is relabeled by performing 8-connected-component analysis. Later, the centroids the relabeled matrices are stored for tracking. If an object is approaching towards the camera, then the centroid would move in positive y direction. Furthermore, the

Kalman filter was implemented for tracking the objects with centroid of each objects as the current position in the state space model. However, since there is no information about the object directions, a separate routine was maintained to derive directions from the updated equation of the Kalman filter. Based on the previous locations and trajectories for up to $n = 5$ time periods, we deduce object directions.

2.2 Direction of an Object Using Geometric Moments

In Section 2.1, the problem to determine directions was a global one, where we considered the entire scene. The drawback of this approach is that because of nonuniform illumination, shadows and noise, global approaches sacrifice certain information to maximize the efficiency. In order to overcome loss of information, we analyzed geometric moments to infer directions from the scene. The analysis was conducted for a single-object moving in cardinal and inter-cardinal directions are considered. Moments and functions of moments indicate invariant pattern features [14] and are separately calculated based on horizontal and vertical velocities obtained from OF. Table 2a (refer to the last page of the paper) summarizes the interpretations based on the real (horizontal) and imaginary (vertical) components of the flow vectors. Fig. 1 shows the convention of the x and y axes along with eight directions. The moments for real values are given by equations (4)–(7). Likewise, we also compute the geometrical moments (\bar{o}_I, σ_I, S_I and K_I) for imaginary values of the OF matrix. Our hypothesis is that the same results can be applied to multiple objects to obtain collective object directions by partitioning the scene into different windows based on centroids and apply the same analysis within each window corresponding to an object of interest. As a preliminary result, in this work we have presented the OF distributions for a single object case. From Table 2b it is evident that Kurtosis can be used to infer whether the object is approaching or moving away from the camera.

$$\text{Mean}_R = \bar{o}_R = \frac{1}{N} \sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [O_R(k, l)] \quad (4)$$

$$\text{Variance}_R = \sigma_R = \frac{1}{N} \times \sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [(O_R(k, l) - \bar{o}_R)^2] \quad (5)$$

$$\text{Skewness}_R = S_R = \frac{1}{N} \times \frac{\sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [(O_R(k, l) - \bar{o}_R)^3]}{\sigma_R^3} \quad (6)$$

$$\text{Kurtosis}_R = K_R = \frac{1}{N} \times \frac{\sum_{k=1}^{k=m} \sum_{l=1}^{l=n} [(O_R(k, l) - \bar{o}_R)^4]}{\sigma_R^4} \quad (7)$$

where $N = m \times n$.

2.3 Direction of an Object Based on Flow Directions

When the objects are small and near vanishing points, their size will not convey much information about direction because the size almost remains same. Because of this there will be ambiguity whether the object is moving away or approaching. As mentioned before, when the depth increases, the magnification factor decreases. Therefore, in contrast to the above two methods, the objective of this method is to extract information from the direction matrix (Eq. 3) for an object's region and make a decision as to whether the object is approaching or moving away from the camera especially near the vanishing points. We In order to determine the direction of the object, a template mask, T , is moved over the direction matrix with the center of the 3×3 matrix being the pixel under consideration. This pixel is assumed to be at the center of the circle as shown in Fig. 2. A score is assigned to determine as to how much the neighboring pixel's direction vector is indicating that it is pointing towards the center pixel is calculated. The scores are calculated from $n = 0$ to $n = 7$. This method is equivalent to finding convergence (sink) or divergence (source) of flow field in a given vector field.

$$T := \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} t_5 & t_6 & t_7 \\ t_4 & 0 & t_0 \\ t_3 & t_2 & t_1 \end{bmatrix} \quad (8)$$

A function $f : (\text{dir}(k, l), n) \mapsto c(k, l)$, maps OF vectors' direction matrix to a real number $[-1, 1]$ considering the neighborhood of $c(i, j)$. For a pixel at the center of T , the eight neighboring pixels are considered. In each direction as shown in Fig. 2, the $c(k, l)$ is computed as given by: $c(k, l) = \frac{1}{8} \sum_{n=0}^{n=7} I_n$, where where,

$$I_n = \begin{cases} +1, & |r| \leq (\pm \frac{\pi}{8}) \\ 0, & \pm \frac{\pi}{8} < |r| \leq \pm \frac{7\pi}{8} \\ -1, & |r| > (\pm \frac{7\pi}{8}) \end{cases} \quad (9)$$

and

$$r = (n \times \frac{\pi}{4}) - t_n \quad (10)$$

where n indicates the position of the current pixel being analyzed from center and t_n is equal to the $\text{dir}(k, l)$ along that direction in the neighborhood. The value $\frac{\pi}{4}$ is chosen as a threshold such that r of the center pixel determines whether the neighboring pixel is within $\pm \frac{\pi}{8}$ radians from the center and I_n assigns scores based of deviation from the center pixel's value. The score is incremented or decremented (by 1) based on whether the pattern agrees with the indented direction (within $\pm \frac{\pi}{8}$) or in the opposite direction, and left unchanged for any other directions. This is then summed for all the eight directions and normalized. For this score to apply, the magnitude of center pixel must be zero. Either Fig. 1-(a) or Fig. 1-(b) must be considered and scores must be applied. For instance, considering Fig. 1-(a), if an element $c(k, l)$ yields a score of 1, it implies that there is an object moving away from the camera and a score of -1 for $c(k, l)$ indicates that the object is approaching the camera.

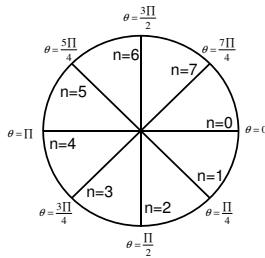


Fig. 2. Depicts the θ value in radians for n

3 Results and Discussion

All implementations were carried out in MATLAB 8.0 using Computer Vision System Toolbox on Windows XP-SP2, Intel i7 - 2600, running at 3.4 GHz on a 32-bit computer utilizing 512 MB ATI RadeonTM HD 5450 graphics card.

The result of determining the direction using the centroids is given in Fig. 3, where Fig. 3-(a) shows the centroid at location (368, 128), Fig. 3-(b) shows the centroid at location (370, 130) and Fig. 3-(c) at location (370, 132) for a video collected from a major sporting venue corridor camera. Kalman filter was used to keep track of the object and its location and the result for three different frames are shown in Fig. 3. By keeping track of centroid locations, we estimate the trajectory along optical axis.



Fig. 3. Tracking centroid of object using Kalman Filter for object id 30 - the value of index y is increasing by 2 as object is approaching the camera - video collected from a major sporting venue

For the second method, based on the rules in Table 2b, the features in Table 1 were extracted using the OF magnitude distributions of the scene. It is evident that kurtosis can be used to determine whether the object is approaching or moving away. Additionally, mean values of horizontal and vertical velocities provide movements in X and Y directions. Skewness can be used to measure the object's movements along diagonal directions. Fig. 4 shows two cases for a video that was filmed in our lab specifically for calculating the moments.

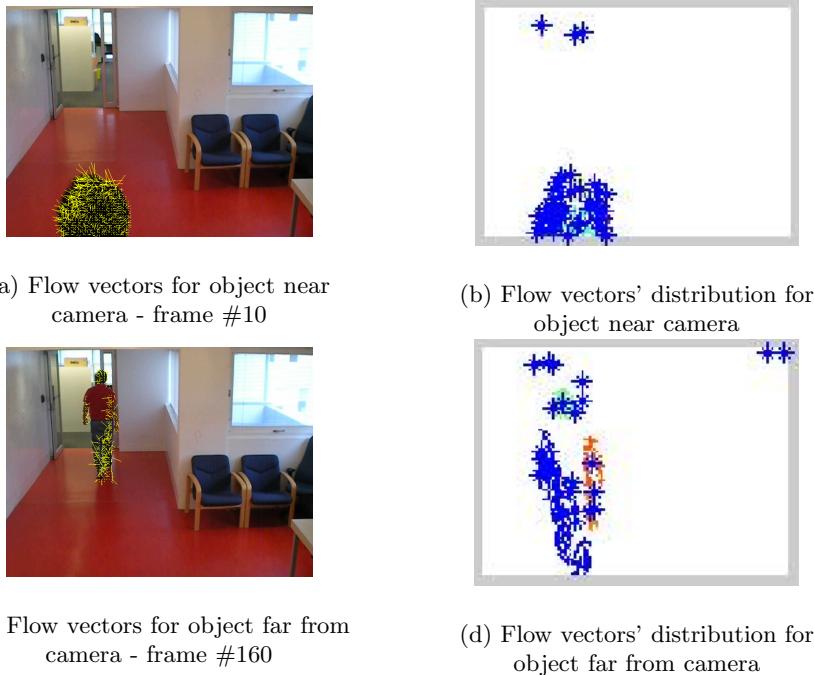


Fig. 4. Geometric moments are calculated based on OF distributions in (b) and (d)

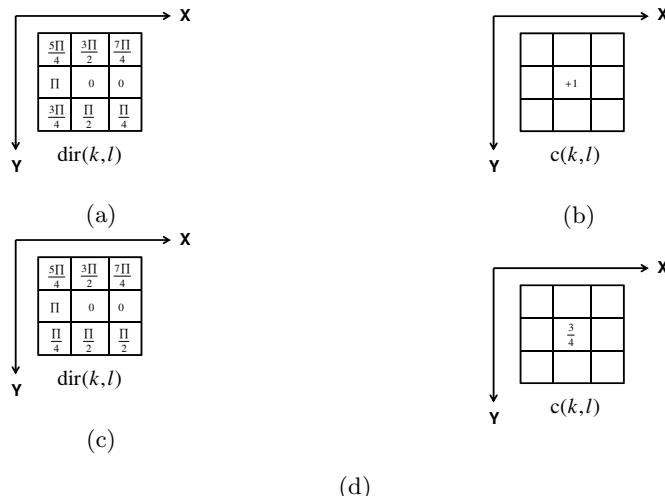


Fig. 5. (a) shows the ideal $\text{dir}(k,l)$ for an object approaching the camera and (b) the corresponding $c(i,j) = +1$. (c) shows the nonideal condition for an object approaching the camera and (d) the corresponding $c(k,l) = \frac{3}{4} < 1$.

Table 1. Features extracted from moments

Moments	Vector coefficients	Feature
Mean	Real, Imaginary	X,Y directions
Skewness	Real	Diagonal movements
Kurtosis	Real, Imaginary	Object closeness

The result for the third result was simulated and shows that if the flow vectors' direction for an object approaching the camera agrees with the Fig. 1-(a), then $c(k, l) = 1$. On the other hand, if the flow vectors' directions are not agreeing with either Fig. 1-(a) or Fig. 1(b), the method would detect this by providing score between 1 and -1 to the region being analyzed. Pragmatically, encountering the intended pattern exactly would be minimal. Hence, once can relax the tolerance in each directions while calculating the scores. For the same reason, we have shown only the simulated results. Furthermore, one can extend the 3×3 matrix to any $p \times p$ such that $p \leq \min(m, n)$.

In a surveillance system we often come across single and multiple objects. In case of multi-object scenarios, when the objects are larger in size (because of less depth along optical axis) the first method provides object direction along optical axis. Kalman filter is required to keep track of individual object's centroids and velocities to estimate the direction. Further during surveillance, we can separate multiple objects into single objects and then apply the geometric moments on flow vectors (this requires less processing cycles since the OF results are already available) to find out the directions. It is to be noted that when multiple objects are present, the OF directions do not convey meaningful information. In order to make sense of the OF vectors, the distribution of geometric moments are used. When the size of the objects becomes smaller, the first two methods would not provide accurate results. Therefore, we concentrate on the optical flow vectors of identified small region of the frame and apply the third method for direction information (diverging or converging).

4 Conclusion

Determination of object direction using optical flow along the optical axis was presented. Three methods were proposed to calculate the direction based on OF centroids, geometric moments, and direction flow pattern at any given instance. In terms of suitability, the first method is applicable for situations where objects are clearly separated. The method uses optical flow and background subtraction to obtain centroids. Further, the Kalman filter is used for tracking and a subroutine to deduce directions based on previous location trajectories. The second method is devised for obtaining directions when the scene is cluttered. In this case the scene is divided into partitions and geometric moments are calculated to infer collective group directions. The third method is suited for crowded and small-sized objects when the objects appear not to be moving along optical axis. The results show that object direction along the optical axis can be deduced from the three methods for different scenarios.

Table 2a. Interpretation of geometric moments for an object moving in eight directions

Moments	Vector coefficients	Absence of object (to N)	Forward (to S)	Backward (SW to NE)	Diagonal (SE to NW)	Object Near Camera	Object Near away from Camera
Mean	Real	0	constant	constant	positive	negative	N/A
	Imaginary	0	negative	positive	negative	negative	N/A
Variance	Real	0	nominal	nominal	nominal	N/A	N/A
	Imaginary	0	nominal	nominal	nominal	N/A	N/A
Skewness	Real	0	unchanged	unchanged	positive	negative	N/A
	Imaginary	0	unchanged	unchanged	N/A	N/A	N/A
Kurtosis	Real	0	Low to High	High to Low	High to Low	Low	High
	Imaginary	0					

Table 2b. Geometric moments calculated for an object near camera and moving away from camera as shown in Fig. 4

Moments	Vector coefficients	Object near camera	Minimum	Maximum	Object moving away from camera	Minimum	Maximum
Mean	Real	-0.331786	-0.3625	0	-0.003019	-0.3625	0.4703
	Imaginary	-0.059070	-0.0624	0	0.014716	-0.2593	0.0367
Variance	Real	18.215907	0	18.2159	1.048159	0	25.8190
	Imaginary	15.030494	0	15.0305	1.014135	0	21.7704
Skewness	Real	-4.220154	-10.8099	0	1.779557	-11.8754	17.9986
	Imaginary	-0.546849	-9.7723	0.3561	10.219373	-18.2828	10.2194
Kurtosis	Real	65.825336	0	332.0975	425.208734	0	1051.20
	Imaginary	65.519053	0	328.4689	551.674870	0	1209.00

References

1. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 246–252. IEEE (1999)
2. Lepisk, A.: The use of optic flow within background subtraction. Master's thesis. Numerisk analys och datalogi, NADA, Stockholm, Sweden (2005)
3. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
4. Kinoshita, K., Enokidani, M., Izumida, M., Murakami, K.: Tracking of a moving object using one-dimensional optical flow with a rotating observer. In: 9th International Conference on Control, Automation, Robotics and Vision, pp. 1–6. IEEE (2006)
5. Kinoshita, K., Murakami, K.: Moving object tracking via one-dimensional optical flow using queue. In: 10th International Conference on Control, Automation, Robotics and Vision, pp. 2326–2331. IEEE (2008)
6. Shibata, M., Makino, T., Ito, M.: Target distance measurement based on camera moving direction estimated with optical flow. In: 10th IEEE International Workshop on Advanced Motion Control, pp. 62–67. IEEE (2008)
7. Ishii, Y., Hongo, H., Yamamoto, K., Niwa, Y.: Real-time face and head detection using four directional features. In: Proceeding of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 403–408. IEEE (2004)
8. Lertniphonphan, K., Aramvith, S., Chalidabhongse, T.H.: Human action recognition using direction histograms of optical flow. In: 2011 11th International Symposium on Communications and Information Technologies, ISCIT, pp. 574–579. IEEE (2011)
9. Lijia, W., Songmin, J., Xiuzhi, L., Shuang, W.: Human gait recognition based on gait flow image considering walking direction. In: 2012 International Conference on Mechatronics and Automation, ICMA, pp. 1990–1995. IEEE (2012)
10. Barandiaran, J., Murguia, B., Boto, F.: Real-time people counting using multiple lines. In: Ninth International Workshop on Image Analysis for Multimedia Interactive Services, pp. 159–162. IEEE (2008)
11. Rizzon, L., Massari, N., Gottardi, M., Gasparini, L.: A low-power people counting system based on a vision sensor working on contrast. In: IEEE International Symposium on Circuits and Systems, pp. 786–786. IEEE (2009)
12. Garcia, J., Gardel, A., Bravo, I., Lazaro, J., Martinez, M., Rodriguez, D.: Directional people counter based on heads tracking. IEEE Transactions on Industrial Electronics PP, 1–10 (2012)
13. Horn, B.K.P.: Robot vision. MIT Press, Cambridge (1986)
14. Teh, C.H., Chin, R.T.: On image analysis by the methods of moments. IEEE Transactions on Pattern Analysis and Machine Intelligence 10, 496–513 (1988)

Evolutionary Techniques for Procedural Texture Automation

Alaa Eldin M. Ibrahim

University of Sharjah

Sharjah, UAE

amibrahim@sharjah.ac.ae

Abstract. We have developed a genetic algorithm approach for automatically generating procedural textures. Our system, known as GenShade, evaluates evolutionarily generated procedural textures by comparing their rendered images with single or multiple target images of real textures. It uses a multiresolution image querying metric to automatically prioritize parents for breeding. GenShade simulates several key factors in natural selection. It employs a multiple generation breeding population, a notion of gender, and the concept of aging to maintain diversity while providing many breeding opportunities to highly successful offspring. The approach is also especially efficient running in a multiple processor, multiple selection-strategy mode using multiple settings. This paper discusses and evaluates these Genetic Algorithm techniques.

1 Introduction

1.1 Texturing Models

A number of researchers have laid the foundation of procedural texturing [2]. The disadvantages of procedural texturing include: the limited types of textures that can be generated, the procedural texturing is a manual process and depends heavily on the experience of the shader writer, and the parameters of a texturing procedure are difficult to tune. In more existing techniques, storage-efficient procedures using basis functions [1] can create high quality 3D textures with no distortion and no discontinuity.

1.2 Genetic or Evolutionary Algorithms

There have been many uses of interactive evolutionary techniques in computer graphics. Todd and Latham [12] applied biomorph concepts to help generate computer sculptures made with constructive solid geometry techniques. Sims [9] has presented several applications of interactive evolution, using the evolutionary mechanisms of variation and selection to evolve complex equations and procedures used by procedural models for computer graphics.

Automatic texture generation systems [16] solve problems with interactive methods. In these systems, evaluations of image features of generated textures are automatically compared with those of target images. This relieves artists from evaluating each generating texture. Another approach [8] investigated the automatic synthesis of aesthetically pleasing images using genetic programming. These systems use a fitness function that encodes a model of aesthetics.

More recently, there have been work in procedural texture particles [4], using interactive evolution to discover camouflage patterns [7], and computational aesthetic evaluation [3].

It is apparent that there is still a need to automatically generate efficient procedural textures that perform well on complex 3D objects. Our system was designed to solve this problem.

2 GenShade System Architecture

2.1 Genome Representation

In our system, a shader is represented by a hierarchical directed acyclic graph of nodes that describe a RenderMan shader [14]. As shown in Figure 1, the nodes themselves are functions written in the RenderMan shader language.

2.2 Multiple Generation Population

Figure 2 shows the basic GenShade [17] generation process. In order to maintain a large and diverse population, avoid stagnation, and give ample opportunity to strong genetic material, GenShade uses a multiple-generation population of shaders [11][13]. This architecture allows us to preserve a large set of the best scored shaders across a range of generations, and means that we can experiment knowing that there is nothing to lose by producing a generation of "poor" shaders. In the worst case, the population will remain essentially as is. Our experiments have used a maximum population size between two and three hundred. Shaders are inserted into the population in sorted order, with higher scored shaders replacing lower scored ones when the population maximum is exceeded. The multiple-generation population allows parents and children to coexist in the same population, which models an important feature of natural evolution. Initially, the system retrieves shaders from a predefined database of shaders and inserts the best scored shaders into these populations. Shaders are inserted into male and female populations based on their image scoring. We call shaders that score high on illumination the "male" shaders and those that score high on chromaticity the "female" shaders. This organization is used in selection, which is described below.

Aging is simulated in the multiple-generation population, to allow successful shaders several opportunities to produce direct offspring, while assuring that no single shader continues to dominate the population forever. This provides a good balance between keeping strong genetic material in the pool and assuring diversity in the population. Each shader in the population is allowed to survive intact up to a certain

generation threshold σ . The default is ten generations. After the generation threshold is reached, an exponentially decreasing aging factor is applied to reduce a shaders' score and therefore its probability of being selected for breeding and to increase its probability of being removed from the population. α determines the rate of aging. The aging factor α is selected so that once the generation threshold is passed, the shader disappears quickly from the population, modeling another feature of natural evolution.

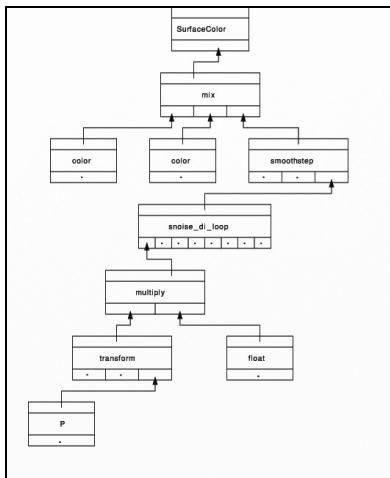


Fig. 1. Genome Representation

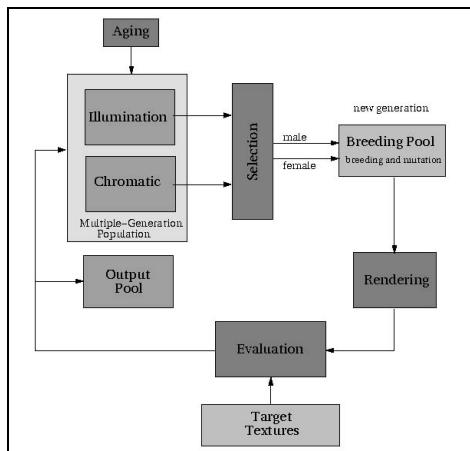


Fig. 2. GenShade Generation Process

2.3 Output Pool

In our system, a shader is represented by a hierarchical directed acyclic graph of nodes that describe a RenderMan shader [14]. The nodes themselves are functions written in the RenderMan shader language. Genshade handles nodes that contain geometry related functions very carefully to ensure that generated procedural textures work well over complex 3D surfaces.

2.4 Selection

"Parent" shaders are selected from the multiple-generation population in gender based pairs and bred to create new children for the ongoing population. Our previous experience working interactively with the system showed that breeding shaders whose images have chromatic content close to that of the target, with shaders whose images have illumination content close to that of the target is very useful in creating shaders that resemble the target. This notion has the appeal of being similar to the notions of "male" and "female" in real populations, where some of the standards of suitability for mating are highly gender determined. The selection procedure selects pairs of shaders from the male or illumination population for breeding with shaders in the female or

chromatic population based on their rank. GenShade uses a probabilistic selection procedure based on the gaussian distribution. There are two major selection strategies. The first is to focus on high scored shaders by using a small standard deviation. The second is to introduce diversity by using a high standard deviation. A selection strategy that we have found to be very useful is to alternate generations, where the first strategy is used in odd generations and the second in even ones.

2.5 Evaluation (Fitness Function)

The system evaluates images of textures in the ongoing population by comparing them to single or multiple target images. We use a multiresolution wavelet decompositions of the target image and database images. So, the score of each shader is how close its rendered image to the target image. We use the YIQ color system, with several unique considerations. First, to implement our notion of gender, we score each shader in three ways: illumination only by considering the Y component of the YIQ system, chromaticity only by considering the I and Q components, and overall by considering all three components. Second, we have provided the capability to compare generated images with multiple target images. To do this, the most significant common wavelets among the multiple target images are compared to those of generated images. This allows us to match the look of a group of related textures, rather than simply matching a particular image of that texture.

2.6 Phenome Generation

As we consider a RenderMan shader our genome, the corresponding phenome or offspring is a texture tile produced by the shader. To create a shader from a hierarchy, hierarchy nodes are traversed to create a text file formatted in the RenderMan shading language [14]. First the hierarchy is traversed to define all input variables. All un-linked input variables are output as instance variables, with predefined values. All linked variables are output as local variables. The hierarchy is then traversed again. Each linked output variable is visited and an assignment statement is output, whose left side is the name of the input variable in the parent node that the output variable is linked to, and whose right side is an expression that operates on the input variables in the current node. Variables can be of the types float, color, point, or string. The system provides conversion between types. This shader is then attached to a square tile and rendered via a RenderMan compliant renderer to produce a texture image.

2.7 Multiple Processor Algorithm Extension

Because of its organization, it is easy to configure GenShade to run in a multiple process mode, where each process contributes to the evolution of the multiple-generation population [5]. Besides increasing speed, this has the advantage that each process may use different generation parameters and selection strategies. Figure 3 shows GenShade running in multiple processes mode, with all processes updating the

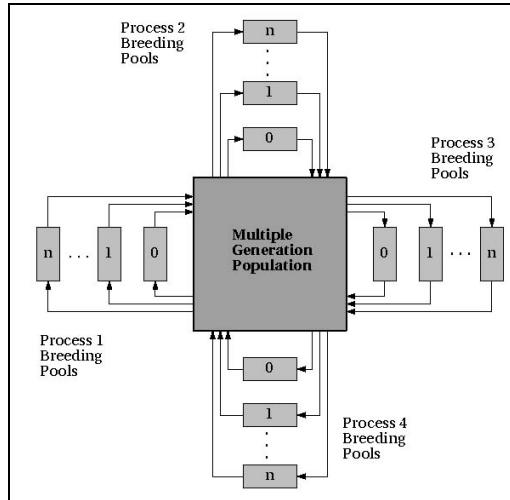


Fig. 3. Multiple Processes Mode

common multiple-generation population. Reproduction takes place in parallel, which models another feature of natural evolution. We have implemented this in communicating parallel processors sharing a common data store.

3 Experimental Results

In order to demonstrate the performance of GenShade modifications and extensions to the standard genetic algorithm, several experiments were conducted. The same generation parameters and target images were used in all of the experiments. However, in each successive experiment, an additional extension to the selection and breeding algorithms was used. Each experiment was run for 130 generations, with eighteen parent pairs (36 parent total) selected for breeding in each generation, with each pair, producing two children (36 children total).

The following experiment titles describe the extension:

1. Standard Genetic Algorithm
2. Multiple Generation Population
3. Gender Based Selection
4. Aging
5. Multiple Processes

Figure 4 shows the initial population of shaders that was drawn from the database and used in all experiments. Figures 5 through 9 show the results of experiments one through five. In each figure, the results of running for 80, and 130 generations consecutively are shown. Figures 10 and 11 show comparison fitness plots of the median and mean of scores of the best thirty-six shaders. In the genetic algorithm literature

plots such as this generally show the best solution over time. Here we plot mean and median rather than best score since the focus in this study is on the quality of the entire population and not only on the best shader. Statistics of running GenShade have shown that 80% of CPU time is taken by calling RenderMan to compile shaders and render the images, and 15% by the scoring system. The remaining 5% is taken by all other calculations. Therefore, running the system using a population of size 200 does not require much more time than using a population of size thirty-six. In standard genetic algorithms, usually multiple runs are made and ensemble statistics are presented across runs. Because running GenShade requires much more time per generation than a standard genetic algorithm problem and because of its intensive space requirements, it was only possible to make a single run per experiment.

3.1 Standard Genetic Algorithm

First experiment is the base case, using a standard genetic algorithm approach. In this experiment, shaders are chosen based on gaussian ranking selection. Population size was set to 200 shaders. The targets were four images of the tree bark. The following parameters were used:

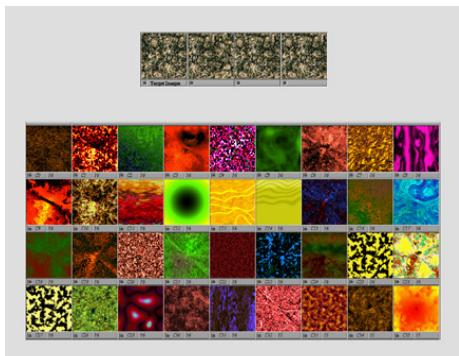
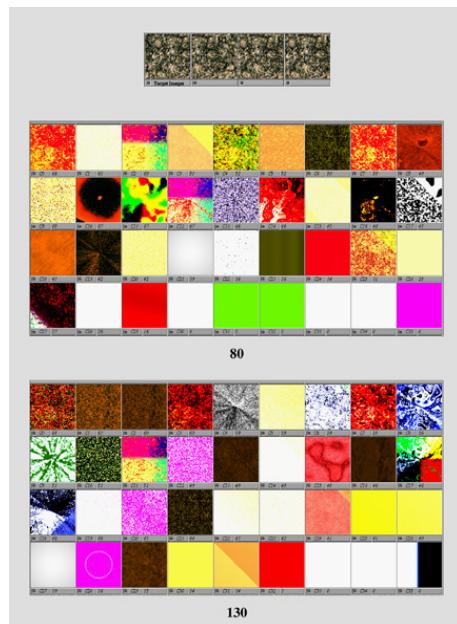
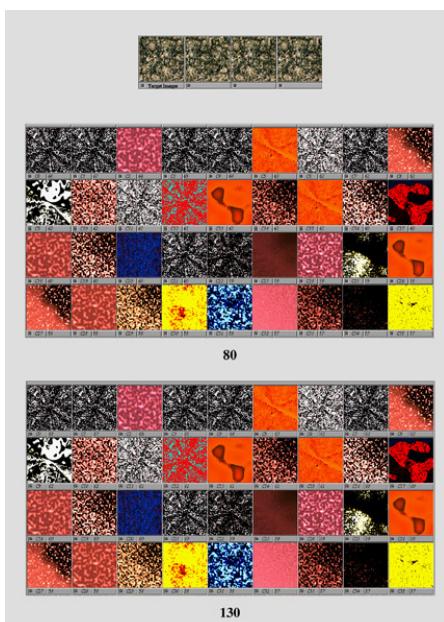
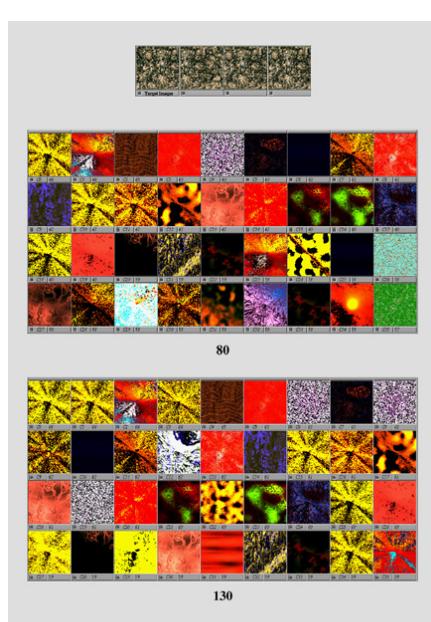
1. mutation value of 0.1: providing a probability of 0.1 to select an element to be mutated.
 2. shader level value of 3, to allow medium selection of subtrees to be selected for swapping when breeding shaders.
 3. gaussian standard deviation of 10.0, to select shaders within a certain rank range.
- The standard genetic algorithms uses an implementation of Elite [6][13], keeping the best four shaders in every generation. Figure 4 shows the results of using a standard genetic algorithm. It can be seen that many shaders do not look like the target images at all. In generation 80 results are even worse but this improves somehow by generation 130. Another major observance is premature convergence.

3.2 Multiple Generation Population

Figure 5 shows the results of extending the base experiment to use a multiple generation population of size 200 are shown. In the Figure, generations 80, and 130 are almost the same, again showing stagnation and lack of progress. But now at least we have more diversity. Figures 9 and 10 clearly show improvement, in the mean and median scores, over the standard genetic algorithm, and a multiple generation population of size thirty-six. We also see that progress is monotonic, with no drops in score. The flat curve between generations 50 and 130 verifies the stagnation mentioned above.

3.3 Gender Based Selection

Figure 6 shows the results of using a gender selection along with using multiple generation population of size 200. Figures 9 and 10 show that Gender Based Selection extension improves over multiple generation population after generations 60-70.

**Fig. 4.** Initial Population**Fig. 5.** Standard Genetic Algorithms**Fig. 6.** Multiple Generation Population**Fig. 7.** Gender

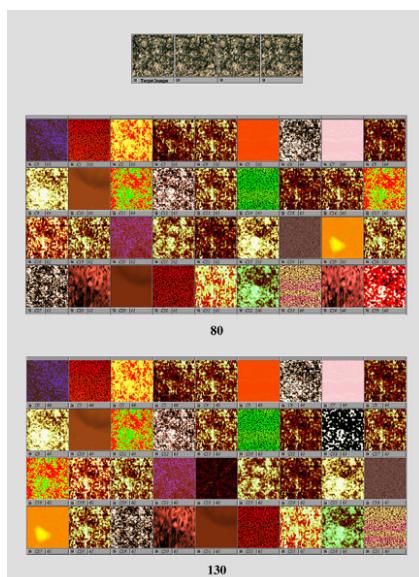


Fig. 8. Aging

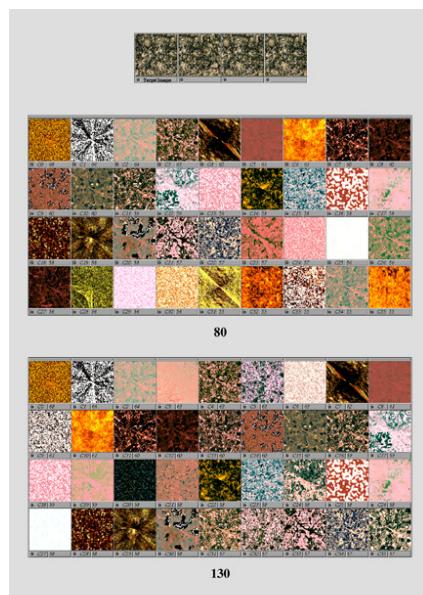


Fig. 9. Multiple Processes

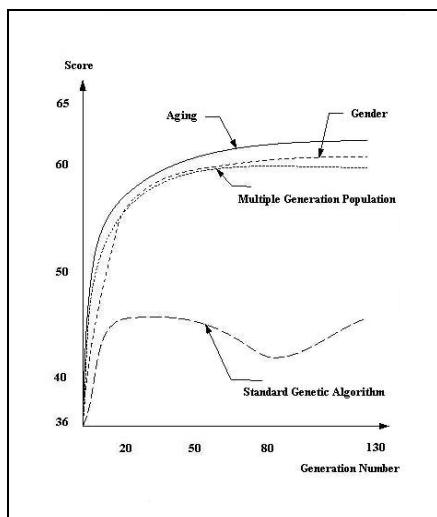


Fig. 10. Median score of best 36 Shaders

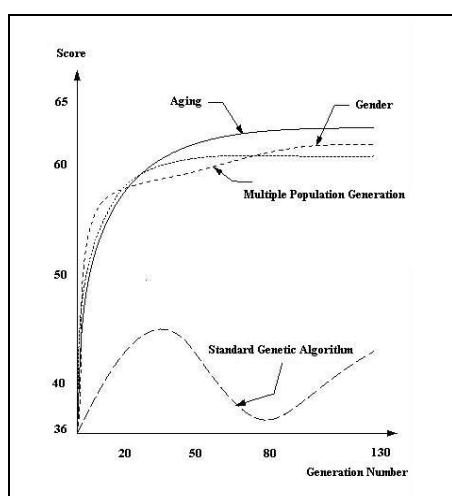


Fig. 9. Mean score of best 36 Shaders

It seems that, on the long run, the gender-based selection is a favorable extension to the multiple generation population. Figure 7 show results of applying aging to a multiple generation population with gender are shown. As shown in the Figure, the resulting shaders improve from one generation to the other. Generations 80 and 130 includes some of the closest shaders to the target images so far. Figures 9 and 10 show a larger jump over previous techniques.

3.4 Multiple Processes Method

Figure 8 shows the results of running six multiple processes that all use multiple generation population, gender, and aging extensions. Three of the processes used conservative selection strategies and the other three used risky strategies. As shown in the Figures, although the stagnation problem has been avoided, the resulting shaders do not look closer to the target than the previous aging extension. Figures 9 and 10 show that adding the multiple processes extension provides slight improvement, if any, over the previous method.

4 Conclusion

Using a multiple-generation population has shown to be an improvement over the standard genetic algorithm. It is clear that the gender extension has introduced diversity and avoided stagnation. It is obvious that introducing the concept of aging allowed successful shaders many chances to be selected but made sure that they were eventually removed so that they did not dominate the population. This has provided more diversity and formed the best combination of extensions so far. It seems that the multiple processes extension may not improve results but will generate them n times faster, on the average, where n is the number of processes used. It would be interesting to analyze what might make interesting shaders. Interesting shaders could be analyzed and certain patterns, e.g. subtrees, could be compared. Mutation could favor substituting these subtrees for hierarchy nodes. For example, there might be a subtree that creates wood rings, another that creates curls, and one that twists its subtree, and the artist could direct the system to favor using these subtrees.

Acknowledgment. I would like to thank Dr. Donald H. House, chairman of Division of Visual Computing, School of Computing at Clemson University for his generous contributions towards this research work

References

1. Cook, R., DeRose, T.: Wavelet Noise. In: Siggraph, pp. 803–811 (2005)
2. Ebert, D., Musgrave, F., Peachy, D., Perlin, K., Worley, S.: Texturing and Modeling, A Procedural Approach, 3rd edn. Morgan Kaufmann Publishers (2002)

3. Galanter, P.: Computational aesthetic evaluation: steps towards machine creativity. In: SIGGRAPH 2012 Courses (August 2012)
4. Gilet, G., Dischler, J.: Procedural texture particles. In: I3D 2010: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (2010)
5. Godart, C., Kruger, M.: A Genetic Algorithm with Parallel Steady-State Reproduction. In: Alliot, J., et al. (eds.) Artificial Evolution, European Conference. Springer (September 1995)
6. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and machine learning*. Addison-Wesley Publishing Co. (1989)
7. Reynolds, C.: Interactive Evolution of Camouflage. *Artificial Life* 17(2) (2011)
8. Ross, B., Ralph, W., Zong, H.: Evolutionary image synthesis using a model of aesthetics. In: Yen, G.G., Wang, L., Bonissone, P., Lucas, S.M. (eds.) *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp. 3832–3839 (2006)
9. Sims, K.: Artificial Evolution for Computer Graphics. *Computer Graphics* 25(4), 319–328 (1991)
10. Stollnitz, E.J., DeRose, T.D., Salesin, D.H.: Wavelets for computer graphics: A primer. Part I. *IEEE Computer Graphics and Applications* 15(3), 76–84 (1995)
11. Syswerda, G.: A Study of Reproduction in Generational and Steady-State Genetic Algorithms. In: *Foundations of Genetic Algorithms*, pp. 94–101. Morgan Kaufmann Publishers (1991)
12. Todd, S., Latham, W.: Mutator, a Subjective Human Interface for Evolution of Computer Sculptures. *IBM United Kingdom Scientific Centre Report* 248 (1991)
13. Toffolo, A., Benini, E.: Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evolutionary Computation* 11(2), 151–167 (2003)
14. Upstill, S.: *The RenderMan Companion, A Programmer's Guide to Realistic Computer Graphics*. Addison-Wesley Publishing Company (1989)
15. Whitley, D.: The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In: David Schaffer, J. (ed.) *Proc. of the Third Int. Conf. on Genetic Algorithms*, pp. 116–121. Morgan Kaufmann Publishers, San Mateo (1989)
16. Wiens, A., Ross, B.: Gentropy: Evolutionary 2D Texture Generation. *Computers and Graphics Journal* 26(1), 75–88 (2002)
17. GenShade's ShaderBank digital collection, keywords: Shaderbank collection (2011), <http://www.turbosquid.com>

Voxel-Based Harmonic Map for Voxel-Based Model Deformation/Manipulation

Tomoaki Nagaoka

Electromagnetic Compatibility Laboratory,
National Institute of Information and Communications Technology, Japan
nagaoka@nict.go.jp

Abstract. A three-dimensional discrete harmonic map based on voxel data, a voxel-based harmonic map, and its application to deformation of voxel-based computational human models is proposed. If two voxel regions and a map from one boundary region to another boundary region (voxelized boundary condition) are provided, the voxel values of one region can be mapped to another region by using the voxel-based harmonic map. The voxel model can be deformed and manipulated with the boundary of the voxel model, and the voxelized boundary condition is encoded to the triangular mesh for efficient manipulation. The algorithm for pose deformation using a real voxel-based human model is verified.

1 Introduction

Recently, voxel-based computational human models with anatomy have been widely used in various research fields such as medical and biological engineering, as shown in Figure 1 [1], [2], [3], [4]. These computational human models are generated on the basis of medical images such as X-ray computed tomography and magnetic resonance imaging (MRI), in which body shape and articulation are limited. Volume deformation techniques have been applied to the standard voxel-based human model to increase the variation of models.

A deformation method, called “volume refilling”, for a voxel-based computational human model with anatomy has been previously proposed [5]. Volume refilling deforms a model in two steps. The first step is deformation of the boundary, in which volume refilling deforms the subregion of a voxel model (i.e., region of interest of deformation; ROI-D). Here the term region is defined as a set voxels. This ROI-D is achieved by the deformation of the boundary of the ROI-D. Each boundary voxel of the deformed or target ROI-D corresponds to the boundary voxel of the non-deformed or source ROI-D. The corresponding voxel indices are set to the voxel values on the boundary of the target ROI-D. In this study, this set of boundary voxels is referred to as the voxelized boundary condition. The voxelized boundary condition defines the map from the deformed boundary to source boundary, and the corresponding voxel indices are referred to as the sampling coordinates. The second step is the interpolation of the map. The sampling coordinates for interior voxels are generated

as a harmonic function with the voxelized boundary condition. Finally, the source volume data is resampled with the generated sampling coordinates.

Because a voxel is difficult to manipulate, the voxelized boundary condition is encoded to the boundary triangular mesh. The triangular mesh is deformed to the desired shape by applying various mesh deformation methods using consumer software, and high-quality deformation is achieved in a sophisticated manner. To recover the voxel model, the deformed boundary mesh must be voxelized as a voxelized boundary condition. It is natural to ask the deformation algorithm to generate a voxel model is the same as the original model in case the boundary mesh is not deformed. This property of the algorithm is referred to as the identical recovery property (IRP). Unfortunately, the existing volume refilling algorithm did not make sufficient use of the IRP because the voxelization algorithm of the boundary triangular mesh was not assured to generate the same boundary voxels as the original boundary voxels.

In this study, an improved volume refilling algorithm that includes the IRP is proposed. A detailed description of the voxelized boundary condition generating an algorithm is provided, and the algorithm for pose deformation using a real voxel-based human model is verified.

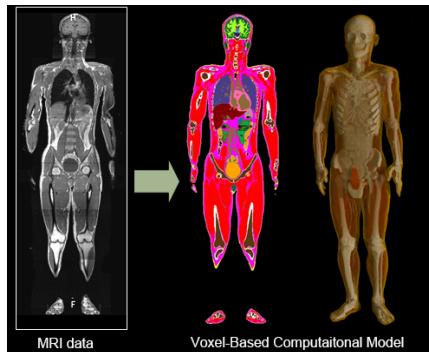


Fig. 1. Voxel-based computational human model with anatomy based on MRI data of the human body

2 Related Work

2.1 Harmonic Coordinate and Volumetric Parameterization

Coordinate generation has been studied for several decades. Volume refilling is based on the boundary fit method (BFM) [6]. The BFM was developed to map a rectangular solid computational region to a curvilinear region. Within the BFM, the boundary of the rectangular solid region is deformed to a curved boundary that fits to the surface of a curved object, such as the wing of an airplane. This region is represented as a regular grid. Thereafter, the BFM generates the coordinates of points in the interior region by solving an elliptic partial differential equation (PDE) with the boundary

condition defined by the boundary value. The generated coordinate functions are harmonic functions, i.e., the Laplacian of functions vanish within the interior region, and are called as boundary-fitted coordinates (BFC). Harmonic volumetric mapping is defined as a map from a solid object to another solid object that is harmonic [7]. In addition, it generates harmonic coordinate functions at the points in the interior region that are represented as a tetrahedral mesh. This coordinate generation is based on the method of fundamental solution (MFS). Theoretically, both BFC and harmonic volumetric mapping use harmonic coordinate functions for mapping; however, they consider different algorithmic approaches. Volume refilling also utilizes harmonic coordinates, but it is a voxel-based algorithm.

Spline-based parameterization of volume has been proposed in refs. [8], [9]. The surface mesh is used to define the volumetric region, and spline-based coordinates are used to map the volume data [8]. A polycube is used to define the domain of parameterization in ref. [9]. A spline is algebraic, and the properties of coordinates, such as the Jacobian of mapping, can be computed directly. However, the domain of spline-based coordinates is restricted to a curved cubic region. Therefore, when the target volumetric region is not topologically equivalent to the cubic region, decomposition of the region is required. Using harmonic coordinates as the solution of the boundary value problem is beneficial because it is free from this type of topological problem and is well suited to our application.

2.2 Voxel-Based Model Manipulation and Constraints

Free-form deformation (FFD) is one of the most widely used methods for volume deformation [10]. The voxel-based model in the volume dataset is covered with a set of rectangular solids that defines the region to be deformed. A rectangular solid is specified by eight control points, and voxel model deformation is achieved by manipulating these control points. FFD is a simple algorithm; however, it has some limitations. The setting and manipulation of control points is not directly connected to voxel model deformation. Cage deformation is another method of volume deformation [11]. Cage deformation uses a polyhedral hull to define the region to be deformed, and the coordinates of the points within the deformed cage are generated. In contrast to FFD, the cage generated by cage deformation is not limited in shape. Therefore, the cage can be set to achieve direct manipulation with denser control points compared with FFD.

If FFD or cage deformation is used, the deformed voxel model is constrained by coarse geometric objects, such as points and surfaces. However, if a finer constraint exists in the deformation, more direct manipulation becomes available. Boundary-constrained inverse consistent image registration (BICIR) is a non-rigid registration algorithm that is constrained by the boundary surface [12]. A voxel model is usually given as a volume dataset; however, there is some freedom or ambiguity in position within the volume dataset. It is natural to constrain a boundary such as body surface.

3 Detailed Description of Volume Refilling Algorithm

3.1 Voxel-Based Models

A voxel-based model is represented as volumetric data [13]. Volume refilling specifies that the region of interest (ROI) is a subset of volume data containing the voxel model. For example, the human body region is one ROI. The remaining voxels, which are referred to as background or exterior voxels, are ignored in the deformation process. The voxels in the ROI whose adjacent voxels set contains a background voxel are called boundary voxels.

A voxel is basically not a geometric object. However, voxel information is encoded into the triangular mesh, and the coordinates of a voxel model need to be defined. A single voxel, whose indexes are (i,j,k) , hereafter referred to as (i,j,k) -voxel in this paper, is regarded as a cubic geometry with eight vertices. The position coordinates of (i,j,k) -voxel are (i,j,k) , $(i+1,j,k)$, $(i,j+1,k)$, $(i,j,k+1)$, $(i+1,j+1,k)$, $(i+1,j,k+1)$, $(i,j+1,k+1)$, and $(i+1,j+1,k+1)$ (Fig. 2).

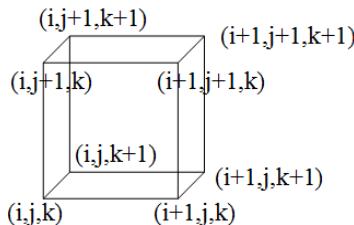


Fig. 2. Geometry of (i,j,k) -voxel

A triple, such as (i,j,k) , is often used to represent different targets such as position coordinates, texture coordinates, and the voxel indices. We denote the triple with the label “pos” for position coordinates, “tex” for texture coordinates, and “ind” for voxel indices when disambiguation is required.

3.2 Boundary Triangular Mesh and Voxelized Boundary Condition

Volume refilling deforms and manipulates a voxel model with the boundary triangular mesh. The deformed triangular mesh is reconverted using the voxelized boundary condition.

Voxelized Boundary Condition. Volume refilling computes harmonic mapping as a solution of a Dirichlet boundary problem. Elliptic PDEs for coordinate functions in three dimensions are solved using a finite difference scheme. In this numerical problem, the voxel version of a boundary condition is used. The voxelized boundary condition consists of boundary voxels and sampling coordinates at each boundary voxel. In the volume refilling algorithm, the boundary voxels of the original voxel model are encoded to the triangular mesh and the deformed mesh is decoded to the

voxelized boundary condition. A set of boundary voxels is required to form a close boundary region, which divides volume data into interior and exterior regions. A voxelized boundary is defined as follows:

- 1) The six-neighborhood of the interior voxel does not contain any exterior voxels.
- 2) The six-neighborhood of the exterior voxel does not contain any interior voxels.

This set of conditions is required for the finite difference scheme, which is the computational scheme later used to solve a boundary value problem.

In the special case in which the triangular mesh is not deformed, the reconstructed voxelized boundary must be identical to the original voxelized boundary. This property is referred to as the IRP of the voxelized boundary.

Generating the Boundary Triangular Mesh. The triangular mesh is generated on the boundary voxel face that is shared with the adjacent exterior voxel. The positions of the vertices of the triangle are the same as those of the cubic geometry described above. For example, if the (i,j,k) -voxel is the boundary voxel and the $(i-1,j,k)$ -voxel is the exterior, then the face with the four vertices of position coordinates (i,j,k) , $(i,j+1,k)$, $(i,j+1, k+1)$, and $(i,j,k+1)$ is triangulated into two triangles (Fig. 3, face rendered in striped-pattern). These two triangles are related to the (i,j,k) -voxel by adding the indices of the voxel as texture coordinates.

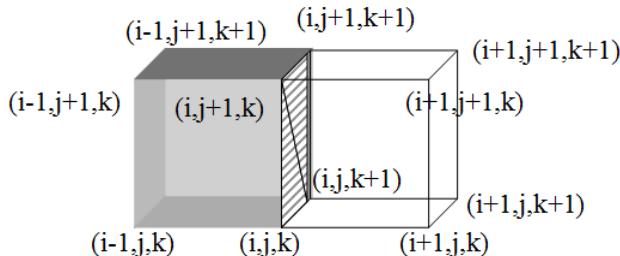


Fig. 3. The triangles generated at the boundary of a voxel. The voxel, illustrated as a shaded cube, shows the boundary voxel and the exterior voxel illustrated as a wire frame cube shows the exterior. Two striped triangles are generated as boundary voxel triangles.

For each non-exterior voxel, six adjacent voxels are tested to determine if they are exterior voxels. If an adjacent voxel is an exterior voxel, then two triangles are generated. One triangle has vertices with position coordinates (i,j,k) , $(i,j+1,k)$, and $(i,j+1,k+1)$, and the texture coordinates are the same as the position coordinates. Another triangle is defined in the same manner. These two triangles are related to the (i,j,k) -voxel through texture coordinates. The algorithm for computing the voxel index from the three distinct texture coordinates of the triangle will be described in the next section. Note that the mesh is closed and is used to identify the deformed voxel model region.

Reconstruction of Voxelized Boundary. The boundary triangular mesh is voxelized to reconstruct the voxelized boundary condition. In our implementation, the voxelized boundary condition consists of two volume datasets: a voxel identification volume and sampling coordinate volume. The voxel identification volume holds information about the type of each voxel, such as internal, boundary, or background voxels. The sampling coordinate volume holds a three-dimensional vector-valued voxel, which addresses the voxel in the source volume.

To convert the deformed triangular mesh to boundary voxels, each voxel is tested to determine if it a boundary voxel. Our intersection test algorithm is based on the distance between the voxel center, $\text{pos}(i+0.5, j+0.5, k+0.5)$, and the triangle. The test passes if one of the following conditions holds:

- 1) The distance is less than 0.5 (half of the voxel size)
- 2) The distance is equal to 0.5 and the voxel center resides in the interior side of the plane of the triangle, where the interior side is defined as the normal vector of the plane.

For each identified boundary voxel, the sampling coordinates must be computed from the texture coordinates at the triangle vertices. The goal of the voxel index computation is to identify the voxel index (i, j, k) from the texture coordinate of the triangles. The basic idea of computation takes the element-wise minimum of the texture coordinate.

For "type A" triangles illustrated in the left cube in Fig.4 we obtain (i, j, k) by taking the element-wise minimum values of the texture coordinates. For example, for the lower front "type A" triangle with $\text{tex}(i, j, k)$, $\text{tex}(i+1, j, k)$, and $\text{tex}(i+1, j+1, k)$, the element-wise minimum is equal to $(\min(i, i+1, i+1), \min(j, j+1), \min(k, k, k)) = (i, j, k)$, which is the desired result.

For "type B" triangles illustrated in the right cube in Fig.4, taking the element-wise minimum is insufficient. For the upper front triangle with $\text{tex}(i, j+1, k)$, $\text{tex}(i+1, j+1, k)$, and $\text{tex}(i+1, j+1, k+1)$, the element-wise minimum results in $(i, j+1, k)$, where the expected values are (i, j, k) . To adjust the result, we must subtract the v-component with 1 from the result, or subtract the vector $(0, 1, 0)$. Fortunately, the vector $(0, 1, 0)$ can be found as the cross product of two vectors: $\text{tex}(i+1, j+1, k) - \text{tex}(i, j+1, k) = (1, 0, 0)$ and $\text{tex}(i+1, j+1, k+1) - \text{tex}(i, j+1, k) = (1, 0, 1)$. By computing the cross product of two vectors, $(1, 0, 0)$ and $(1, 0, 1)$, we obtain the vector $(0, 1, 0)$. The texture coordinates are equal to the position coordinates of the non-deformed mesh. Therefore, the cross product represents the normal vector of the non-deformed triangle.

The normal vector of the type A triangles has the positive value components $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. However, the normal vector of the type B triangles has the negative value components $(-1, 0, 0)$, $(0, -1, 0)$, and $(0, 0, -1)$. Therefore, the triangle type can be identified by its normal vector.

The algorithm for computing the voxel index is as follows:

- 1) Take the element-wise minimum value
- 2) Compute the face normal vector
- 3) If the triangle is type B, then subtract the normal vector from the result of (1).

If multiple triangles intersect the voxel, then the nearest triangle is selected and its texture coordinates are associated with the voxel.

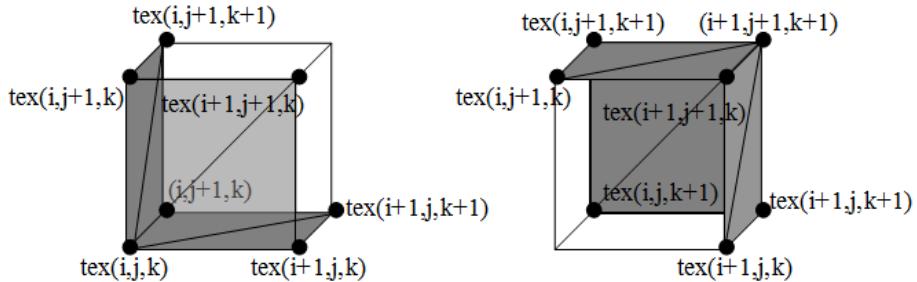


Fig. 4. Two type of possible boundary mesh face. The triangles belongs to the face that contains the vertex with $\text{tex}(i,j,k)$ are identified as "type A" in the left figure. The triangles that do not contain the vertex with $\text{tex}(i,j,k)$ are identified as "type B" in the right figure.

3.3 Harmonic Coordinate Generation and Identical Recovery Property

The voxel-based boundary condition is used to define the boundary value problem. The sampling coordinates of each interior voxel are interpolated from the voxel-based boundary condition generated at the last step using the element-wise harmonic function. Each set of sampling coordinates are a triplet of coordinate elements on the X, Y, and Z axes. The harmonic function vanishes by an action of the Laplacian. A function f is harmonic if and only if the following equation holds:

$$\begin{aligned} \text{Laplacian } L &= \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2, \\ Lf(x, y, z) &= 0. \end{aligned} \quad (1)$$

In the discrete setting, the Laplacian is computed using a finite difference scheme. For each X, Y, and Z axis directions, the coordinate function is computed. From the definition of an interior voxel, this finite difference computation addresses only the boundary or interior voxels. The each element of sampling coordinate at the interior voxel is computed as the solution of a Dirichlet boundary value problem. By resampling the source voxel model using the generated sampling coordinates, the deformed voxel model is generated.

If the boundary triangular mesh is not deformed, then the resampled voxels using the generated harmonic coordinate match the original voxel model. Hence, the voxelized boundary condition matches the original. Because the original linear sampling coordinates are harmonic, they are solutions for the Dirichlet boundary value problem. Therefore, the generated coordinates sample the voxels that are equal to the original voxels.

4 Experimental Results

In this section, the pose deformation of a voxel-based model using the proposed method is examined. Fig. 5 shows the example of pose deformation of a real voxel-based human model with a resolution of 2 mm segmented into 51 different tissue types [4]. The boundary triangular mesh is extracted from the source model. The triangular mesh is loaded into Autodesk Maya 2013, and a skeleton is set up. The skeleton subspace deformation [14], [15] technique is applied to the mesh. In the example, a walk pose is set up by manipulating the skeleton. The deformed mesh is voxelized to the voxelized boundary condition. The harmonic coordinates for the interior voxels are generated with the boundary condition. The resulting deformed voxel model is generated by mapping the source voxel model onto the interior and boundary voxels. From the figure, it is clear that the deformed models are deformed smoothly while maintaining the continuity of the internal tissues.

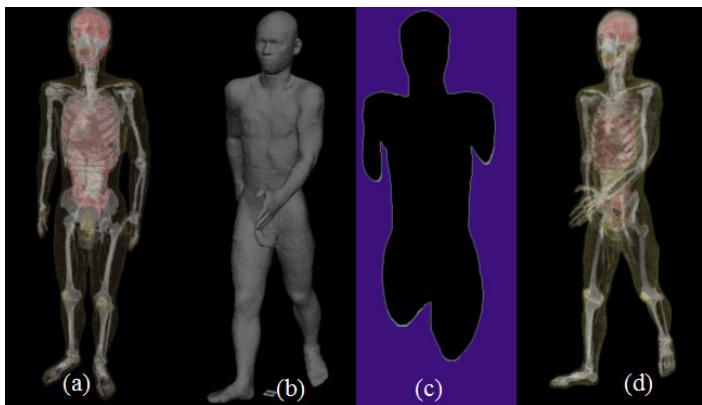


Fig. 5. Pose deformation task flow. (a) volume rendered the source male voxel model (b) the deformed triangular mesh (c) a slice of the voxelized boundary condition of the deformed boundary triangular mesh. (d) the resulting deformed voxel model.

Fig. 6 shows the difference between the voxel counts of tissues from the source model and those from the pose deformed model. The graph shows the number of source and deformed voxels for each tissue, accompanied by the percentage of per tissue difference. The difference in total voxel counts of these models is within 0.3%. For the large tissues (first eight tissues) that accounted for approximately 90% of the whole body voxels in the original data, the greatest difference is 2.64%. On the other hand, for small tissues, the greatest difference is larger than that of the large tissues (-8.57%). Because the nearest neighbor sampling algorithm is used in our mapping, sampling errors may occur for relatively small tissues. Variation in tissues for the pose deformed model is reasonable for numerical simulation in medical and biological engineering.

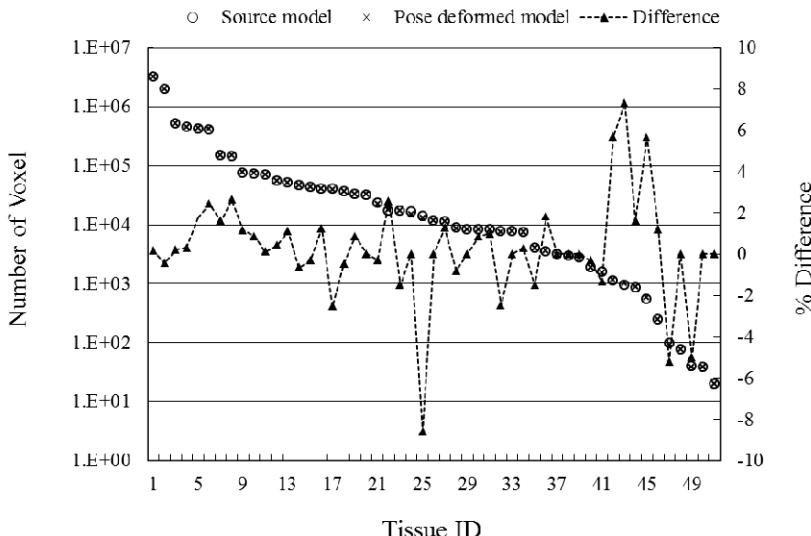


Fig. 6. Difference between the source and pose deformed models. The models consist of 51 tissues. The tissues are sorted by voxel count

5 Conclusion

A deformation and manipulation method based on voxel-based harmonic mapping for a voxel-based model was presented. The proposed algorithm encodes mapping information into the boundary triangular mesh, and deformation mapping is generated through voxelizing the boundary mesh followed by harmonic coordinate generation. Hence, the algorithm relies strongly on the boundary mesh voxelization quality. The algorithm can reconvert the boundary triangular mesh to the voxelized boundary condition perfectly in cases where the mesh is non-deformed, leading to identical recovery of the original voxel model.

A pose deformation of a voxel-based human model using our proposed method was attempted and examined. From the experiment, it was found that the pose deformed models are deformed smoothly while maintaining the continuity of the internal tissues, and variations in the internal tissue of the deformed model are reasonable.

Acknowledgments. The author is grateful to Mr. Makoto Matsumura of Adelic Research LCC. This work was partly supported by the Japan Society for the Promotion of Science KAKENHI Grand-in-Aid for Young Scientists (B).

References

1. Xu, X.G., Eckerman, K.F. (eds.): *A Handbook of Anatomical Models for Radiation Dosimetry*. Taylor Francis (2009)
2. Anzai, D., Aoyama, S., Wang, J.: Impact of propagation characteristics on RSSI-based localization for 400 MHz MICS band implant body area networks. In: Proceedings of the 6th International Symposium on Medical Information and Communication Technology (ISMICT 2012), pp. 1–4 (2012)

3. Collins, C.M., Wang, Z.: Calculation of radiofrequency electromagnetic fields and their effects in MRI of human subjects. *Magnetic Resonance in Medicine* 65, 1470–1482 (2011)
4. Nagaoka, T., Watanabe, S., Sakurai, K., Kunieda, E., Watanabe, S., Taki, M., Yamanaka, Y.: Development of realistic high-resolution whole-body voxel models of Japanese adult and females of average height and weight, and application of models to radio-frequency electromagnetic-field dosimetry. *Phys. Med. Biol.* 49, 1–15 (2004)
5. Nagaoka, T., Watanabe, S.: Voxel-based variable posture models of human anatomy. *Proceedings of the IEEE* 97, 2015–2025 (2009)
6. Thompson, J.F. (ed.): *Numerical Grid Generation*. North-Holland, Amsterdam (1982)
7. Li, X., Guo, X., Wang, W., He, Y., Gu, X., Qin, H.: Harmonic volumetric mapping for solid modeling applications. In: *Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling (SPM 2007)*, pp. 109–120 (2007)
8. Li, B., Li, X., Wang, K., Qin, H.: Generalized PolyCube Trivariate Splines. In: *Proceedings of the 2010 Shape Modeling International Conference (SMI 2010)*, pp. 261–265 (2010)
9. Li, B., Qin, H.: Feature-Aware Reconstruction of Volume Data via Trivariate Splines. In: *Proceedings of the 19th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2011)*, Kaohsiung, Taiwan, September 21-23 (2011)
10. Nagaoka, T., Watanabe, S.: Postured voxel based human models for electromagnetic dosimetry. *Phys. Med. Biol.* 53, 7047–7061 (2008)
11. Faraj, N., Thiery, J.M., Boubekeur, T.: VoxMorph: 3-Scale Freeform Deformation of Large Voxel Grids. *Computer & Graphics* 36, 562–568 (2012)
12. Kumar, D., Geng, X., Hoffman, E.A., Christensen, G.E.: BICIR: Boundary-Constrained Inverse Consistent Image Registration Using WEB-Splines. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW 2006)*, p. 68 (2006)
13. Kaufman, A.E.: Voxels as a Computational Representation of Geometry. In: *The Computational Representation of Geometry*. In: *SIGGRAPH 1994 Course Notes* (1994)
14. Magnenat-Thalmann, N., Laperrière, R., Thalmann, D.: Joint-dependent local deformations for hand animation and object grasping. In: *Proceedings on Graphics Interface 1988*, pp. 26–33. Canadian Information Processing Society, Toronto (1989)
15. Lewis, J.P., Cordner, M., Fong, N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (Siggraph 2000)*, pp. 165–172 (2000)

A Novel Approach to Retrieval of Similar Patterns in Biological Images

Andrzej Sluzek

Khalifa University, Abu Dhabi

Abstract. Novel descriptors of keypoints are proposed for matching (primarily) biological images. The descriptors incorporate characteristics of limited-size neighborhoods of keypoints. Descriptors are quantized into small vocabularies representing photometry of images (SIFT words) and geometry of their neighborhoods, so that significant distortions can be tolerated. In order to keep *precision* at a high level, Harris-Affine and Hessian-Affine detectors are independently applied. The retrieval results are accepted only if confirmed by both techniques. Using several test datasets, we preliminarily show that the method can retrieve semantically meaningful data from unknown and unpredictable images without any training or supervision. Low computational complexity of the method makes it a good candidate for scalable analysis of biological (e.g. zoological or botanical) visual databases.

1 Introduction

Visual inspection of random collections of images is one of the most tedious tasks in biological and biomedical practice. Algorithms performing automatic analysis of images are continuously developed for selected applications, where the objective is to identify well-defined objects/phenomena, e.g. cell counting, MRI segmentation, etc. However, in case of vaguely defined problems (e.g. searching for similarly looking components in images of plants, detection for similar phenomena in microscopic images of bacteria, etc.) images might be too diversified or too unpredictable for such specialized algorithms. The tasks are particularly difficult if the correspondence between semantics of image contents and their pictorial representations is not straightforward.

A prospective solution for the above challenges is to apply *data mining*, which in the visual domain is often referred to as *content-based visual information retrieval* (CBVIR).

Reported CBVIR applications generally focus on three areas, i.e.

- i. Retrieval of near-duplicate images (including partial near-duplicates, e.g. the same objects).
- ii. Retrieval of (semantically) the same category objects.
- iii. Retrieval of (semantically) the same category scene.

In (i), the expected result is a collection of image pairs depicting physically identical scenes or objects (subject to deformations caused by the viewpoint change,

camera quality, visibility conditions, partial occlusions, etc.). No training or supervision (regarding image semantics) is usually required. In (ii) and (iii), however, training is indispensable to generalize visual characteristics of semantic categories/objects, because individual images from the same category may be (visually) very different. In all the above areas, keypoint-based approaches are one the most fundamental tool (in particular *visual words*, BoW, and other word-based techniques, e.g. semantic topics built upon visual words).

Unfortunately, many types of biological images do not fit any of these areas. First, biological objects (even if considered identical) usually have diversified forms (and, subsequently, their images are only approximately near-duplicate). Secondly, in many problems the numbers and types of the semantic categories are not specified (e.g. the contents of images might not be fully classified).

The objective of this paper is to partially fill the gap between (i) and (ii)/(iii). We propose a scheme (based on the general concept of keypoint description and matching) to preliminarily identify images (e.g. biological images) which may contain meaningfully similar contents in spite of a wide range of deformations within these contents. No training (or pre-existing knowledge about the images) is assumed so that the scheme can quickly switch from one application to another. Moreover, the scheme is computationally efficient; we use a novel affine-invariant keypoint description where individual matches indicate the presence of visually similar fragments. The main difference, compared to typical descriptors like SIFT or SURF, is that our descriptions represent not only keypoints but also their limited-size neighborhoods.

The method is proposed as a tool for preliminary search and analysis in collections of (primarily) biological images with random and unknown contents (although the general scope of these collections should be known for easier interpretation of the results). As *proof-of-concept* examples, we use three small (but diversified) datasets containing images of butterflies, leaves and viruses.

In Section 2 of the paper we overview pre-existing tools and techniques contributing to the proposed method. Its description is provided in Section 3. Results for the selected datasets are presented in Section 4, while Section 5 summarizes the paper.

2 Typical Keypoint-Based Tools and Techniques

Affine-invariant keypoint detectors are the low-level tools of the method. Two popular detectors, i.e. Harris-Affine and Hessian-Affine, [8], are used because of their mutually supplementing characteristics. Harris-Affine highlights corner-like saliences, while Hessian-Affine returns saliences corresponding to blobs.

Keypoints are represented by SIFT descriptor, [7], quantized into visual words. A relatively small vocabulary SV of 2000 words is used. Such a vocabulary is able to accept significant photometric distortions of image contents, but the discriminative power of individual words is very low (e.g. comments in [10], [13]). Nevertheless, as shown in Section 3, we actually use a 3D Cartesian product of vocabularies so that the practical resolution of descriptions is much higher.

In general, keypoints can be matched using either O2O (typically *mutual nearest neighbor*) or M2M (typically *the same visual word*) schemes. However, regardless the scheme (and regardless the vocabulary size) none of the schemes based on individual keypoint correspondences can reliably distinguish between actually similar image fragments and random locally similar contents. Fig. 1 shows examples of matching in a pair of images sharing the same object and in an unrelated pair. For all schemes, there is no qualitative difference between the results for both pairs.

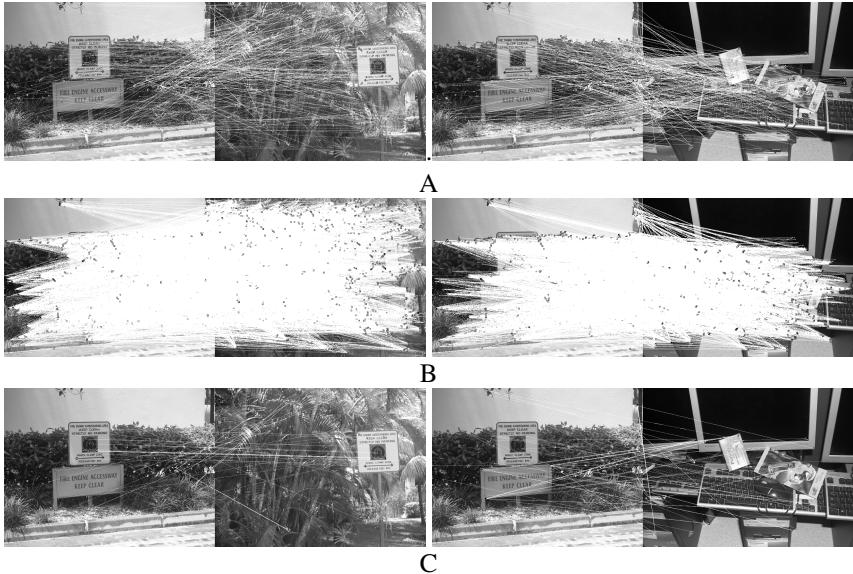


Fig. 1. Exemplary keypoint matching results using: (A) O2O scheme over Harris-Affine keypoints, (B) M2M (2000 words) over Hessian-Affine keypoints, (C) M2M (2^{20} words) over Harris-Affine keypoints

Partial near-duplicates are usually detected by the verification of configuration constraints for unspecified groups of preliminarily matched keypoints (e.g. [1, [3], [9], [14]]). However, this is a computation-intensive operation which is not fully scalable to large databases. In particular, if numerous matches are found using a small vocabulary (e.g. Fig. 1B) the configuration verification can be prohibitively costly. Moreover, it is generally believed (e.g. [13]) that small vocabularies excessively reduce *precision* of retrieved results.

Unfortunately, in biological images we need both small vocabularies and relaxed configuration constraints (because of highly diversified appearances of nominally the same objects – e.g. individual butterflies of the same species). Thus, the available methods and techniques are generally unable to handle such flexibility, unless they are trained using sufficiently diversified positive and (sometimes) negative examples to classify/recognize images or objects of predefined categories, e.g. [2], [5], [15].

3 Principles of the Method

Our objective is to develop a method which can overcome the difficulties highlighted in Section 2 for a wide range of biological images of diversified contents. In particular, our intensions are:

1. To bypass the configuration analysis for preliminarily matched keypoints by incorporating affine-invariant descriptions of keypoint neighborhood geometry into descriptors of the keypoints themselves. Descriptors of geometry are quantized into another small-size vocabulary so that significant geometric distortions can be tolerated.
2. To improve *precision* of image retrieval (when small vocabularies are used) by combining independently obtained results for Harris-Affine and Hessian-Affine keypoints. Only images retrieved in both operations are accepted (subject to additional details discussed below).

3.1 Extended Description of Keypoints

To represent geometry of keypoint neighborhoods, we propose to use a modification of the method discussed in [12]. First, we build limited-size neighborhoods of extracted keypoints. The neighborhoods consist of a limited number (not more than 20) keypoints of similar sizes (between 50% and 150% of the area of the central keypoint) and within a limited distance (between 70% and 200% of the Mahalanobis distance defined by the ellipse of the central keypoint) from the keypoint of interest; see Fig. 2A for illustration.

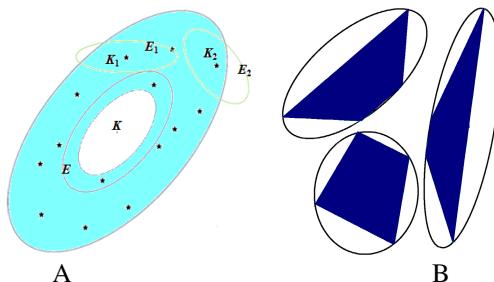


Fig. 2. A keypoint and its neighbors (A); the size of ellipses for two neighbors is shown as an illustration. Trapezoids built within a triplet of elliptic keypoints (B).

Subsequently, triplets of the keypoints are formed and their ellipses are represented by trapezoids shown in Fig. 2B. Then, the least complex affine-invariant moment expression

$$Inv = \frac{M_{20}M_{02} - M_{11}^2}{M_{00}^4}. \quad (1)$$

is used to characterize each of the trapezoids.

The range of *Inv* invariant is actually quantized into 12 values (words) so that geometry of the whole triplet of ellipses is described by a small vocabulary *GV* of $12^3 = 1728$ words. Altogether, a triplet of keypoints can be represented by SIFT words from a vocabulary of 2000 words (visual properties of individual keypoints) and one word from 1728 words of *GV* vocabulary (geometry of the whole triplet).

Given a keypoint K and its neighbors $\{K_1, K_2, \dots, K_N\}$, a number of $\{K, K_i, K_j\}$ triplets can be built around K as explained above. Our experiments show that the most typical numbers of such triplets are between 50 to 80 only (because we exclude triplets for which the triangles are too narrow). There are, nevertheless, some keypoints around which no triplets are found (e.g. large keypoints surrounded by very small keypoints only).

Altogether any keypoint K and its neighborhood are represented by the SIFT word $SV(K)$ and a set of 3-word phrases $SoP(K)$:

$$SoP(K) = \bigcup \{ SV(K_i), SV(K_j), GV(K, K_i, K_j) \} \quad (2)$$

from the Cartesian product $SV \times SV \times GV$.

3.2 Matching Keypoints and Images

Using descriptions proposed in Section 2.1, matching keypoints (i.e. actually matching their neighborhoods as well) is straightforward. Two keypoints K and L match if

$$SV(K) = SV(L) \quad \text{and} \quad SoP(K) \cap SoP(L) \neq \emptyset \quad (3)$$

i.e. the keypoints are visually similar and their neighborhoods are similar (at least partially) both visually and geometrically.

Compared to the correspondences obtained by matching only the keypoints themselves, i.e. $SV(K) = SV(L)$ (see the second row of Fig. 1) the results are more meaningful. The results in Fig. 3 show relatively few correspondences (and many of them are correct in the global context). Note that no verification of configuration constraints is needed; the results are obtained by simple keypoint matching.

Nevertheless, some correspondences are between locations which are semantically rather different. This is primarily because small *SV* and *GV* vocabularies can tolerate significant photometric and geometric distortions.

Thus, as the final verification step, we combine Harris-Affine and Hessian-Affine correspondences. A pair of matched Harris-Affine keypoints (K_{HA}, L_{HA}) is accepted only if there is another pair of Hessian-Affine keypoint (K_{HE}, L_{HE}) overlapping it (and another way around). Formally, “overlapping” means that the Mahalanobis distances (defined by the shapes of keypoint ellipses) between the corresponding keypoints are below the threshold. Fig. 4 provides an illustrative example (using 140% of the corresponding Mahalanobis unit distances as the threshold; this value is used in the paper).

Eventually, a pair of images is retrieved (i.e. is it assumed that both images contain partial near-duplicate of noticeable size) if at least one pair of keypoint correspondences is retained after the final verification. The coordinates of those keypoints indicate the approximate locations of the partial near-duplicates in the matched images.

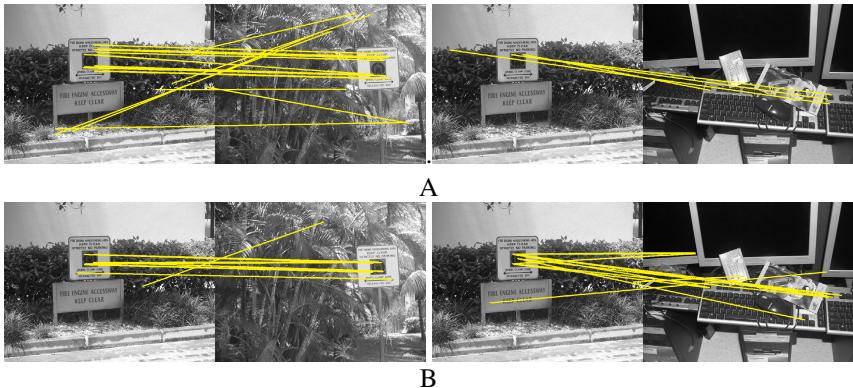


Fig. 3. Keypoint correspondences obtained by the proposed method for Harris-Affine keypoints (A) and Hessian-Affine keypoints (B). The same images are used in Fig. 1.

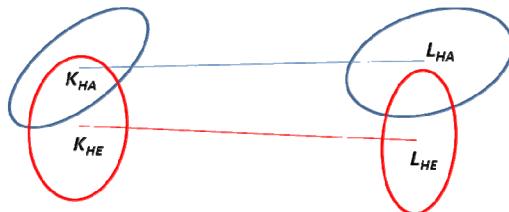


Fig. 4. A pair of matched Harris-Affine keypoints (blue) and Hessian-Affine keypoints (red) in close proximity



Fig. 5. Keypoint correspondences after the final verification. The same images are used in Figs 1 and 3.

It should be noted that each pair of keypoint correspondences additionally indicates an unspecified number of matches between keypoints from both neighborhoods.

The results after the final verification for exemplary images from Figs 1 and 3 are given in Fig. 5. The same roadsign is correctly detected as a partial near-duplicate in Fig. 5A. However, one of the keyboard keys is also sufficiently similar (because of small vocabularies) to the roadsign and recognized as a partial near-duplicate as well (Fig. 5B). This can be considered a disadvantage, but in biological or biomedical images such relatively weak similarities should be usually retrieved as potential candidates for semantically meaningful partial near-duplicates.

4 Experiments

The proposed approach is currently being tested on diversified databases of biological and biomedical images. The objective is to evaluate the practical significance of the results (relevance to the human interpretation of the images, retrieval of unspecified “hidden” semantics, etc.).

Three small-scale examples are discussed in this paper. In all examples only grey-level images are used (so that the visual analysis is more difficult).

4.1 Images of Butterflies

A part of the Ponce Research Group butterfly dataset¹ has been used. The same dataset was analyzed in [6] with similar concepts of keypoint matching. However, both training and geometric consistency verification were used. Other reported works on automatic recognition of butterflies (e.g. [4], [11]) also heavily rely on the specific properties of butterfly images.

The presented method achieves comparable results without any preliminary knowledge about the image domain and by using only individual keypoint matches (note that complexity of the final verification illustrated in Fig. 4 is negligible).

The total number of matched image pairs is 7,140 of which 1,140 are the semantic *ground truth* (i.e. showing butterflies of the same species). Our method retrieved 1,578 image pairs from which:

- ✓ 743 pairs contain butterflies of the same species (examples in Fig. 6A);
 - ✓ 696 pairs show butterflies with similar fragments of wings (see Fig. 6B);
 - ✓ 47 pairs indicate similarities between background plants (examples in Fig. 6C);
 - ✓ 87 pairs indicate similarities between butterflies and the background plants (good camouflage?);
- and
- ✓ 5 pairs show random similar fragment.

Altogether, *recall* of the method (in retrieval of image pairs containing butterflies of the same species) is 65.2%, while *precision* is nominally 47.1%. However the *actual precision* is 94.2% because both similarly looking wing fragments and similarly looking background plants are also semantically correct partial near-duplicates (especially because the domain of images is not specified and we should not distinguish between partial near-duplicates within different types of objects).

Additionally, we have built similarity graphs between the dataset images. It was found that its K -connected sub-graphs consist of nodes representing images of the same species butterflies. We do not discuss details of this issue in the paper. However, it is again mentioned in the following Section 4.2.

¹ http://www-cvr.ai.uiuc.edu/ponce_grp/data/

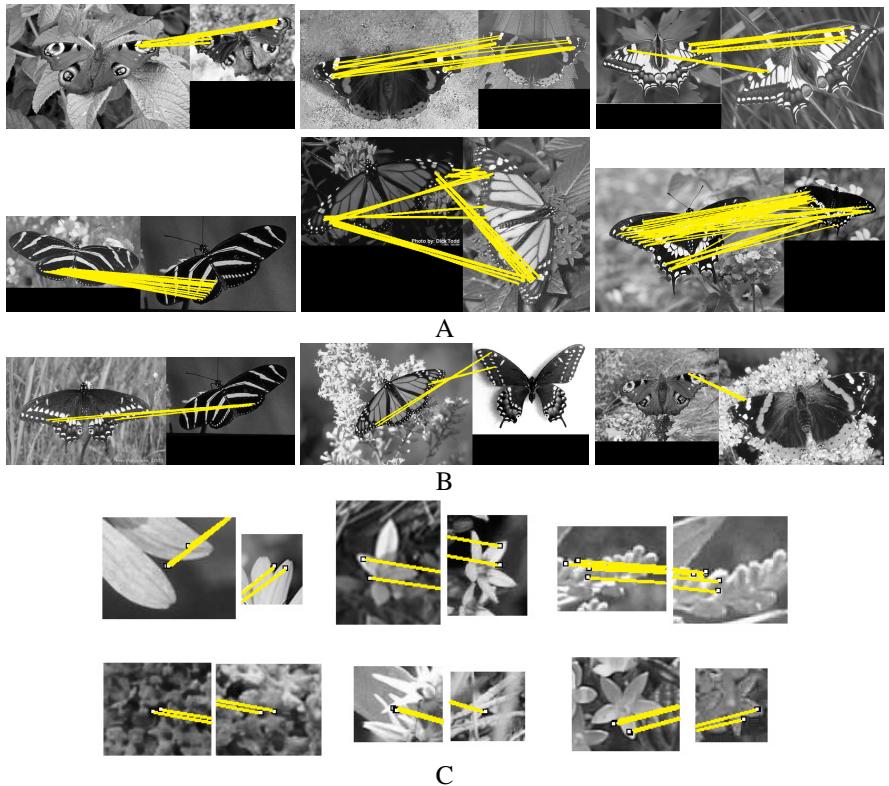


Fig. 6. Exemplary matches between the same species butterflies (A), between similar fragments of butterfly wings (B) and between plants in the image backgrounds (C).

4.2 Other Images

The other two datasets contain images of plant leaves² and viruses (low-resolution images collected from the web).

The leaf images are of rather low quality (see examples in Fig. 7). The total number of image pairs to be matched (images showing only contours of leaves were deleted) is 2,775. However, the ground truth is not available (even though preliminary grouping is provided).

The algorithm has retrieved 83 pairs of images containing partial near-duplicates. Surprisingly, the similarity graph of the matched images is strongly connected (the similarity graph is 5-connected) so that the retrieved pairs of images – with only 14 images contributing to these pairs - can prospectively define some properties of the retrieved images. Seven of those images are given in Fig. 8, and it can be clearly seen that all of them have jagged (at least partially) borders. Thus, we can conclude that this is the (semantic) property of leaves automatically identified by the algorithm.

² [http://www.imageprocessingplace.com/root_files_V3/
image_databases.htm](http://www.imageprocessingplace.com/root_files_V3/image_databases.htm)

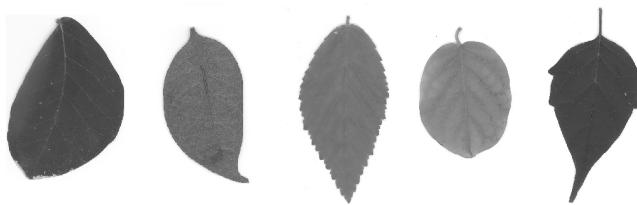


Fig. 7. Exemplary images from the dataset of leaves

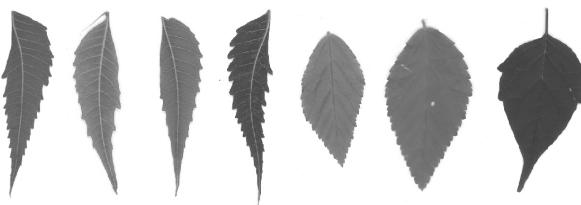


Fig. 8. Seven examples of leaves with (partially) jagged borders

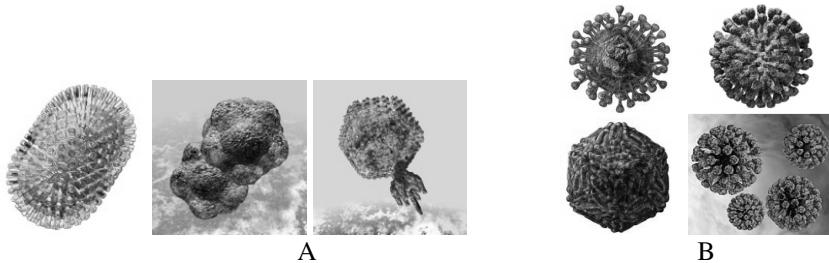


Fig. 9. Exemplary images of viruses (A) and two pairs of images retrieved by the system (B)

In the third test, 528 pairs of images (some are given in Fig. 9A) showing viruses of various diseases have been matched. Eventually, partial near-duplicates have been found in only two pairs of images (Lassa fever virus sharing partial near-duplicates with rubella virus, and SARS virus with West Nile fever virus) – see Fig. 9B.

In general the numbers of images in these two datasets are too small to reliably evaluate performances (e.g. *recall* and *precision*) especially because the ground truth classification is missing for some images.

Nevertheless, the visual inspection of the results confirms that the retrieved visual partial near-duplicates actually have some semantic significance.

5 Summary

We show that very simple and general tools, i.e. (1) keypoint matching using small vocabularies representing visual and geometric properties of keypoints and their neighborhoods and (2) superposition of results obtained for Harris-Affine and Hessain-Affine keypoints, can be often used to retrieve meaningful similarities

(approximated by partial near-duplicates) from visual databases semantically. No training or domain-specific approaches are required.

Because of small vocabularies used, the method can tolerate significant photometric and geometric distortions so that it is particularly suitable for processing biological/biomedical images.

The presented experiments preliminary confirm the practicality of the method. Currently, experiments are conducted on diversified and much larger databases of biomedical images.

References

1. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: Proc. IEEE Conf. CVPR 2009, pp. 17–24 (2009)
2. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proc. IEEE Conf. CVPR 2003, vol. 2, pp. 264–271 (2003)
3. Jegou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. Int. J. Comp. Vision 87, 316–336 (2010)
4. Kang, S.-H., Jeon, W., Lee, S.-H.: Butterfly species identification by branch length similarity entropy. J. Asia-Pacific Entomology 15, 437–441 (2012)
5. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Proc. IEEE Conf. CVPR 2009, pp. 951–958 (2009)
6. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: Proc. BMVC 2004, pp. 779–788 (2004)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Com. Vision 60, 91–110 (2004)
8. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comp. Vision 60, 63–86 (2004)
9. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. IEEE Conf. CVPR 2007, pp. 1–8 (2007)
10. Romberg, S., August, M., Ries, C.X., Lienhart, R.: Robust feature bundling. In: Lin, W., Xu, D., Ho, A., Wu, J., He, Y., Cai, J., Kankanhalli, M., Sun, M.-T. (eds.) PCM 2012. LNCS, vol. 7674, pp. 45–56. Springer, Heidelberg (2012)
11. Silveira, M., Monteiro, A.: Automatic recognition and measurement of butterfly eyespot patterns. Biosystems 95, 130–136 (2009)
12. Śluzek, A.: Large vocabularies for keypoint-based representation and matching of image patches. In: Fusello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012 Ws/Demos, Part I. LNCS, vol. 7583, pp. 229–238. Springer, Heidelberg (2012)
13. Stewénius, H., Gunderson, S.H., Pilet, J.: Size matters: Exhaustive geometric verification for image retrieval accepted for ECCV 2012. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 674–687. Springer, Heidelberg (2012)
14. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. In: Proc. IEEE Conf. CVPR 2009, pp. 25–32 (2009)
15. Yang, K., Wang, M., Hua, X.-S., Yan, S., Zhang, H.-J.: Assemble new object detector with few examples. IEEE Im. Proc. 20, 3341–3349 (2011)

Variational Model for Image Segmentation

Qiong Lou¹, Jialin Peng^{3,1}, Fa Wu², and Dexing Kong²

¹ Center of Mathematical Sciences, Zhejiang University, Hangzhou, China

² Department of Mathematics, Zhejiang University, Hangzhou, China

³ The School of Computer Science and Technology,
Huaqiao University, Xiamen, China

Abstract. Image segmentation is a fundamental problem in the field of image processing and computer vision with numerous applications. In the recent years, mathematical models based on partial differential equations and variational methods have led to superior results in computer vision. In this paper, we present a variational model which uses total variation as a regularization for image segmentation, and develop a primal-dual hybrid gradient algorithm to our model. We discuss images in the cases of Gaussian, Poisson and multiplicative speckle noise, the performance of our model is illustrated by experimental results on synthetic and real data. The proposed model can also preserve small anatomical structures better than Alex Sawatzky's region-based segmentation model with less CPU times.

1 Introduction

Image segmentation is a fundamental problem in image processing. The main goal of segmentation is to recover an object of interest from a given dataset by partitioning it into disjoint compartments. In general, edge-based ([1–3]), region-based ([4–6]) and edge-region-based ([7–9]) methods are common and useful in image segmentation.

Various tools have been proposed to solve image segmentation problem. Variational models have been extremely successful in a wide variety of image restoration problems, variation remains one of the most active areas of research in mathematical image processing and computer vision. The two-phase piecewise constant Mumford-Shah (MS) image segmentation model [4, 10] is one of the most important image segmentation model, and has been studied extensively in the last two decades. The convex segmentation model based on MS [11] can be regarded as an image restoration model. [11] unifies the image processing works of image segmentation and image restoration.

In [12], Alex Sawatzky proposed a region-based variational segmentation framework to segment images incorporating physical noise. This paper discuss the cases of Gaussian, Poisson and multiplicative speckle noise intensively. When come to the experiment of synthetic data with anatomical structures of different size, they choose parameter of the length of edge according to different criterions. But they do not get satisfying segmentation results. Actually, this model also

cannot get ideal segmentation when come to the case of inhomogeneous image with high Gaussian noise.

In the study of overcoming the unsatisfying results in [12], we study the recent advances of variational model [13] and region-based model [11]. We joint image segmentation and restoration together in this paper, and present a new variational model. In this model, we use the total variation (TV) term as a regularization. As for the fidelity term, we introduce approximate image. As for the numerical realization of our model, we develop an efficient and fast primal-dual hybrid gradient (PDHG) method [14,15]. This descent type algorithm alternates between primal and dual formulations and exploits the information from both primal and dual variables. The performance of the proposed model is illustrated by experimental results on synthetic and real data.

The paper is organized as follows. In section 2, we propose a novel variational model for segmentation. The PDHG algorithm corresponding to our model is introduced in section 3. In section 4, we give the experimental results and the discussions. Finally, a brief conclusion is presented in section 5.

2 The Proposed Model

In this section, we propose a new variational model for image segmentation.

2.1 Basic Operators

Let $\Omega \subset R^2$ be a bounded open connected set, Γ be a compact curve in Ω , and $f : \Omega \rightarrow R$ be a given image. Without loss of generality, we restrict the range of f in $[0, 1]$ and hence $f \in L^\infty(\Omega)$.

Let $u : \Omega \rightarrow R$ be a continuous or even differentiable in $\Omega \setminus \Gamma$ but may be discontinuous across Γ . u is the restoration of f . $\|\cdot\|$ denotes the 2-norm on region set.

For simplicity, we restrict our model to a two-phase segmentation problem, and then we introduce the following notations. First, we assume that we want to segment the image domain Ω into a background and a target subregion, which we denote with Ω_b and Ω_t , respectively. Subsequently, we introduce an indicator function χ in order to represent both subregions in the form that

$$\chi(x) = \begin{cases} 1, & \text{if } x \in \Omega_b, \\ 0, & \text{else.} \end{cases} \quad (1)$$

Finally, we use the well-known relation between the Hausdorff measure and the total variation of an indicator function, which implies that

$$\mathcal{H}^{d-1}(\Gamma) = |\chi|_{BV(\Omega)}. \quad (2)$$

Here $|\cdot|_{BV(\Omega)}$ denotes the total variation of a function in Ω .

2.2 The Proposed Model

The two-phase variational model can be written as

$$\begin{aligned} \min_{\{u, u_b, u_t\}} \{E(u, u_b, u_t)\} = & \frac{1}{2} \int_{\Omega} (f - Ku)^2 dx + \sum_i \frac{\alpha_i}{2} \int_{\Omega_i} (u - u_i)^2 dx \\ & + \sum_i \frac{\gamma_i}{2} \int_{\Omega} |\nabla u_i|^2 dx + \xi |u|_{TV(\Omega)} + \beta \mathcal{H}^{d-1}(\Gamma), \end{aligned} \quad (3)$$

where $i \in \{b, t\}$, u_i is the approximation of u in Ω_i . K can be the intensity operator or a blurring operator. In our experiments we always set K to be identity operator. Thus together with Eq. (1) and Eq. (2), the proposed model can be shown as

$$\begin{aligned} & \min_{\{u, u_b, u_t, \chi\}} \{E(u, u_b, u_t, \chi)\} \\ = & \frac{1}{2} \int_{\Omega} (f - u)^2 dx + \sum_i \frac{\alpha_i}{2} \int_{\Omega_i} (u - u_i)^2 dx \\ & + \sum_i \frac{\gamma_i}{2} \int_{\Omega} |\nabla u_i|^2 dx + \xi |u|_{TV(\Omega)} + \beta |\chi|_{TV(\Omega)}. \end{aligned} \quad (4)$$

We can split the minimization problem into separate subproblems as following minimization problems,

$$u^{k+1} \in \arg \min_u \left\{ \frac{1}{2} \int_{\Omega} (f - u^k)^2 dx + \sum_i \frac{\alpha_i}{2} \int_{\Omega_i} (u^k - u_i^k)^2 dx + \xi |u^k|_{TV(\Omega)} \right\}. \quad (5)$$

$$u_b^{k+1} \in \arg \min_{u_b} \left\{ \frac{1}{2} \int_{\Omega} \alpha_b \chi^k (u^{k+1} - u_b^k)^2 dx + \frac{\gamma_b}{2} \int_{\Omega} (\nabla u_b^k)^2 dx \right\}. \quad (6)$$

$$u_t^{k+1} \in \arg \min_{u_t} \left\{ \frac{1}{2} \int_{\Omega} \alpha_t (1 - \chi^k) (u^{k+1} - u_t^k)^2 dx + \frac{\gamma_t}{2} \int_{\Omega} (\nabla u_t^k)^2 dx \right\}. \quad (7)$$

$$\begin{aligned} \chi^{k+1} \in & \arg \min_{\chi} \left\{ \frac{1}{2} \int_{\Omega} \alpha_b \chi^k (u^{k+1} - u_b^{k+1})^2 \right. \\ & \left. + \alpha_t (1 - \chi^k) (u^{k+1} - u_t^{k+1})^2 dx + \beta |\chi|_{TV(\Omega)} \right\}. \end{aligned} \quad (8)$$

3 Numerical Realization

For the sake of simplicity, our model is restricted in two-phase condition.

Because of the total variation of u can be defined as

$$\begin{aligned} TV_{\Omega}(u) &= \int_{\Omega} |\nabla u| dx \\ &= \max_{p \in C_0^1, \|p\| \leq 1} \int_{\Omega} \nabla u \cdot p = \max_{\|p\| \leq 1} \int_{\Omega} -u \operatorname{div} p dx. \end{aligned} \quad (9)$$

The primal-dual formulation of the proposed model is given by

$$\begin{aligned}
 & \min_{\{u, u_b, u_t, \chi\}} \max_{\|q\| \leq 1} \max_{\|p\| \leq 1} \{E(u, u_b, u_t, \chi, p, q) \\
 & := \frac{1}{2} \int_{\Omega} (f - u)^2 dx + \sum_i \frac{\alpha_i}{2} \int_{\Omega_i} (u - u_i)^2 dx \\
 & \quad + \sum_i \frac{\gamma_i}{2} \int_{\Omega} (\nabla u_i)^2 dx + \xi \int_{\Omega} p \nabla u dx + \beta \int_{\Omega} q \nabla \chi dx \\
 & = \frac{1}{2} \int_{\Omega} (f - u)^2 + \alpha_b \chi (u - u_b)^2 + \alpha_t (1 - \chi) (u - u_t)^2 dx \\
 & \quad + \sum_i \frac{\gamma_i}{2} \int_{\Omega} (\nabla u_i)^2 dx + \xi \int_{\Omega} p \nabla u dx + \beta \int_{\Omega} q \nabla \chi dx \}
 \end{aligned} \tag{10}$$

The specific PDHG algorithm corresponding the proposed model is as follows.

1. Dual Step

(1) Fix $q = q^k$, $u = u^k$, $u_b = u_b^k$, $u_t = u_t^k$, $\chi = \chi^k$, apply one step of projected gradient acescent method to the maximization problem

$$\max_{\|p\| \leq 1} E(u^k, u_b^k, u_t^k, \chi^k, p, q^k). \tag{11}$$

The ascent direction is $\nabla_p E(u^k, u_b^k, u_t^k, \chi^k, p, q^k) = \nabla u^k$, so we update p as

$$p^{k+1} = P_X(p^k + \xi \tau_k \nabla u^k), \tag{12}$$

where τ_k is the dual stepsize and X is the following space

$$X = p : p \in R^{N \times N}, \quad \|p\| \leq 1. \tag{13}$$

P_X denotes the projection onto the set X , i.e.,

$$P_X(p) = \frac{p}{\max\{\|p\|, 1\}}. \tag{14}$$

(2) Fix $p = p^{k+1}$, $u = u^k$, $u_b = u_b^k$, $u_t = u_t^k$, $\chi = \chi^k$, apply one step of projected gradient acescent method to the maximization problem

$$\max_{\|q\| \leq 1} E(u^k, u_b^k, u_t^k, \chi^k, p^{k+1}, q). \tag{15}$$

The ascent direction is $\nabla_q E(u^k, u_b^k, u_t^k, \chi^k, p^{k+1}, q) = \nabla \chi^k$, so we update q as

$$q^{k+1} = P_Y(q^k + \beta \gamma_k \nabla \chi^k), \tag{16}$$

where γ_k is the dual stepsize and Y is the following space

$$Y = q : q \in R^{N \times N}, \quad \|q\| \leq 1. \tag{17}$$

P_Y denotes the projection onto the set Y , i.e.,

$$P_Y(q) = \frac{q}{\max\{\|q\|, 1\}}. \tag{18}$$

2. Primal Step

(1) Fix $p = p^{k+1}$, $q = q^{k+1}$, $u_b = u_b^k$, $u_t = u_t^k$, $\chi = \chi^k$, apply one step of projected gradient acescent method to the maximization problem

$$\min_u E(u, u_b^k, u_t^k, \chi^k, p^{k+1}, q^{k+1}). \quad (19)$$

The Euler-Lagrange equation with respect to u without constraint is

$$\begin{aligned} & \nabla_u E(u^k, u_b^k, u_t^k, \chi^k, p^{k+1}, q^{k+1}) \\ &= (u^k - f) + \alpha_b \chi^k (u^k - u_b^k) + \alpha_t (1 - \chi^k) (u^k - u_t^k) - \beta \operatorname{div} p^{k+1} \\ &= 0. \end{aligned} \quad (20)$$

So we update u as

$$u^{k+1} = P_U(u^k + \theta_u (-\nabla_u E)), \quad (21)$$

where θ_u is the primal stepsize and U is the following space

$$U = u : u \in R^{N \times N}, \quad 0 \leq u(x) \leq 1. \quad (22)$$

P_U denotes the projection onto the set U , i.e.,

$$P_U(u) = \max\{0, \min\{u, 1\}\}. \quad (23)$$

(2) Fix $p = p^{k+1}$, $q = q^{k+1}$, $u = u^{k+1}$, $u_t = u_t^k$, $\chi = \chi^k$, apply one step of projected gradient acescent method to the maximization problem

$$\min_{u_b} E(u^{k+1}, u_b, u_t^k, \chi^k, p^{k+1}, q^{k+1}). \quad (24)$$

The Euler-Lagrange equation with respect to u_b without constraint is

$$\begin{aligned} & \nabla_{u_b} E(u^{k+1}, u_b^k, u_t^k, \chi^k, p^{k+1}, q^{k+1}) \\ &= \chi^k \alpha_b (u_b^k - u^{k+1}) - \gamma_b \operatorname{div} \left(\frac{\nabla u_b^k}{|\nabla u_b^k|} \right) = 0. \end{aligned} \quad (25)$$

So we update u_b as

$$u_b^{k+1} = P_U(u_b^k + \theta_{u_b} (-\nabla_{u_b} E)), \quad (26)$$

where θ_{u_b} is the primal stepsize.

(3) Fix $p = p^{k+1}$, $q = q^{k+1}$, $u = u^{k+1}$, $u_b = u_b^{k+1}$, $\chi = \chi^k$, apply one step of projected gradient acescent method to the maximization problem

$$\min_{u_t} E(u^{k+1}, u_b^{k+1}, u_t, \chi^k, p^{k+1}, q^{k+1}). \quad (27)$$

The Euler-Lagrange equation with respect to u_t without constraint is

$$\begin{aligned} & \nabla_{u_t} E(u^{k+1}, u_b^{k+1}, u_t^k, \chi^k, p^{k+1}, q^{k+1}) \\ &= \alpha_t (1 - \chi^k) (u_t^k - u^{k+1}) - \gamma_t \operatorname{div} \left(\frac{\nabla u_t^k}{|\nabla u_t^k|} \right) = 0. \end{aligned} \quad (28)$$

So we update u_t as

$$u_t^{k+1} = \text{P}_U(u_t^k + \theta_{u_t}(-\nabla_{u_t} E)), \quad (29)$$

where θ_{u_t} is the primal stepsize.

(4) Fix $p = p^{k+1}$, $q = q^{k+1}$, $u = u^{k+1}$, $u_b = u_b^{k+1}$, $u_t = u_t^{k+1}$, apply one step of projected gradient ascent method to the maximization problem

$$\min_{\chi} E(u^{k+1}, u_b^{k+1}, u_t^{k+1}, \chi, p^{k+1}, q^{k+1}). \quad (30)$$

The Euler-Lagrange equation with respect to χ without constraint is

$$\begin{aligned} 0 &= \nabla_{\chi} E(u^{k+1}, u_b^{k+1}, u_t^{k+1}, \chi^k, p^{k+1}, q^{k+1}) \\ &= \frac{1}{2}(\alpha_b(u^{k+1} - u_b^{k+1})^2 - \alpha_t(u^{k+1} - u_t^{k+1})^2) - \beta \text{div} q^{k+1}. \end{aligned} \quad (31)$$

So we update χ as

$$\chi^{k+1} = \text{P}_{\Phi}(\chi^k + \theta_{\chi}(-\nabla_{\chi} E)), \quad (32)$$

where θ_{χ} is the primal stepsize and Φ is the following space

$$\Phi = \{\chi \in \text{BV}(\Omega; \{0, 1\})\}. \quad (33)$$

To sum up, we have the following Algorithm 1 for the proposed model. Where

Algorithm 1. PDHG.

Initialization:

$u^0 = f$, $\chi = \chi^0$, $u_b^0 = F_0$, $u_t^0 = F_0$, $q^0 = 0$, $p^0 = 0$. Fixed α_b , α_t , γ_b , γ_t , ξ , β .

STEP 1:

Choose the stepsize τ_k , γ_k , θ_u , θ_{u_b} , θ_{u_t} , θ_{χ} , and set $k \leftarrow 0$.

STEP 2:

Update p^{k+1} by Eq. (12),

Update q^{k+1} by Eq. (16),

Update u^{k+1} by Eq. (21),

Update u_b^{k+1} by Eq. (26),

Update u_t^{k+1} by Eq. (29),

Update χ^{k+1} by Eq. (32).

STEP 3:

Terminate if a stopping criterion is satisfied; otherwise set $k \leftarrow k + 1$ and return to step 1.

$F_0 = \chi \text{mean}(f\chi^0) + (1 - \chi^0) \text{mean}(f(1 - \chi^0))$, here $\text{mean}(\cdot)$ means the mean value. In the algorithm, we choose

$$\frac{\|u^{k+1} - u^k\|^2}{\|u^k\|^2} < 1e - 5 \quad (34)$$

as the stopping criterion.

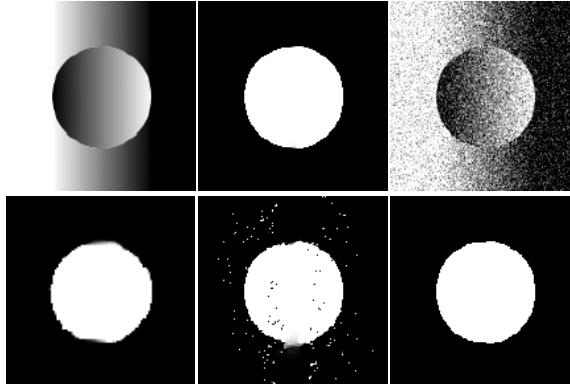


Fig. 1. Comparison of segmentation for synthetic data biased by Gaussian noise. The clean data, its segmentation and noisy image are shown in the first row. The segmentation results of MS, AM and our model, are shown in the second row respectively.

4 Experimental Results and Discussion

In this section, we test our variational model, and compared with the the models of MS [4] and AM [12]. We report on experiments for three test problems in image segmentation. The noisy images are generated by adding Gaussian, multiplicative speckle, and Poisson noise to the clean images using the MATLAB function *imnoise*. In order to implement the study, we set $\alpha_i = 1.0$, $\gamma_i = 0.2$, $\xi = 1.5$, $\beta = 0.2$ throughout the experiments. Images add Gaussian noise with variance set to 0.04 , add speckle noise with parameter set to 0.1.

Firstly, we investigate the experimental results on synthetic image with inhomogeneities. For this purpose we use an image of size 143×143 pixels scaled to $[0,1]$ with a simple object structure illustrated in Fig. 1. This image has the same mean value in Ω_t and Ω_b , but with strong intensity changes at the border of the object structure. We can see that our model can get satisfying segmentation when AM cannot.

Secondly, we need segmentation of the main structure without noise, and preservation of small anatomical structures without loss of details. For this purpose we use a synthetic data of size 150×150 with anatomical structures of different size as we show in Fig. 2. In this image, there are three small squares with size of 1, 2 and 4 pixels to simulate minor structures. From the experimental results in Fig. 2, we can find that our model can get better result than AM and MS. In Fig. 2, it is easily to find that AM and MS both cannot preserve the small squares when synthetic image biased by multiplicative speckle noise, but our model performs well.

Fig. 3 gives the segmentation results of real noisy image, which is added mammogram with poisson noise. The original mammogram shows cyst in the breast tissue, the aim of our segmentation is getting the edge of the cyst. It is easy to find that our model is less sensitive to noise than AM .

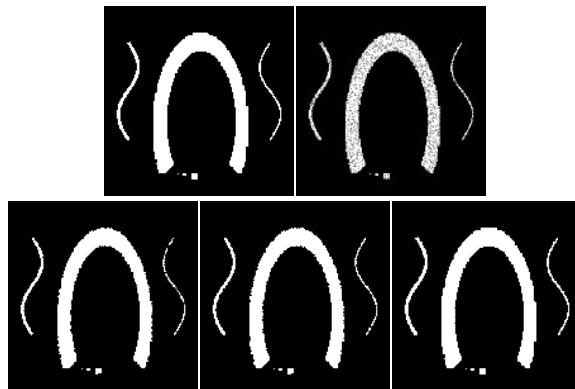


Fig. 2. Comparison of segmentation of synthetic data with anatomical structures of different size by adding multiplicative speckle noise. The clean data and noisy image are shown in the first row. The segmentation of MS, AM and our model are presented in the second row, respectively.

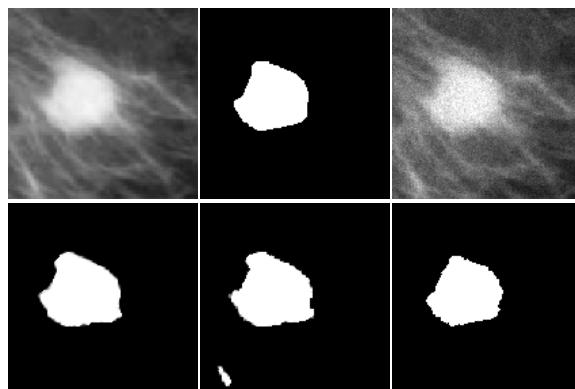


Fig. 3. Comparison of segmentation of cyst by adding Poisson noise. The original data, its segmentation and noisy image are shown in the first row. The segmentation of MS, AM and our model are presented in the second row, respectively.

From the experimental results, we can easily find that our model is more robust to the noise of different types than the AM model. In the present study, we use $|u|_{TV(\Omega)}$ and the smoothed image u in $\|u - u_i\|$, rather than $\|f - u_i\|$ in AM, that may help get better denoising results than AM. Actually, our model also reduces the iterative steps, and needs less time for getting satisfying results. Also, Our model can preserve the small anatomical structures without loss of details better than AM model. The number of iterations and CPU times (in seconds) for the experimental results of our model and AM in Fig. 1, Fig. 2 and Fig. 3 are shown in Table. 1.

Table 1. Iterative steps and times

Model	Fig.1		Fig.2		Fig.3	
	Iter	Time	Iter	Time	Iter	Time
proposed	9	7.76	4	4.55	13	10.86
AM	16	12.79	12	10.16	16	12.62

5 Conclusion

In this paper, we propose a variational model for image segmentation. For the numerical realization of our model, we develop its PDHG algorithm. In particular, we implement synthetic data and real image with physical noise, especially, Gaussian noise, Poisson noise, and multiplicative speckle noise. We use the presented model for automated image segmentation. In our model, we utilize the difference of smoothed image between background and target subregion. Therefore it is less sensitive to noise than AM. Actually, our model is robust to high noise data. It can get the main structure without noise artifacts, and preserve small anatomical structures with less loss of details under noisy cases. Another advantage of our method is that the algorithm we presented needs less CPU times than AM. The experimental results show promising results with efficiency.

Acknowledgements. This work was supported in part by the NNSF of China (Grant No.: 11271323), Zhejiang Provincial Natural Science Foundation of China (Grant No.: Z13A010002) and a National Science and Technology Project during the twelfth five-year plan of China (2012BAI10B04).

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision* 1, 321–331 (1988)
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. *International Journal of Computer Vision* 22, 61–79 (1997)
3. Li, C., Xu, C., Gui, C., Fox, M.D.: Level set evolution without re-initialization: a new variational formulation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 1, pp. 430–436. IEEE (2005)
4. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics* 42, 577–685 (1989)
5. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Transactions on Image Processing* 10, 266–277 (2001)
6. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision* 50, 271–293 (2002)
7. Wani, M.A., Batchelor, B.G.: Edge-region-based segmentation of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 314–319 (1994)

8. Mueller, M., Segl, K., Kaufmann, H.: Edge-and region-based segmentation technique for the extraction of large, man-made objects in high-resolution satellite imagery. *Pattern Recognition* 37, 1619–1628 (2004)
9. Zhang, Y., Matuszewski, B.J., Shark, L.K., Moore, C.J.: Medical image segmentation using new hybrid level-set method. In: Fifth International Conference on BioMedical Visualization, MEDIVIS 2008, pp. 71–76. IEEE (2008)
10. Krishnan, D., Pham, Q.V., Yip, A.M.: A primal-dual active-set algorithm for bilaterally constrained total variation deblurring and piecewise constant mumford-shah segmentation problems. *Advances in Computational Mathematics* 31, 237–266 (2009)
11. Cai, X., Chan, R., Zeng, T.: Image segmentation by convex approximation of the mumford–shah model (2012) (preprint)
12. Sawatzky, A., Tenbrinck, D., Jiang, X., Burger, M.: A variational framework for region-based segmentation incorporating physical noise models. *Journal of Mathematical Imaging and Vision*, 1–31 (2012)
13. Peng, J.L., Dong, F.F., Kong, D.X.: Recent advances of variational model in medical imaging and applications to computer aided surgery. *Applied Mathematics-A Journal of Chinese Universities* 27, 379–411 (2012)
14. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report, 08–34 (2008)
15. Zhu, M., Wright, S.J., Chan, T.F.: Duality-based algorithms for total-variation-regularized image restoration. *Computational Optimization and Applications* 47, 377–400 (2010)

Sky Segmentation by Fusing Clustering with Neural Networks

Ali Pour Yazdanpanah¹, Emma E. Regentova¹, Ajay Kumar Mandava¹,
Touqeer Ahmad², and George Bebis²

¹ Dept. of Electrical and Computer Engineering, University of Nevada, Las Vegas
Maryland Parkway Las Vegas, Nevada 89154-4026

² Dept. of Computer Science & Engineering, University of Nevada, Reno
1664 N. Virginia Street Reno, Nevada 89557-0208

Abstract. Sky segmentation is an important task for many applications related to obstacle detection and path planning for autonomous air and ground vehicles. In this paper, we present a method for the automated sky segmentation by fusing K-means clustering and Neural Network (NN) classifications. The performance of the method has been tested on images taken by two Hazcams (ie., Hazard Avoidance Cameras) on NASA's Mars rover. Our experimental results show high accuracy in determining the sky area. The effect of various parameters is demonstrated using Receiver Operating Characteristic (ROC) curves.

1 Introduction

NASA's Mars Exploration Rover mission (MER) and Mars Science Laboratory mission (MSL) are ongoing robotic space missions involving three rovers, exploring Mars. Two of the most important tasks during their missions are route planning, and path finding. The first step in route planning and path finding is to determine the suitability of the terrain for traversal. This includes extracting appropriate features for assessing rover navigation difficulty. To accomplish this task, accurate sky segmentation is required. This is not an easy task, however, due to the diversity of skyline shapes (boundaries between sky regions and non-sky areas) and clutter like clouds.

There are two main categories of sky segmentation found in computer vision literature. In the first category, the problem is addressed as finding a horizon line/sky line which mostly depends on edge detection and some post processing on top of detected edges. The regions above the horizon line are labeled as sky whereas the regions below the horizon are labeled as non-sky. In the second category of sky segmentation, the problem is formulated as a pixel wise classification problem so every pixel in the given image gets a sky or non-sky label. In [4] Lie et al. have presented an edge based horizon line detection method. They formulate the horizon finding problem as a multi-stage graph problem where a shortest path is found extending from the left most column to the right most column. A Sobel/Canny edge detector is applied on the given gray scale image. The detected edges are used as graph nodes and links with zero

or higher costs are placed between nodes if they are adjacent or have gaps. The gaps between edges (nodes) are filled using interpolation by introducing dummy nodes with higher costs. This method is robust in nature but it is time consuming and depends heavily on certain parameters (e.g. tolerance of gap (t_{og}) used for gap filling).

Kim et al. [5] try to extract the sky line in cluttered and cloudy environments. First, they model the clutter and then, using an iterative scheme, they find the sky line using multistage edge filtering. Their approach is based on a limited scenario and requires modeling the clutter before the extraction of sky line which is not always available. Although not directly focused on the problem of sky segmentation, their method of ray-based color image segmentation can be easily adapted for sky segmentation. The image segmentation process of Xu et al. [6] models the segments as clusters with centroids and applies ray shooting from the centroids towards the boundaries. In their results of outdoor scenes, a single centroid is mostly found for the sky; hence, it can be used to distinguish the sky region from the non-sky regions. Their approach is robust as compared to the popular K-means and normalized graph cuts algorithms used for image segmentation.

McGee et al. [1] have presented a sky segmentation technique based on color channels. In their approach, they try to find a linear separation (linear line) between sky and non-sky regions using Support Vector Machines (SVMs). Their approach is motivated by the objective of obstacle detection for small UAVs. The underlying assumption of this method is that a linear boundary divides the sky and non-sky region is not general enough and gets violated very often. The authors of [3] also assume that the horizon line is a line; they find the true horizon line among various candidates as the line that best segments the sky and non-sky regions. They have used various color and texture features (e.g., mean intensity values of three color channels, entropy, smoothness, uniformity and third moment etc.) to train several classifiers. Although their approach finds a good horizon line, the underlying assumption of the horizon line being linear is again not generally valid. The underlying assumption in the approach of Ettinger et al. [7] is also that the horizon line is linear. They model the sky and non-sky regions using Gaussian distributions and try to find the optimum boundary by dividing these two distributions.

In [2], Croon et al. have addressed this issue using shallow decision trees (J48 Implementation of C4.5 algorithm) based on color and texture features. The choice of decision trees is motivated by the computational efficiency achieved at run time since their goal is to use sky segmentation for static obstacle avoidance by Micro Air Vehicles (MAVs). They have extended the features used in [3] and introduced new features such as cornerness, grayness feature and Fisher discriminant features etc. In contrast to [3], they have used an extended database to train their classifier and a large number of features; hence their approach is more robust and capable of finding non-linear sky boundaries.

Todorovic et al. [8] have tried to circumvent the earlier assumptions [7] of the horizon line being linear and modeling the sky/non-sky regions using Gaussian distributions. In [8], they built prior statistical models for sky and non-sky regions based on

color and texture features. They argue the importance of both color (Hue and Intensity) and texture features (Complex Wavelet Transform) due to enormous variations in sky and ground appearances. A Hidden Markov Tree model was trained based on these features, yielding a robust horizon detection algorithm, capable of detecting non-linear horizons as well.

Although color could provide significant information to a detector/classifier, the use of grayscale images is advantageous because of faster processing time. The proposed method employs grayscale image characteristics. Our goal is to obtain a precise skyline; thus the assumption of the horizon line being linear as in [1], [3], and [7] is not valid. Most of the methods discussed above have been evaluated using a limited dataset. The method in [3] was tested on two sets of 10 images yielding an accuracy in the range of 90-99%. In [4], experiments were conducted using 25 grayscale images. The method in [5] was tested using 38 images, yielding 84.2% accuracy. Croon et al. [2] addressed non-linear sky boundaries. The performance of their method depends on the proportion of the sky area in the test images. Precision over 90% has been attained on images which contain more than 25% of sky.

The goal of our research is to develop a high performance method and evaluate it on a sufficiently large data set. The objective is to increase the True Positive rate while reducing False Positives. As machine learning techniques have shown good potential, our method uses a Neural Network (NN) classifier for pixel classification with successive refinements. A total of 16 features were used including raw intensity values and texture features. In a post-processing step, the output of the NN classifier is refined using geometric properties and some heuristics. The output of post-processing is further refined by fusing it with clustering result obtained using K-means. The results of fusion are further post-processed. We discuss the effect of various parameters of the method and provide Receiver Operating Characteristic (ROC) curves using a large number of test images.

The paper is organized as follows. Section 2 describes the proposed method for sky segmentation. Section 3 presents our experimental results. Conclusions and the future work are presented in Section 4.

2 Proposed Algorithm for Sky Segmentation

In our method, the classification of the image into sky and non-sky regions is performed by fusing the results of K-means clustering with those obtained using a NN classifier. Fig. 1 shows the block diagram of the proposed sky segmentation algorithm. First, we classify the input image into sky and non-sky regions using the NN classifier. The input to the NN classifier include pixel values and texture features extracted from 9×9 non-overlapping image blocks. The output is ‘1’ (for sky) or ‘0’ (for non-sky). The results of the NN classifier are post-processed and fused with the clustering results obtained using K-means clustering [9]. Finally, a second post-processing stage produces a valid mask wherein ‘1’ marks the sky region and the rest is marked as ‘0’.

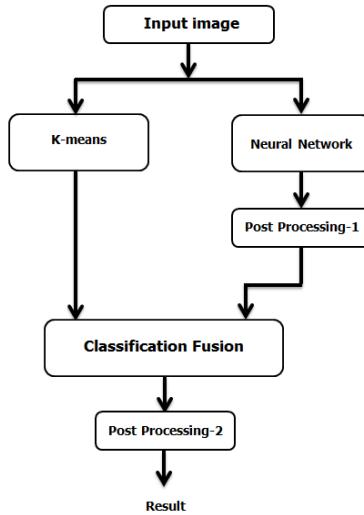


Fig. 1. Block diagram of our proposed sky segmentation method

2.1 Neural Network

Neural networks have been successfully applied to a variety of real world classification tasks in industry, business and science [10]. The NN used here is two-layer feed-forward back-propagation network with 16 inputs, 20 nodes in the hidden layer and one output node indicating “1” for the sky; “0” for non-sky region (Fig. 2).

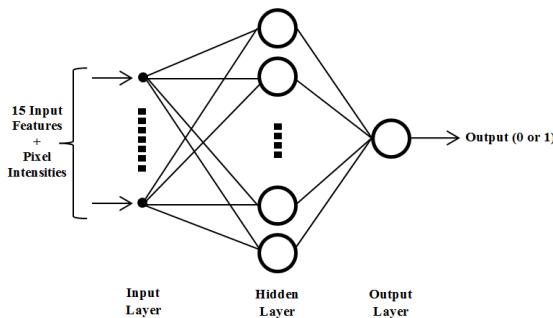


Fig. 2. Architecture of the Neural Network

The NN was trained using the gradient descent method. The features used were extracted from 9×9 non-overlapping blocks from 236 images selected randomly from our dataset. The NN was trained using pixel intensities and 15 texture features extracted from the patches (includes GLCM Features such as Dissimilarity, Energy, Entropy, Maximum probability, Sum entropy, Difference variance, Difference entropy, Inverse difference normalized, Inverse difference moment normalized,

Homogeneity, Cluster Prominence, Information measure of correlation1, Information measure of correlation2, Cluster Shade) [11],[12],[13].The total number of training blocks for the sky region is 127679 and the total number of training blocks for the non-sky region is 408749. The result of the NN is a binary map of sky/non-sky region. Fig. 3 shows the NN output for a sample image.

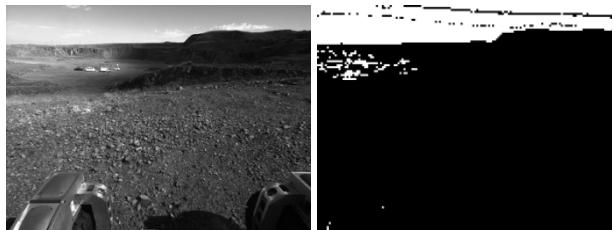


Fig. 3. Left: input image; Right: NN output

2.2 Post-processing

In the first post-processing stage, we remove “non-sky” patches classified as “sky” and “sky” patches classified as “non-sky”. To be considered as part of the sky, a region should satisfy the following assumptions:

- 1- The size of the connected region in the sky part of the image should be larger than a certain threshold; the default value is 450 pixels for our dataset.
- 2- The connected region of the sky should be adjacent to the upper edge of the image, or should be connected to the upper n pixels of the image (default value of n is 10 pixels).
- 3- If the sky region is adjacent to either the left, right or bottom boundaries, then sky segmentation has failed.
- 4- If the region of sky appears in an internal portion of the image, the segmentation is considered false and that region is removed.

2.3 K-Means Clustering

K-means clustering is a method of cluster analysis which aims to partition n observations into k clusters where each observation belongs to the cluster with the nearest mean [9]. The method is applied to partition the image into k clusters; it outputs a label matrix, where each pixel is assigned a label of the cluster it belongs to. The default value for the number of means is 10. This value was obtained experimentally as shown in Section 3. A number of disjoint clusters can be assigned the same label i ($i = 1 \dots Num$, where Num is a number of clusters), so generally there could be S regions with the label i , that is, the output of clustering is $R_j = \{R_1, R_2, \dots, R_{S_i}\}$ - disjoint clusters sharing label i (Fig.4).

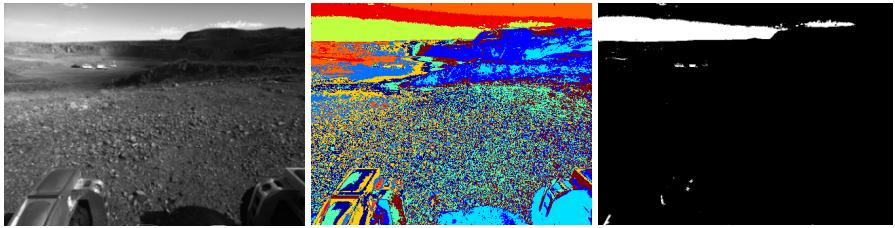


Fig. 4. Left: input image, middle: k-means result ($k=10$), Right: 106 disjoint clusters with label $i = 8$ shown from input image (left). ($R_j = \{R_1, R_2, \dots, R_{106}\}$)

2.4 Fusion Algorithm

At the fusion step, the algorithm checks if the size of the intersection between each region obtained using K -means and the corresponding region determined as sky in the post-processed NN results constitutes a certain part of the K -means clustered region. If this condition holds, the algorithm marks the whole region as sky. The algorithm uses a threshold to evaluate the relative size of the intersection. The default value is $Th = 0.5$. In Section 3, we will show how the performance of the method is affected by the choice of the threshold. Next, all sky regions are merged together to produce the final binary map - F matrix. Fig. 5 shows the pseudo-code for the fusion methodology, where

Num - number of clusters.

F - final segmentation matrix (initially, zero matrix).

Th - tunable threshold (between 0 and 1).

N - output matrix for post-processed neural network stage.

$N_j = \{N_1, N_2, \dots, N_{S_i}\}$ - clusters in matrix N paired with/one to one relationship with $\{R_1, R_2, \dots, R_{S_i}\}$ clusters.

$F_j = \{F_1, F_2, \dots, F_{S_i}\}$ -clusters in matrix F paired with/one to one relationship with $\{R_1, R_2, \dots, R_{S_i}\}$ clusters.

E_j - number of pixels in R_j .

In the second post-processing stage, we use the same rules as in the first post-processing stage, except Rule #1.

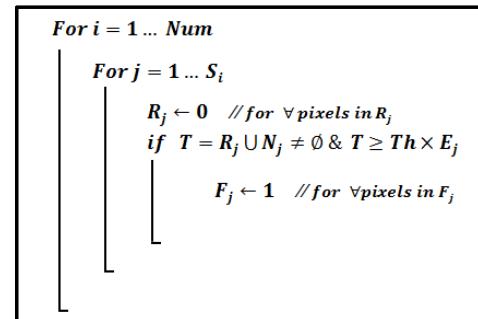


Fig. 5. Fusion algorithm

3 Experimental Results

3.1 Database

The database used in this work consists of 1482 (1038x1388) grayscale images taken by two Hazcams (hazard avoidance cameras) cameras on NASA's Mars rovers. Hazcams are photographic cameras sensitive to the visible light. They have a wide field of view (approximately 120° both horizontally and vertically) to allow a large amount of terrain to be visible. They are mounted on the front and rear of NASA's Mars rovers. These images are used by the rover to autonomously navigate around hazards.

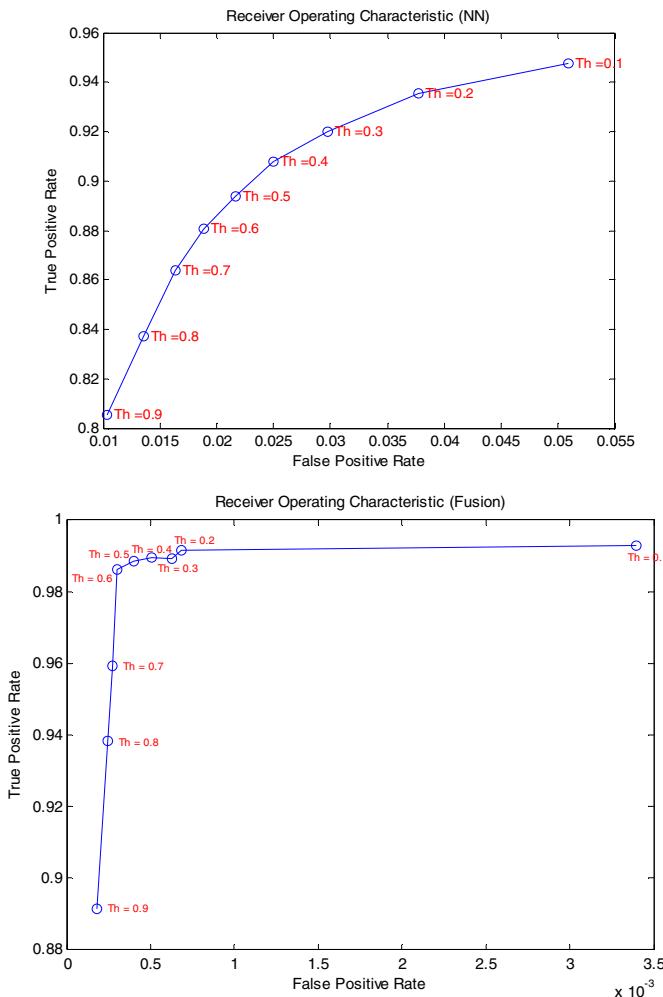


Fig. 6. ROC curves (top: NN only, bottom: proposed fusion method)

3.2 Evaluation

We have manually labeled the boundary between the sky and the ground, creating an accurate ground truth in the form of binary maps. For evaluation, we calculate the true positive (TP), false negative (FN), false positive (FP), and true negative (TN) rates. The total value of TPR and FPR rates for M images in the database is calculated as follows,

$$\begin{aligned} TP_{total} &= \sum_{k=1}^M (TP)_k \\ FN_{total} &= \sum_{k=1}^M (FN)_k \quad TPR = \frac{TP_{total}}{TP_{total} + FN_{total}} \quad FPR = \frac{FP_{total}}{FP_{total} + TN_{total}} \\ FP_{total} &= \sum_{k=1}^M (FP)_k \\ TN_{total} &= \sum_{k=1}^M (TN)_k \end{aligned}$$

The ROC curves for the NN classifier and the proposed fusion method are shown in Fig. 6. As it can be observed, the approach based on fusion shows better performance than the NN method. Also, the effect of different number of clusters (K) can be seen in Fig. 7. In this figure, five sets of ROC curves are presented for five different values of K . We observe that for $k \geq 10$ the performance does not change significantly but as the value of K gets larger, the implementation is more computationally expensive. Therefore, the value of K was set to 10. $TPR = 0.9886$ and $FPR = 4.0461 \times 10^{-4}$ are obtained for $K=10$. Fig. 8 shows segmentation results for some test images in our dataset.

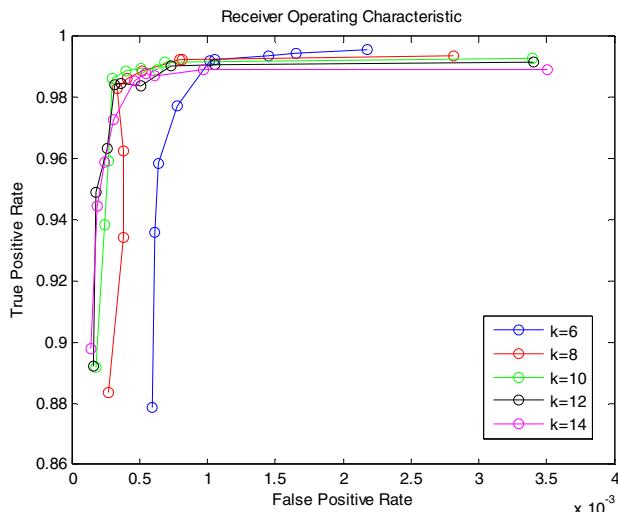


Fig. 7. ROC for different number of clusters in K-means clustering

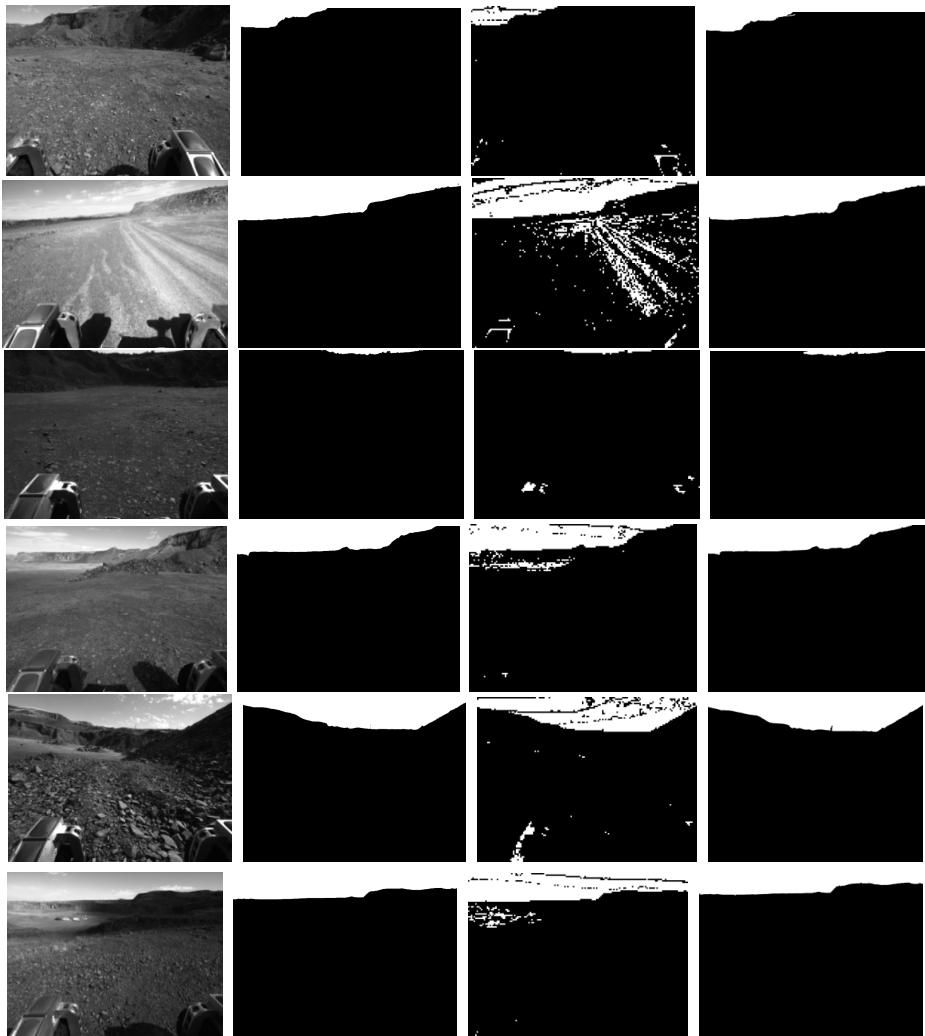


Fig. 8. Experimental results for six test images. From left to right: input images, ground truth maps, NN results, and results by the proposed method.

4 Conclusions and Future Work

The main contribution of this paper is the development of a highly accurate method for sky segmentation. This is an important task for NASA's rover tasks such as route planning, and path finding as well as for path planning and control of air- and ground unmanned vehicles. Due to the diversity of skylines and clouds, sky segmentation is a challenging task. We have proposed an automated framework for segmenting images into sky and non-sky regions by fusing K-means clustering with NN classifications. The method involves two post-processing steps which depend on certain parameters.

We have analyzed the performance of the method with regards to these parameters. For future work, we plan to optimize the feature set and implement the method in hardware. This will make the algorithm well-suited for the real-time applications.

Acknowledgment. This research was supported by NASA EPSCoR under cooperative agreement No. NNX10AR89A.

References

1. McGee, T.G., Sengupta, R., Hedrick, K.: Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation. In: International Conference on Robotics and Automation (ICRA 2005), pp. 4679–4684 (2005)
2. de Croon, G.C.H.E., Remes, B.D.W., De Wagter, C., Ruijsink, R.: Sky Segmentation Approach to Obstacle Avoidance. In: IEEE Aerospace Conference, pp. 1–16 (2011)
3. Fefilatyev, S., Smarodzinava, V., Hall, L.O., Goldgof, D.B.: Horizon detection using machine learning techniques. In: 5th International Conference on Machine Learning and Applications (ICMLA 2006), pp. 17–21 (2006)
4. Lie, W.-N., Lin, T.C.-I., Lin, T.-C., Hung, K.-S.: A robust dynamic programming algorithm to extract skyline in images for navigation. *Pattern Recognition Letters* 26, 221–230 (2005)
5. Kim, B.-J., Shin, J.-J., Nam, H.-J., Kim, J.-S.: Skyline Extraction using a Multistage Edge Filtering. *World Academy of Science, Engineering and Technology* 55, 108–112 (2011)
6. Xu, C., Lee, Y.J., Kuipers, B.: Ray-based Color Image Segmentation. In: Canadian Conference on Computer and Robot Vision, pp. 79–86 (2008)
7. Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M.: Vision-Guided Flight Stability and Control for Micro Air Vehicles. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 2134–2140 (2002)
8. Todorovic, S., Nechyba, M.C., Ifju, P.G.: Sky/Ground Modeling for Autonomous MAV Flight. In: International Conference on Robotics and Automation (ICRA 2003), pp. 1422–1427 (2003)
9. Seber, G.A.F.: *Multivariate Observations*. John Wiley & Sons, Inc., Hoboken (1984)
10. Widrow, B., Rumelhard, D.E., Lehr, M.A.: Neural networks: Applications in industry, business and science. *Commun. ACM* 37, 93–105 (1994)
11. Soh, L., Tsatsoulis, C.: Texture Analysis of SAR Sea Ice Imagery using Gray Level Co-Occurrence Matrices. *IEEE Transactions on Geoscience and Remote Sensing* 37(2), 780–795 (1999)
12. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features of Image Classification. *IEEE Transactions on Systems, Man and Cybernetics, SMC* 3(6), 610–621 (1973)
13. Clausi, D.A.: An analysis of co-occurrence texture statistics as a function of grey level quantization. *Can. J. Remote Sensing* 28(1), 45–62 (2002)

An Interactive Web Based Spatio-Temporal Visualization System

Anil Ramakrishna, Yu-Han Chang, and Rajiv Maheswaran

Department of Computer Science,
University of Southern California,
Los Angeles, CA

{akramakr,maheswar}@usc.edu, ychang@isi.edu

Abstract. Exploratory data analysis can be used to uncover several hidden but interesting patterns that may not be obvious otherwise. Spatio-temporal data offer numerous challenges for such analyses and several approaches have been proposed to address them. In this paper, we present an interactive web based system that can visualize large spatio-temporal data set on a standard web browser. This approach is applied to visualize a large spatio-temporal data set with 2.5 million records and several key patterns of the data are revealed in this process.

1 Introduction

The presence of compact global positioning systems (GPS) in modern smart phones and other hand-held devices has resulted in the collection of huge amounts of location aware data. In [1], the authors estimate that nearly 85 % of all information contains a spatial element. Several data sources also include a time stamp to signify the occurrence of important events. Visualizing such **Spatio-Temporal data** could reveal several interesting patterns [2] that are not obvious otherwise or may require extensive analysis to be uncovered. However, owing to complications arising due to their sheer size, such data are not fully utilized in data analysis [3]. Geographic Information Systems (GIS) provide a few means to analyze such data but they are not fully optimized towards large scale visualization of temporal data.

In this paper, we present a *software mashup* [4] [5] based approach to visualize Spatio-Temporal data by combining modern web frameworks to create a system that provides interactive visualization using a web browser.

The remainder of the paper is organized as follows: In section 2, we discuss related work and publications. In section 3, the technologies used with the system are introduced and in Section 4, the system design is discussed. Section 5 offers concluding remarks on the utility of our approach.

2 Related Work

Spatio-Temporal visualization has been a subject of interest for researchers for several years. Andrienko et al. [2] discussed several spatio-temporal animation

techniques such as map animation, filtering, visual querying, etc. giving real world examples that implement these. Guo et al. [3] presented a geo-visual analytic system that combined and leveraged well known visualization techniques in multivariate spatio-temporal data visualization. Wood et al. [5] explored the use of software mashups in data visualization and identified several key issues with this approach. Mashups are created by combining existing software technologies for rapid prototyping of various techniques. Slingsby et al. [4] also apply the mashup technique to combine tag clouds and tag maps, and evaluate them on Yahoo's tag map applet and Google Earth.

[6] discussed several java applet based online visualization tools along three themes of visualization: visualization of instant events, visualization of spatial movement and visualization of changing thematic data. Buja et al. [7] presented a taxonomy of visualization techniques in higher dimensions and implemented a few of them on their own home grown X-based visualization system called X-Gobi. Dykes et al. [8] discussed the applicability of a visualization tool called Location Trends Extractor(LTE) for spatio-temporal data. MacEachren et al. [9] combined technologies from Geo-Visualization and Knowledge Discovery methods and applied them on climate data visualization. Keim et al. [10] discuss the generic Visual Data Exploration paradigm: Overview first, zoom and filter, and then details-on-demand along with their own classification of visualization techniques.

3 Design

We present an online visualization system that uses modern web frameworks such as jQuery, D3, etc. to effectively visualize large scale data on a web browser. Though there are a few existing systems that leverage these frameworks for data visualization, they limit user interactions to a minimum and provide no means of navigation along the temporal attributes of the data set. One such example is the recently published article from Los Angeles times that reported the average time taken by the Los Angeles Fire Department (LAFD) units to arrive at locations of incidence. Associated with this report was a map based web page [11] that highlighted the average arrival time for various geographic regions within the Los Angeles city.

The work we present here extends the info-graphic from LA times by providing user interactions to control the visualization and to navigate along the temporal element of the data. We do this by providing the user a way to narrow down the visualization to various time resolutions such as weekday, month and year. The approach we present has many re-configurable elements that are generic enough to be used with any spatio-temporal data set.

3.1 Technologies

In this section we briefly describe the individual components of the system. Fig. 1 shows the data flow diagram among these components.

Geocoder. Geocoding is the process of converting addresses that are in human readable format such as street names, zip codes, etc. into geographic coordinates such as latitude and longitude. Several geocoders exist that provide an API for online geocoding or provide rules for offline geocoding [12]. We use the open source Perl module Geo::Coder::US available from the Perl archive cpan.org [13]. This module uses the Tiger/Line data set [14] released by the US census bureau to generate rules for the geocoding process. The module also provides several functions that use these rules to parse the input address and return the corresponding geographic coordinates.

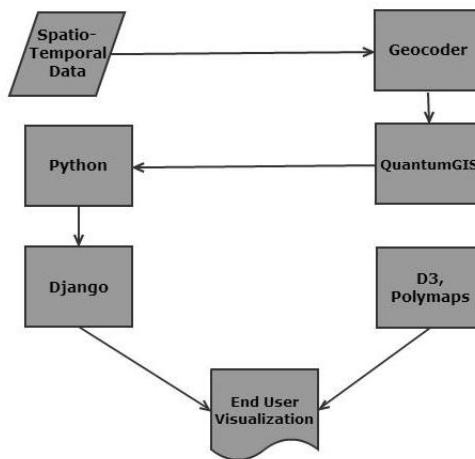


Fig. 1. Data flow diagram

QuantumGIS. QuantumGIS [15] is an open source Geographic Information System (GIS) developed by the Open Source Geospatial Foundation (OSGeo). It provides capabilities for geospatial data analysis, editing and visualization, among other functionalities. We use QGIS to *bin* the data into geographic hexagonal cells as shown in Fig. 2. Binning is the process of grouping geospatial data into adjacent cells. QGIS supports binning through the MMQGIS plugin [16].

Python. Python [17] is a general purpose programming language that supports functional, object-oriented and imperative programming styles. We use python to parse the data files and generate various statistics related to the data at various time resolutions.

Django. Django [18] is an open source web framework written in Python. It emphasizes the Model-View-Controller (MVC) software architectural pattern

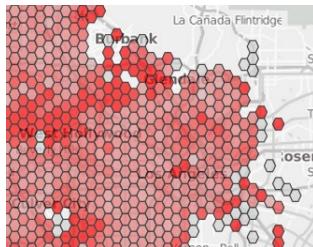


Fig. 2. Hexagonal grids overlayed on the map of Los Angeles city

and promotes rapid development and code re-usability. We use Django to develop the front end web application.

D3. D3 [19] is a JavaScript library that supports a data-driven approach to manipulating elements of the Document Object Model (DOM). It provides efficient methods for data-driven manipulation of the DOM with an emphasis on Web Standards. We use D3 to draw the geographic grid to manipulate the individual cells in the visualization.

3.2 Data

The data we have used comes from the call history of LAFD over five years. This data set contains approximately 2.5 million records with each entry including the type, location, etc. of the incident and several time stamps such as incidence time, arrival time, etc that correspond to the occurrence of various events.

The data records are geocoded in Perl using the Geo::Coder::US module. These records are then binned by QGIS into adjacent geographic cells which are saved separately. Various statistics are collected from the data records in each cell using Python. The process is repeated over various *time resolutions*, i.e. all data entries that belong to one particular weekday, month or year are grouped and the relevant statistics are collected. If the data contains multiple types, multiple sets of statistics are created, one for each type of the data, and stored. This is the only data specific component of the whole design and by suitably modifying these scripts, the approach can be applied to visualize any spatio-temporal data set. The statistics that are collected over all the time resolutions and cells are saved independently and made available to a web server in JSON format.

3.3 The Front End

The end user works with a web page consisting of a map focused on the locality of the data. The map is rendered using the polymaps framework [20] and the tiles are fetched from cloudmade servers [21]. The grids are visualized using the framework D3. The data driven design of D3 was especially accommodating for

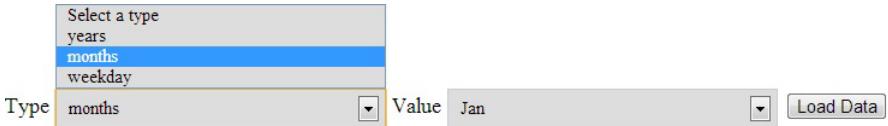


Fig. 3. Menu to control the time resolution

our needs. Each cell from the hexagonal grid structure was colored independently according to some relevant statistic of that cell using D3.

When loaded, the web page presents the user with a default grid layout overlaid on the map along with two menu items. In the first menu, the user selects a time resolution from one of weekday, month or year. Based on the user's selection, the second menu is populated with all the available time frames of that resolution as shown in Fig. 3. For example, if the user selects year, the second menu is populated with all the years that appear in the data set. Based on the selection from the second menu, a JSON file corresponding to that time frame is loaded from the server. These two menus provide the user a way to narrow down the visualization to one particular time frame and resolution. If the data contains multiple types, all the available types are listed as radio buttons. Based on the current selection of the radio button, the corresponding data set is visualized.

3.4 User Interactions

A slider is provided as shown in Fig. 4 through Fig. 6 to add interaction to the visualization. The slider was created using jQuery's slider plugin and has thirty divisions with each division representing time in minutes. By moving the slider, the color of each cell of the grid layout is changed based on the current value of the slider and some pre-calculated statistic from the data set corresponding to that geographic cell. With the LAFD dataset, we computed the percentage of records that take response time greater than or equal to the current value of the slider. For a given value of the slider, the percentage of records that had response time greater than the value of the slider is converted into a color using d3's scale function and visualized on the cells. Higher the number of records with time more than the slider's value, more darker the cells are, as shown in Fig. 4.

3.5 Discussion

Our observations have revealed few key patterns and confirmed a few previously established ones associated with LAFD's operations. For example, as shown in Fig. 4, the outer grids are generally darker than the inner grids. Since the intensity of color signifies the no of incidents with high response time, we can draw the intuitive conclusion that LAFD takes significantly higher times to respond to localities that lie in the boundary of the city as opposed to ones that are in the center. It is also clear that regions such as Bel Air, Los Angeles that have mountains and other difficult to navigate regions have higher average

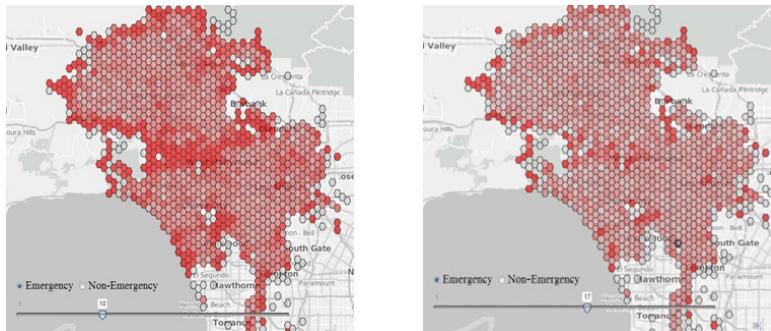


Fig. 4. Visualization at slider value=10(left) and slider value=17

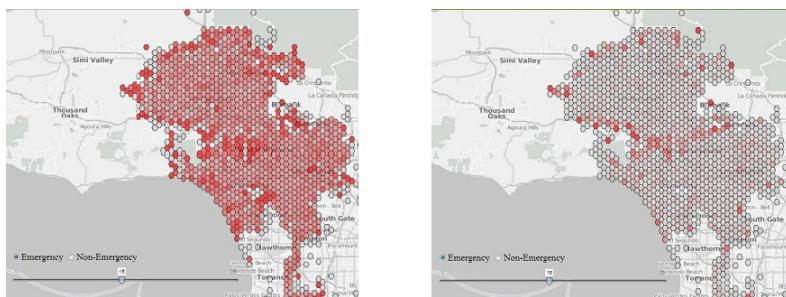


Fig. 5. Visualization for Sunday(left) and Thursday

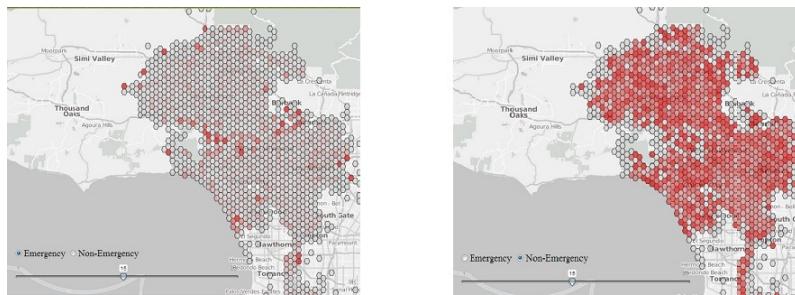


Fig. 6. Visualization for Emergency(left) and Non-Emergency incidents

response time when compared to other parts of the city. This is evident from the left figure in Fig. 4 in which Bel Air is located in the center dark cluster.

We also noticed that the average response time is much higher on a Sunday compared to the rest of the week as shown in Fig. 5. Non-emergency incidents showed relatively high average response times compared to emergency incidents as shown in Fig. 6. However they did not appear to show the same pattern of delayed response time in the boundary regions as the emergency incidents.

We did not notice any significant difference across the 12 months but we did notice a gradual increase in the number of regions with higher than average response times each year.

4 Conclusion

We presented a Spatio-Temporal Visualization system that combines open source technologies with modern web frameworks to create powerful web based visualizations. This system was applied to visualize the call history of LAFD and a few previously observed patterns in the data were identified, successfully demonstrating the effectiveness of the system. In addition, several interesting inferences were made about the operations of LAFD.

Future work includes providing a more continuous navigation across the time attributes using a slider and providing additional interactions to control other attributes of the data.

References

- [1] MacEachren, A.M., Kraak, M.: Research Challenges in Geovisualization. *Cartography and Geographic Information Science* 28(1) (2001)
- [2] Andrienko, N., Andrienko, G., Gatalsky, P.: Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing* 14(6), 503–541 (2003)
- [3] Guo, D., Chen, J., MacEachren, A.M., Liao, K.: A Visualization System for Space-Time and Multivariate Patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 1461–1474 (2006)
- [4] Slingsby, A., Dykes, J., Wood, J., Clarke, K.: Interactive Tag Maps and Tag Clouds for the Multiscale Exploration of Large Spatio-temporal Datasets. In: 11th International Conference on Information Visualization, pp. 497–504 (2007)
- [5] Wood, J., Dykes, J., Slingsby, A., Clarke, K.: Interactive Visual Exploration of a Large Spatio-temporal Dataset: Reflections on a Geovisualization Mashup. *IEEE Transactions on Visualization and Computer Graphics* 13(6), 1176–1183 (2007)
- [6] Andrienko, N., Andrienko, G., Gatalsky, P.: Visualization of spatio-temporal information in the Internet. In: 11th International Workshop on Database and Expert Systems Applications, pp. 577–585 (2000)
- [7] Buja, A., Cook, D., Swayne, D.F.: Interactive High-Dimensional Data Visualization. *Journal of Computational and Graphical Statistics* 5, 78–99 (1996)
- [8] Dykes, J.A., Mountain, D.M.: Seeking structure in records of spatio-temporal behaviour: visualization issues, efforts and applications. *Computational Statistics & Data Analysis* 43(4), 581–603 (2003)
- [9] MacEachren, A.M., Gahegan, M., Pike, W., Brewer, I., Cai, G., Lengerich, E., Hardisty, F.: Geovisualization for knowledge construction and decision support. *IEEE Computer Graphics and Applications* 24(1), 13–17 (2004)
- [10] Keim, D.A., Panse, C., Sips, M.: Information Visualization: Scope, Techniques and Opportunities for Geovisualization. *Exploring Visualization* (2005)
- [11] How Fast is LAFD where you Live (2012),
<http://graphics.latimes.com/how-fast-is-lafd>

- [12] List of available Geocoding software,
<http://geoservices.tamu.edu/Services/Geocode/About/GeocoderList.aspx>
- [13] The CPAN GeoCoder module,
<http://search.cpan.org/~sderle/Geo-Coder-US/US.pm>
- [14] United States Census Bureau. TIGER/Line data,
<http://www.census.gov/geo/maps-data/data/tiger.html>
- [15] QuantumGIS, <http://qgis.org>
- [16] MMQGIS, <http://michaelminn.com/linux/mmqgis>
- [17] Python, <http://www.python.org>
- [18] The Django Project, <https://www.djangoproject.com>
- [19] Data-Driven Documents, <http://d3js.org>
- [20] Polymaps, <http://polymaps.org>
- [21] Couldmade, <http://cloudmade.com>

One-to-Two Digital Earth

Ali Mahdavi Amiri, Faraz Bhojani, and Faramarz Samavati

University of Calgary, Department of Computer Science

Abstract. The digital Earth framework is a multiresolution 3D model used to visualize location-based data. In this paper, we introduce a new digital Earth framework using a cube as its underlying polyhedron. To create multiresolution, we introduce two types of 1-to-2 refinement. Having a smaller factor of refinement enables us to provide more resolutions and therefore a smoother transition among resolutions. We also suggest two indexing methods specifically designed for quadrilateral cells resulting from 1-to-2 refinement. We finally discuss the equal area spherical projection that we are using in this framework to model the Earth as a sphere partitioned to equal area cells.

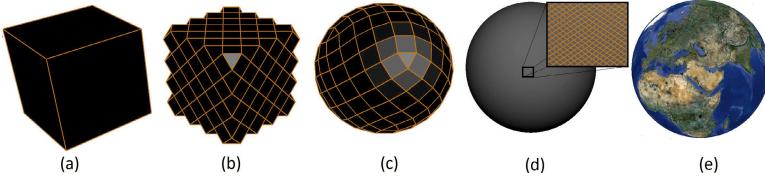


Fig. 1. (a) A cube (b) The cube after applying dual 1-to-2 refinement three times. (c) The refined cube is projected to the sphere using an equal area spherical projection. (d) The spherical cube at a high resolution. (e) The resulting spherical polyhedron is textured using a blue marble image. The blue marble texture is taken from [1].

1 Introduction

The digital Earth is a framework which represents the Earth as a multiresolution 3D model, and visualizes location-based data. Through this framework, users are able to zoom-in, zoom-out and analyze the data at different levels of detail. This framework has many applications in various fields such as computer science, cartography, and Geo-information systems. Google Earth is a particular well known and developed example of a digital Earth framework [2]. In this framework, the Earth is represented using traditional latitude longitude discretization of the Earth. As highlighted in [3], this representation creates issues regarding accuracy, replicability, and documentation.

It is possible to address these issues with a more regular representation of the Earth using a Geodesic Discrete Global Grid System (GDGGS) [4]. In GDGGS,

the Earth can be approximated by a polyhedron whose faces are refined and projected to the sphere (Fig 1). These projected faces are called cells. Using a data structure, location-based data such as borders of countries, height maps, and texture data, are associated with these cells. In fact, a GDGGS has five fundamental parts: the base polyhedron, type of cells, type of refinement, spherical projection, and the employed data structure.

There are several viable polyhedrons that can be used to represent the Earth. In our framework, we use a cube as the underlying polyhedron to represent the Earth. This is reasonable, since a cube is the only polyhedron that can provide quadrilateral cells. Quadrilateral cells are more compatible with hardware and display devices and they are also compatible with familiar Cartesian coordinates.

It is common to use 1-to-4 refinement to support multiresolution or a hierarchical representation for the cells of a cube. In this refinement, each cell is divided into four cells by inserting a vertex in the midpoint of edges. As a result, the number of cells grow exponentially by a factor of four from one resolution to the next. However, choosing a refinement with smaller factor results in a more gradual change in the resolution. In our framework, we use 1-to-2 refinement as it is the minimum factor of refinement. This results in a smooth transition between resolutions, creating an efficient representation for geographic features.

In order to have a spherical representation of the Earth, we need to project faces of a refined polyhedron to the Earth using a spherical projection. Spherical projections may create angular and areal distortion. Equal area projections preserve the area. This projection is commonly used for the digital Earth frameworks since it eases the analysis of data associated with cells. As a result, our framework incorporates an equal area spherical projection that is designed specifically for the grids forming on faces of a cube (see Section 3.3 for more details) [5].

To associate data with cells and to handle adjacency and hierarchical queries, a spatial data structure is needed. In digital Earth frameworks, data can be assigned at very high resolutions. As a result common spatial tree structures might not be efficient [6]. Indexing methods are designed to replace the tree structure. In this paper, we suggest two indexing methods that are adapted for 1-to-2 refinements of a cube.

Our contribution is to provide a new digital Earth framework using a spherical cube as the Earth's representation. We introduce 1-to-2 refinements to provide multiresolution among the cells with the slowest factor of growth. We then provide two indexing methods for the cells, resulting from the refinement, and we suggest an equal area projection.

In Section 2, we discuss the related work of our method. Our proposed framework is introduced in Section 3. We then provide some results and discussion in Section 4. Conclusion and future work are presented in Sections 5.

2 Related Work

The Earth's surface can be partitioned into regular cells using a method called Geodesic Discrete Global Grid System (GDGGS)[7]. As we discussed earlier,

GDDGS are distinguished based on their underlying polyhedron, type of cells, refinement, projection and their employed data structure [7]. We highlight some work related to each of these elements.

Underlying Polyhedrons: Different polyhedrons including platonic solids and truncated icosahedron have been used as an approximation of the Earth [8–11]. Among these polyhedra, spherical cubes have been traditionally used as the Earth representation [12] and they are still commonly used for spherical representations. Cubes are very popular since they can provide regular quadrilateral cells that are hardware-efficient, and adaptable with display devices, existing data structures, and familiar Cartesian coordinates. As a result, they are used as the base for a sphere in many applications such as terrain rendering, environmental mapping, game design, surface modeling, [13–16, 8] as well as the Earth representation [17, 8]. In this paper, we use a cube as an underlying polyhedron for a digital Earth framework.

Type of Cells: Cells of GDGGS can be hexagons, quads or triangles. For example, Dutton uses the triangular faces of an octahedron [10]. Some Digital Earth frameworks also employ hexagonal cells as the base cells (see [18] for a complete survey). However, as mentioned earlier, quadrilateral cells have been used more commonly. In this paper, we also use squares as the base cells of our proposed digital Earth framework.

Type of Refinement: Refinements are mostly applied to subdivision surfaces to create smooth graphical objects [19]. Refinement methods are also used to create levels of detail for a digital Earth framework. A 1-to- n refinement divides a cell with an area of A into cells with an area of $\frac{A}{n}$ where n is the *factor* of the refinement or *aperture* of the digital Earth framework [4].

It is common, for digital Earth frameworks that use a cube as the underlying polyhedron, to use 1-to-4 refinement. However, refinements with smaller factors are desirable for the digital Earth framework, since it is possible to provide more resolutions under the fixed maximum number of cells. 1-to-2 refinement provides the smallest factor of refinement among the refinements, therefore, it creates smooth transition among cells. In this paper, we introduce 1-to-2 refinement for constructing the resolutions.

Data Structure: To assign data to cells and handle necessary inquiries, a spatial data structure is required. The quadtree is one of the most common data structure for quadrilateral cells. To make quadtrees more efficient at high resolutions, some indexing methods have been proposed [20, 6]. Indexing methods can be constructed based on space filling curves, hierarchy of cells at successive resolutions, or a coordinate system defined for the cells [13, 20, 21]. We suggest two kinds of indexing methods based on the hierarchy and a defined coordinate system of cells resulting from 1-to-2 refinement on quadrilateral cells.

Spherical Projection: Digital Earth frameworks vary based on their employed spherical projection. Different projections such as conformal, gnomonic, or equal area can be used to represent the Earth [22]. When a spherical projection is used,

two types of distortion might appear: angular distortion and areal distortion. The projection that preserves the area is called equal area. This projection has been widely used for representing the Earth [17, 23, 5, 11]. Some equal area projections may reveal specific disadvantages. For example, they may create singularities at specific points [17], or iterative techniques, that slow down the handling of inquiries, are used to find its inverse relations [23, 24]. We suggest to use a spherical area projection that is specifically designed for cubes and has a closed form for both projection and inverse projection mappings [5].

3 Framework

In previous sections, we described the elements necessary for a digital Earth framework modeled by a Geodesic Discrete Global Grid System. As we discussed earlier, the underlying polyhedron of our framework is a cube and the cells are quadrilaterals. In this section, we describe other elements of our proposed framework. We first describe 1-to-2 refinement and explain some of its properties. Then, we discuss two possible indexing methods for such a framework. We eventually describe the projection that we use in our proposed digital Earth framework.

3.1 Refinement

Using the concept of lattices to analyze the behavior of refinement methods is very common [25]. To describe the refinement of our proposed method, we also use lattices. Consider a square regular lattice L_0 (see Fig 2(a)). In L_0 , each vertex is connected by four edges to its nearest neighbors.

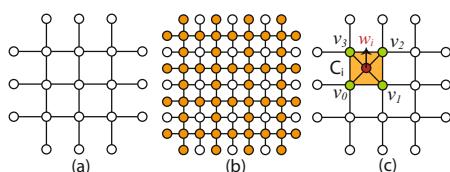


Fig. 2. (a) A portion of square regular lattice L_0 . (b) 1-to-4 refinement applied on L_0 . Orange circle are newly inserted vertices in L_0 . (c) Vertices v_0, v_1, v_2 , and v_3 create cell C_i . w_i is inserted in the midpoint of C_i .

The traditional 1-to-4 refinement is very common for quadrilateral cells (Fig 2(b)). However, as discussed earlier, having a refinement with lower factor is desirable. 1-to-2 refinements for quadrilateral cells have the smallest factor of refinement. Two types of 1-to-2 refinement are defined for L_0 . Consider vertex v_0 and three vertices v_1, v_2 , and v_3 making cell C_i (Fig 2(c)). We insert a vertex w_i in the midpoint of C_i and connect w_i to its nearest vertices v_0, v_1, v_2 , and v_3 (Fig 2(c)) and then discard the old edges. If this refinement is applied on all cells, lattices illustrated in Fig 3 are obtained. This refinement is used by $\sqrt{2}$ subdivision for smoothing graphical objects [26].

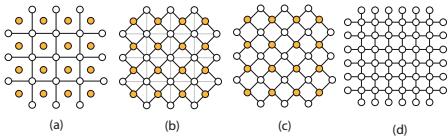


Fig. 3. (a) Inserting midpoints in all cells of L_0 . (b) Connecting midpoints to their closest vertices. (c) Removing old edges. (d) The lattice after two iterations of 1-to-2 refinement.

The other 1-to-2 refinement for quadrilaterals is defined by inserting new vertices in the midpoint of edges. Again consider lattice L_0 and cell C_i with vertices v_0, v_1, v_2 , and v_3 . These vertices form edges e_0, e_1, e_2 , and e_3 as illustrated in Fig 4 (a). To refine cell C_i , vertex w_i is inserted in the midpoint of edge e_i ($0 \leq i \leq 3$). Afterwards, w_i is connected to w_{i+1} (w_3 is connected to w_0) in order to make a new cell. Finally, all edges e_i and vertices v_i are discarded. Fig 4 illustrates steps of refining cell C_i .

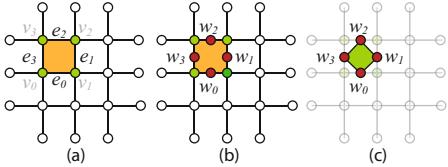


Fig. 4. (a) Edges creating cell C_i . (b) Inserting midpoints w_i at the midpoint of edges. (c) Connecting new vertices and creating a new cell.

Figure 5 illustrates application of 1-to-2 refinement on all cells of L_0 . This type of 1-to-2 refinement is also created if we apply simplest subdivision method on a regular grid [27]. If we apply this refinement twice the lattice illustrated in Fig 5(d) is obtained.

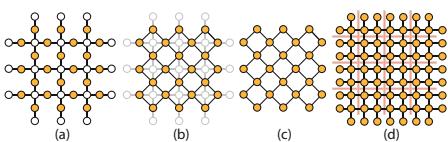


Fig. 5. (a) Inserting vertices at the midpoint of edges. (b) Connecting new vertices. (c) Discarding old vertices and edges. (d) Refining L_0 twice using 1-to-2 refinement. L_0 is illustrated in red.

To distinguish these 1-to-2 refinements, we call the first one *primal* (Fig 3) and the second one *Dual* (Fig 5). In dual 1-to-2 refinement, cells at two successive resolutions have the same midpoints. Such refinements are called *aligned* or *central place* [4]. These types of refinements have some advantages in compared to the alternative.

In GDGGS, the midpoints of cells at resolution r denoted by m_r are interpreted as a sample point of the entire cell. Accuracy, here means the error for representing points by cells, where error is the distance between m_r and a given point p . One of the advantages of dual 1-to-2 refinement is that increasing the resolution enhances the accuracy since the distance between m_r and p is decreased by increasing the resolution. This means $d_r \leq d_{r+1}$ where $d_r = \|p - m_r\|_2$. However, this characteristic is not guaranteed in some other refinements such as the

common 1-to-4 refinement. Fig 6 illustrates this scenario and compares primal 1-to-4 refinement and dual 1-to-2 refinement.

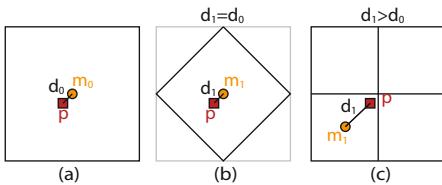


Fig. 6. (a) d_0 is the distance between m_0 , and point p illustrated by the red square. (b) After one iteration of dual 1-to-2 refinement, the accuracy is at least as the previous resolution. (c) After one iteration of 1-to-4 refinement, d_1 is bigger than d_0 . Increasing the resolution may decrease the accuracy in primal 1-to-4 refinement.

3.2 Indexing

Using indexing methods for handling spatial queries is common. These queries include determining the location of a cell, its neighbors and its position in the hierarchy of cells. Here, we suggest two possible indexing methods for dual 1-to-2 refinements. These indexing methods can be used for the primal 1-to-2 refinement with slight modifications. The first indexing method is similar to the hierarchy-based indexing designed for quadtrees [20] and hexagonal cells [4, 28]. The second proposed indexing is adopted based on the coordinates of cells at successive resolutions [29, 21, 4]. We call the first method, hierarchy-based indexing and the second one, coordinate-based indexing.

Hierarchy-Based Indexing Method. Hierarchy-based indexing methods are defined based on the relationships between cells at successive resolutions after the refinement. These indexing methods are very efficient in handling hierarchical queries. In our proposed indexing method, faces of the initial spherical cube are considered as the cells in resolution 0. In dual 1-to-2 refinement, for a cell C_r at resolution r , there is a cell C_{r+1} with the same midpoint (Fig 7(a)). C_{r+1} is called the *midpoint child* of C_r and C_r is called the parent of C_{r+1} . C_r has four other children whose midpoints are aligned with vertices of C_r (Fig 7(b)). These children are called *vertex children*. If cell C_r has index α at resolution r , its midpoint child has index $\alpha 0$ and four other children have indices αi , $1 \leq i \leq 4$, based on their position with respect to C_r . Fig 7(c) illustrates such an indexing for a cell at resolution r .

In hierarchy-based indexing, hierarchical access is handled by appending digits to a cell's index to access its children, or truncating a part of its index to access its parents. Neighbors of cells are found through a look-up table determining the algebra of such 1D indices. Vince [28] has provided such a look-up table for a similar indexing for hexagons resulting from refining an icosahedron. We can adapt it for the quadrilaterals resulting from the cube. In [28], an index is created using the elements of set $A = \{0, 1, 2, 3, 4, 5, 6\}$ since it indexes hexagonal cells

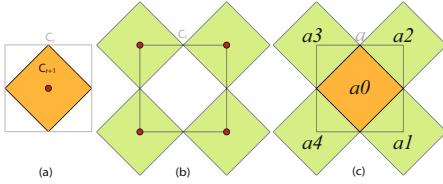


Fig. 7. (a) Cells C_r and C_{r+1} share the midpoint. (b) Vertex children of C_r are illustrated in green. (c) If C_r has index α , its children have indices as illustrated.

with neighbors in 6 directions. However, in our method this set is changed to $\Lambda = \{0, 1, 2, 3, 4\}$ and therefore the look-up table is modified accordingly. Note that the singularities of an icosahedron are pentagons but singularities of a cube are triangles; both are located at the vertices of the initial polyhedron. Triangles produced at the corners are indexed similar to quadrilaterals but they have three neighbors.

Coordinate-Based Indexing Method. Coordinate based indexing is another type of indexing method in which faces of polyhedrons are typically unfolded onto the plane. An index is then defined for each cell by snapping the vertices or midpoint of cells on the integer coordinates [21, 11, 29]. In this section, we introduce coordinate-based indexing for quadrilaterals resulting from dual 1-to-2 refinement applied on the faces of a cube.

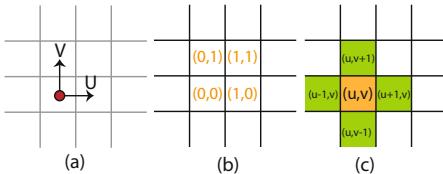


Fig. 8. (a) Coordinate system defined for cells of L_0 (b) Indices of cells. (c) Cell (u, v) (orange) and its neighbors (green).

To describe this indexing, we again use the concept of lattices for representing the connectivity of cells. Consider L_0 as previously defined in Section 3.1. We take two vectors with a 90° difference connecting midpoints of two adjacent cells as the unit vectors (U and V) making a coordinate system for cells (Fig 8). We can allow the midpoint of any cell to be the origin of the coordinate system. With respect to this coordinate system, midpoints of cells can get integer coordinates. These coordinates (u, v) are considered as the index of cells.

In this indexing, neighborhood finding queries are handled by simple algebraic operations. A cell with index (u, v) , has neighbors $(u+1, v)$, $(u, v+1)$, $(u, v-1)$, and $(u-1, v)$ (Fig 8(c)). As well as neighborhood finding, the indexing scheme must be capable of handling hierarchical access queries. Therefore, we need to index cells at various resolutions and find a hierarchical access relation between cells at various resolutions.

We use L_r for the lattice obtained from applying r times of 1-to-2 refinement on L_0 . To index L_1 , we take the same origin as the one chosen for L_0 . We define a coordinate system for L_1 by considering the vectors connecting midpoints of two

adjacent cells as the main vectors of the coordinate system (Fig 9(a)). We index cells according to this new coordinate system. To distinguish cells at different resolutions, we use a subscript indicating the resolution. As a result, a cell with index $(u, v)_r$ is at resolution r and is u and v steps away from the origin, in the direction of the U and V axes respectively.(see Fig 9(b)).

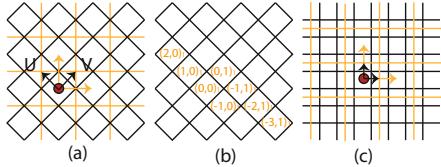


Fig. 9. (a) Coordinate system defined for cells of L_1 . L_0 is illustrated in orange. (b) Indices of cells. (c) L_2 and its coordinate system. L_0 and its coordinate system are illustrated in orange. ● is the origin.

As shown in Fig 9, it is possible to choose coordinate systems for L_2 aligned with L_0 . Similarly, we can choose coordinate systems for L_3 aligned with L_1 . We can extend this property for further resolutions and pick a coordinate system for L_r aligned with L_0 or L_1 depending on if r is even or odd respectively. This property enables us to establish simple hierarchical relationships.

To establish a hierarchical relationship, we explain the transition from a parent to its children. To do this, we find the index of the midpoint child of a cell with index $(u, v)_r$. In fact, this task is equivalent to finding corresponding vectors of $(u, v)_r$ at resolution $r + 1$. Therefore, we can find the corresponding vectors of $(1, 0)_r$ and $(0, 1)_r$ at resolution $r + 1$ and multiply by u and v respectively.

Assume that r is even, therefore as illustrated in Fig 10 (a), $(1, 0)_r = (-1, 1)_{r+1}$ and $(0, 1)_r = (1, 1)_{r+1}$. Thus, $(u, v)_r = (v - u, u + v)_{r+1}$. For odd, we have the same relations (Fig 10(b)). Note that for any r , $(1, 0)_r = (2, 0)_{r+2}$ and $(0, 1)_r = (0, 2)_{r+2}$. Therefore $(u, v)_r = (2u, 2v)_{r+2}$. As a result, we can generalize hierarchical relationships for transitioning from resolution r to resolution $r + k$ by using Equation 1. $(s, t)_{r+k}$ denotes the index of the midpoint child of cell $(u, v)_r$ after k iterations of refinement.

$$(s, t)_{r+k} = \begin{cases} 2^k(u, v)_r & \text{if } k \text{ is even} \\ 2^{k-1}(v - u, u + v)_r & \text{if } k \text{ is odd} \end{cases} \quad (1)$$

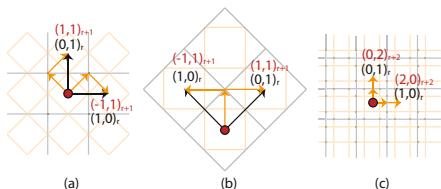


Fig. 10. (a) Equality of vectors when r is even. (b) Equality of vectors when r is odd. (c) Equality of vectors after two resolutions and r is even. ● is the origin.

The transition from a fine cell to a coarse cell is simply the reverse of the process. To apply this indexing method on a cube, we can index each face of

the cube individually and handle the connectivity queries of boundaries using an edge based method that captures the connection of faces [29]. When a cube is refined by a dual 1-to-2 refinement, a triangle is formed at each corner. These triangles are indexed the same as quadrilaterals. The only difference is that they have three neighbors and three vertex children instead of four in the case of quadrilaterals. We can also flatten the polyhedron and index all faces in a 2D domain [21]. This way, triangles and boundary faces are traceable at all resolutions using hierarchical operations. Both methods are applicable for this application. We have implemented both and the results are comparable.

3.3 Projection

To form the spherical cells of our digital Earth framework, we use a spherical equal area projection. An equal area projection is a mapping from a domain Ω to another domain Δ while preserving the area. In our employed projection, Ω and Δ represent a cube and a unit sphere respectively, where both are centered at the origin and have the same area (the cube's edge has length $a = \sqrt{2\pi/3}$). In this projection, Δ is divided into six equal partitions by finding the intersection of planes $z = \pm x$, $z = \pm y$, and $x = \pm y$ with Δ . Fig 11(c) illustrates one partition of Δ in black.

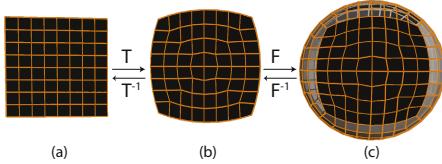


Fig. 11. Steps of the spherical projection. Points on a face f of Ω (a) are projected onto a curved square (b) and then projected onto a partition of the unit sphere Δ (c).

The main idea of the projection is to map each face f of Ω to a partition of Δ . To this end, an intermediate domain, called curved square, is used. As a result the projection has two steps. First f is projected to a curved square on the tangent plane of Δ , parallel to f , using an equal area bijection called T . Afterwards, the curved square is mapped to a partition of Δ using inverse Lambert Azimuthal equal area projection called F (Fig11). For the detailed discussion and derivations of mappings, you can refer to [5].

This projection is suggested due to its property of area preservation and its closed form definitions. One can use a different projection with different properties (such as conformality) based on application needs. Our proposed refinement and indexing methods are not dependent on the employed projection.

4 Results and Discussion

In this Section, we present some results of our framework. We illustrated the results of applying dual 1-to-2 refinement on the cube in Fig 1. Primal 1-to-2 refinement can also be used for our proposed framework (see Fig 12).

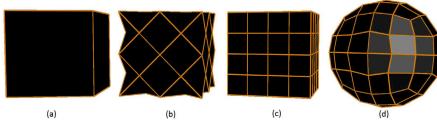


Fig. 12. (a) A cube. (b), (c) Applying primal 1-to-2 refinement three and four times respectively. (d) Projection of (c) to the sphere.

A digital Earth framework should support visualizing raster datasets such as images (Fig 1). In addition, it should support vector data. Vector datasets are coordinate-based data models that represent geographic features. Such features are typically provided as points, lines, and polygons. Fig 13(a) illustrates a vector dataset representing the boundaries of different countries. In GDGGS, each point is approximated by a cell enclosing the given point. Therefore, a feature can be represented by a sequence of cells.

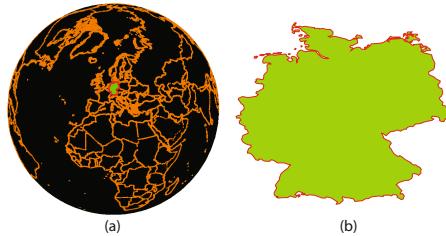


Fig. 13. (a) Vector data representing boundary of countries. Germany is highlighted in Green. (b) Boundary of Germany is zoomed. Data is taken from [30] and rendered in our framework.

To see the details of geographic features, data must be represented at different resolutions. For example, the boundary of Germany is highlighted in green in Fig 13 (a). However, as illustrated in Fig 13 (b) many details of this feature is not visible in the low resolution model. As a result, we need to have a resolution dependent accuracy. Accuracy, again means minimal error for representing feature points by cells.

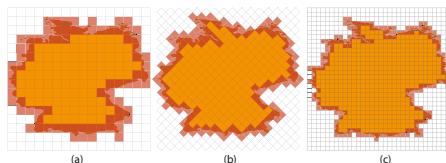


Fig. 14. Cells representing the boundary of germany with errors (a) $\approx 35Km$, (b) $\approx 25Km$, and (c) $\approx 17.5Km$

In comparison to other frameworks [31, 28, 17], our framework provides an efficient representation for features by using the minimal factor of refinement. Consider that we want to have cells representing the boundary of Germany with an error less than $\approx 28Km$. Fig 14(a) illustrates cells representing the boundary with an error $\approx 35Km$ in red. To get the desired accuracy, our framework at the next resolution can approximate the feature points with 114 cells and an error $\approx 25Km$ which is good enough for our purpose (Fig 14(b)). However, if we use

a 1-to-4 refinement (Fig 14(c)), although we have satisfied the required accuracy (error less than $\approx 28Km$), we need to save and render 162 cells (48 more cells than 114.) As a result, using 1-to-2 refinement, we can save 42% of the cells for representing the boundary of Germany as an example of a geographic feature.

5 Conclusion and Future Work

In this paper, we have introduced a new Digital Earth framework modeled by a Geodesic Discrete Global Grid System. This framework uses a cube as its base polyhedron. We use two types of 1-to-2 refinement, which is the minimum factor of refinement. These refinements are used to create cells to discretize the cube's surface. To project such cells, we suggest a projection with closed forms, both for projection and its inverse. We also provide two types of indexing methods to handle hierarchical and neighborhood finding operations.

This framework can be used as the base of many other applications aiming to visualize location based information. Therefore, in this aspect, there are many directions for enhancing the features of the framework. However, there are two future works relevant to the framework's structure. The spherical projection that we are using should be compared with other alternatives such as Snyder projection in terms of time to report the exact difference. Moreover, the angular distortion of the used projection should be calculated and compared with other projections and polyhedrons.

References

1. Map Projections, <http://www.progonos.com/furuti>
2. Google Earth, <http://earth.google.com>
3. Goodchild, M.F., et al.: Next-generation digital earth. *Proceedings of the National Academy of Sciences* (2012)
4. Sahr, K., White, D., Kimerling, A.J.: Geodesic discrete global grid systems. *Cartography and Geographic Information Science* 30, 121–134 (2003)
5. Rosca, D., Plonka, G.: Uniform spherical grids via equal area projection from the cube to the sphere. *J. Computational Applied Mathematics* 236, 1033–1041 (2011)
6. Samet, H.: Foundations of Multidimensional and Metric Data Structures. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers Inc., San Francisco (2005)
7. Goodchild, M.F.: Discrete global grids for digital earth. In: *Proceedings of 1st International Conference on Discrete Global Grids 2000* (March 2006)
8. Cozzi, P., Ring, K.: 3D Engine Design for Virtual Globes, 1st edn. CRC Press (2011)
9. Wickman, F.E., Elvers, E., Edvarson, K.: A system of domains for global sampling problems. *Geografiska Annaler. Series A, Physical Geography* 56, 201–212 (1974)
10. Dutton, G.H.: A Hierarchical Coordinate System for Geoprocessing and Cartography. *Lecture Notes in Earth Sciences Series*. Springer (1999)
11. Sahr, K.: Location coding on icosahedral aperture 3 hexagon discrete global grids. *Computers, Environment and Urban Systems* 32, 174–187 (2008)

12. Chan, F., O'neill, E.: Feasibility study of a quadrilateralized spherical cube earth data base. Technical report, EPRF, Silver Spring, Md. Computer Sciences Corporation. Environmental Prediction Research Facility (1976)
13. Cignoni, P., Ganovelli, F., Gobbetti, E., Marton, F., Ponchio, F., Scopigno, R.: Planet-sized batched dynamic adaptive meshes (p-bdam). In: Proceedings of the 14th IEEE Visualization, VIS 2003, pp. 147–155. IEEE Computer Society (2003)
14. Greene, N.: Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications* 6, 21–29 (1986)
15. Compton, K., Grieve, J., Goldman, E., Quigley, O., Stratton, C., Todd, E., Willmott, A.: Creating spherical worlds. In: ACM SIGGRAPH Sketches, SIGGRAPH 2007. ACM (2007)
16. Grimm, C.M.: Simple manifolds for surface modeling and parameterization. In: Proceedings of the Shape Modeling International, pp. 237–244 (2002)
17. Alborzi, H.: Geometric issues in spatial indexing. Master's thesis, University of Maryland, College Park (2006)
18. Sahr, K.: Hexagonal discrete global grid systems for geospatial computing. *Archives of Photogrammetry, Cartography and Remote Sensing* 22, 363–376 (2011)
19. Cashman, T.J.: Beyond catmull-clark? a survey of advances in subdivision surface methods. *Comput. Graph. Forum* 31, 42–61 (2012)
20. Gargantini, I.: An effective way to represent quadtrees. *Commun. ACM* 25, 905–910 (1982)
21. Mahdavi-Amiri, A., Samavati, F.: Connectivity maps for subdivision surfaces. In: GRAPP/IVAPP, pp. 26–37 (2012)
22. Grafarend, E.W., Krümm, F.W.: Map projections: cartographic information systems. Springer (2006)
23. Snyder, J.P.: An equal area map projection for polyhedral globes. *Cartographica* 29, 10–21 (1992)
24. Harrison, E., Mahdavi-Amiri, A., Samavati, F.: Analysis of inverse snyder optimizations. *Transactions on Computational Science* 16, 134–148 (2012)
25. Ivriessimtzis, I.P., Sabin, M.A., Dodgson, N.A.: A generative classification of mesh refinement rules with lattice transformations. *Comput. Aided Geom. Des.* 21, 99–109 (2004)
26. Li, G., Ma, W., Bao, H.: $\sqrt{2}$ subdivision for quadrilateral meshes. *Vis. Comput.* 20, 180–198 (2004)
27. Peters, J., Reif, U.: The simplest subdivision scheme for smoothing polyhedra. *ACM Trans. Graph.* 16, 420–431 (1997)
28. Vince, A., Zheng, X.: Arithmetic and fourier transform for the pyxis multiresolution digital earth model. *Int. J. Digital Earth* 2, 59–79 (2009)
29. Peters, J.: Patching catmull-clark meshes. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, pp. 255–258 (2000)
30. Thematic Mapping API, <http://thematicmapping.org/>
31. PYXIS Innovation, <http://www.pyxisinnovation.com>

Storygraph: Telling Stories from Spatio-temporal Data

Ayush Shrestha, Ying Zhu, Ben Miller, and Yi Zhao

Georgia State University

Abstract. A major task of spatio-temporal data analysis is to discover relationships and patterns among spatially and temporally scattered events. A most common analytic method is to plot them on a 3D chart with latitude, longitude and time being the three dimensions. The first drawback of this technique is that it fails to scale well when there are thousands of concentrated events since they suffer from cluttering, occlusion and other limitations of 3D plots. Second, it is hard to track the time component if the events are clustered in a region. To overcome these, we present a novel 2D visualization technique called Storygraph that provides an integrated view of location and time. Based on Storygraph, we also present storylines which show the movement of the characters over time. Finally, we present two case studies to demonstrate the effectiveness of the Storygraph.

1 Introduction

One of the major goals of spatio-temporal data analysis is to discover the relationships and patterns among scattered spatio-temporal events. However, it is difficult to construct a visualization that integrates spatial, temporal and other data dimensions. Dimensions other than time and space include type of event, characters involved and other descriptive data about the event for example in a military log, other dimensions would include number of casualties, type of event, actions taken etc. In this paper, we propose an interactive 2D visualization technique called *Storygraph* to address this problem.

A Storygraph is composed of two parallel vertical axes and a horizontal axis. The two parallel vertical axes represent latitude and longitude. The horizontal axis represents the time. Each geographic coordinate or location is represented as a line, called the *location line*, connecting the corresponding latitude and longitude on the two vertical axes in the Storygraph. Events occurring at different times in that location are plotted as markers on the location line. By geometry, the location lines of two locations close to each other on the map, are also close in the Storygraph. Hence, we can say that the Storygraph presents a way to analyze an event in its proximal geographic and temporal context. The shape, size and colors of the markers are used to represent other dimensions in the data beside space and time. In this paper, we assume that all the events in the dataset has latitude, longitude and timestamp.

In many cases, datasets often have characters associated with events. This is particularly common in case of military logs where a task force is deployed at one location for a period of time, then moved to next and so forth. For such datasets, a common analysis is to track the movement of these characters and generate space-time paths. For this purpose, we also introduce storylines based on Storygraph. Storylines are poly lines drawn by connecting all the events in which the character was involved. The storylines can be compared with the space-time path from time-geography [1] which shows the movement of the characters in space and time. To further facilitate the analysis, our implementation of the Storygraph also has a synchronized and interactive map view.

2 Related Work

Our analysis of related work focuses on the previous visualization methods for temporal, spatial and spatio-temporal data. Works on time series or temporal data visualization have tried to visualize storylines, but none integrated time, location and characters [2] [3] [4] [5] [6] [7] [8] [9] [10].

In many visualization methods, spatial and temporal data are visualized in separate views. Many authors have proposed methods to synchronize the temporal data with the spatial data. For example, authors in [11] [12] used small multiple views to link spatial data with temporal data. However it cannot accommodate a small temporal data scale. Jern et al use color coding to link temporal data with spatial data [13]. Using color coding to link temporal data with spatial data is not intuitive. Colors do not have a natural order, while timeline is sequential. Maciejewski et al. use interactive synchronized temporal and spatial views [14]. Interactive synchronized spatio-temporal views fail to present the continuous correlation between the spatial and temporal data. To address the problems of 2D integrated views, some researchers have proposed 3D integrated visualization of spatio-temporal data [15][16][17][18]. An immediate advantage of 3D integrated view is that the time graph does not occlude the 2D map. However, it is difficult to align time data with location in 3D and is hard to read and compare the data value in the vertical dimension.

Most techniques mentioned above in this section either focus more on visualizing time or more on visualizing location. Few visualizations which contains both fail to scale well. Our technique addresses these issues and produces a simplified scalable visualization containing time, location and other dimensions as discussed in the previous section.

3 Method

In this section, we first present the mathematical model of the Storygraph. We then describe storylines, the map view and the user interactions.

3.1 Storygraph

The Storygraph, is a 2D diagram consisting of two parallel vertical axes $V_\alpha \subset \Re$ and $V_\beta \subset \Re$ and an orthogonal horizontal axis $H \subset \Re$. All three of the axes, as in Cartesian graphs, are unbounded at both ends. The values in the axes are ordered in ascending order: from left to right in horizontal axis and bottom to top in vertical axes. In this paper, vertical axes V_α and V_β represent the x and y coordinates of a point on a plane such as latitude and longitude. The horizontal axis, H , represents time. Thus a point plotted on Storygraph, which shall be referred to as *event* in the rest of the paper will have at least three dimensions: parallel coordinates and a timestamp.

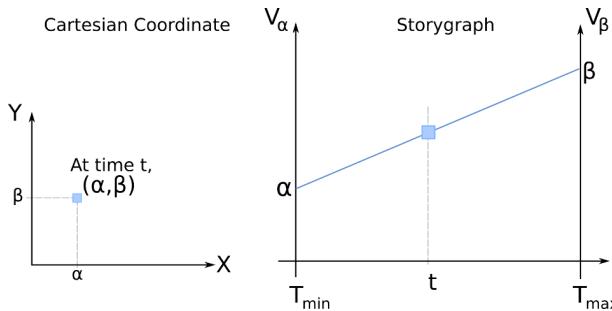


Fig. 1. Left: A point in the Cartesian coordinate system such as a location on a map at time t , Right: Same point represented in a Storygraph with parallel coordinates and timestamp

For any event occurring at (α, β) in time t as shown in Figure 1, our algorithm first draws a *location line* by connecting the points on the two axes, $\alpha \in V_\alpha$ and $\beta \in V_\beta$. The algorithm then returns the point on this line at time t .

The function $f(\alpha, \beta, t) \rightarrow (x, y)$ which maps an event to the 2D Storygraph plane can be formally written as follows:

$$y = \frac{(\beta - \alpha)(x - T_{min})}{T_{max} - T_{min}} + \alpha \quad (1)$$

$$x = t \quad (2)$$

where T_{min} and T_{max} are the maximum and minimum timestamps within the dataset.

Figure 1 illustrates how a location on a regular 2D map, coded with a Cartesian coordinate, is presented in the Storygraph. Equation 1 and 2 is used to convert a location on a regular map to a Storygraph plane, and vice versa. As seen from the equations, such conversion is very efficient and can be done in real time. Because of this, we are able to create an interactive and synchronized map view along side Storygraph as seen in Figure 4.

3.2 Storyline

In this paper, we describe storyline as a series of events associated with a character. Hence, drawing storylines require a dataset having characters in addition to latitude, longitude and timestamps. A character can be a person, a group, or an institution that is identified with an event. Given a set of events associated with a character, the storyline is a poly line drawn by connecting these events sequentially. Multiple storylines can be created for multiple characters as shown in Figure 3. This allows users to explore the relationship and interactions between the characters, which are often difficult to extract from the textual description. Storylines based on storygraph are especially helpful when visualizing the movement of characters as users can see how different characters converge and/or diverge at certain events.

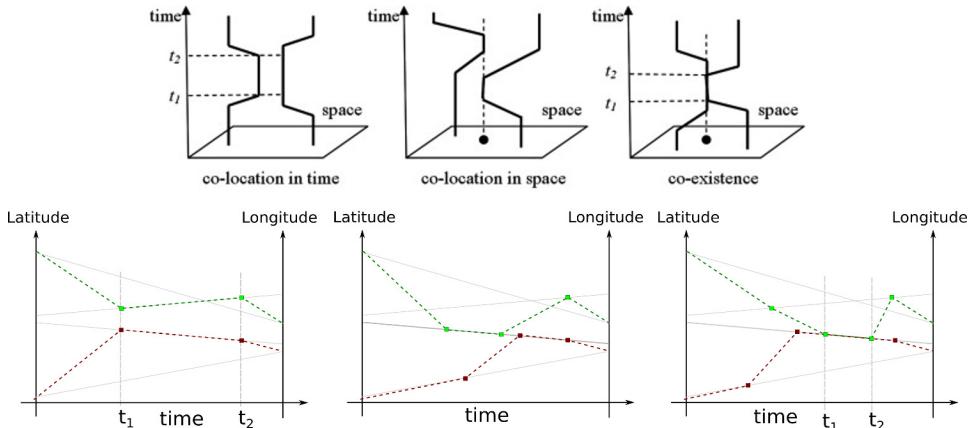


Fig. 2. Comparison of the time-space paths using Time-Geography, adapted from [19] with storylines drawn using Storygraph. Starting from the left, the top figures directly compare to the bottom figure - first showing co-location in time, second showing co-location in space and third showing co-existence

It is useful to compare storylines in Storygraph with the well established space-time path concept in Time-geography [1] because they share similar characteristics. Both techniques attempt to present temporal and spatial data in an integrated way. However, because Time-geography is a 3D visualization, it suffers from occlusion, cluttering, and difficult depth perception as more data points are plotted. On the other hand, Storygraph is a 2D visualization and therefore does not have any occlusion or depth problem. As can be seen from Figure 2, Storygraph can present co-location in time or space and co-existence better than Time-geography.

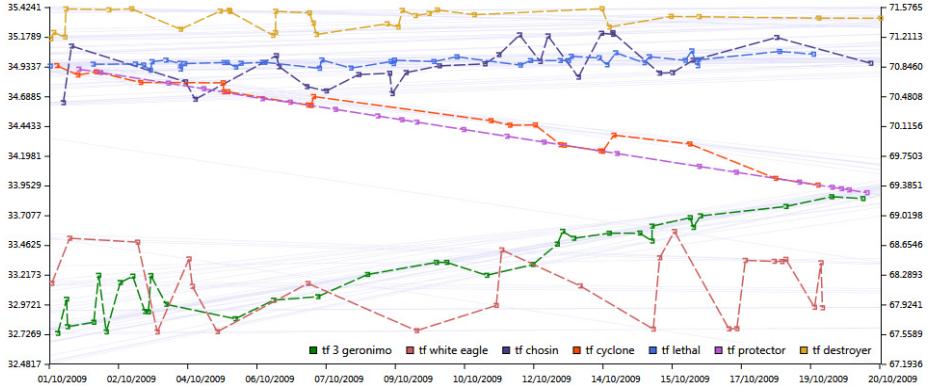


Fig. 3. Storylines of seven Afghanistan based combat units from October 1 - 20, 2009

Figure 3 shows the storylines of seven Afghanistan based combat units from October 1-20, 2009. It can be seen that units 'tf protector' and 'tf cyclone' were at the same location during October 5 - 7 and also during 12 - 14. Similarly, 'tf white eagle' travelled a wider range of locations than the rest, 'tf lethal' travelled a lesser range and 'tf protector' didn't travel at all. Hence, besides showing the mobility of the units and how the unit interacted with other units, the greatest strength of the Storylines based on Storygraph is it also shows the time period during which the unit was most mobile or least mobile.

3.3 Synchronized Map View

It is easy to show locations on a cartographic map but difficult to show temporal information. To take advantage of both Storygraph and map, we provide a synchronized map view to supplement the main Storygraph. Any update in the Storygraph such as zooming, panning and filtering is automatically reflected in the map view, and vice versa. Users can zoom in and out in the map view, or select a rectangular region, and the Storygraph is updated accordingly. As discussed earlier, the conversion of data points from Storygraph to map is really efficient, therefore the synchronization of Storygraph and map happens in real-time. Figure 4 shows a subset of events from the Afghanistan war data plotted on the Storygraph and on the map. The map view helps users analyze the geographical proximity of events, and the Storygraph help users examine the relationship among events in both space and time. Together, the Storygraph and the map view provide a platform for comprehensive analysis of spatio-temporal data.

3.4 User Interactions

Storygraph allows users to hide location lines to reduce data points cluttering and occlusion as shown in Figure 5. The hiding of the location lines is useful when

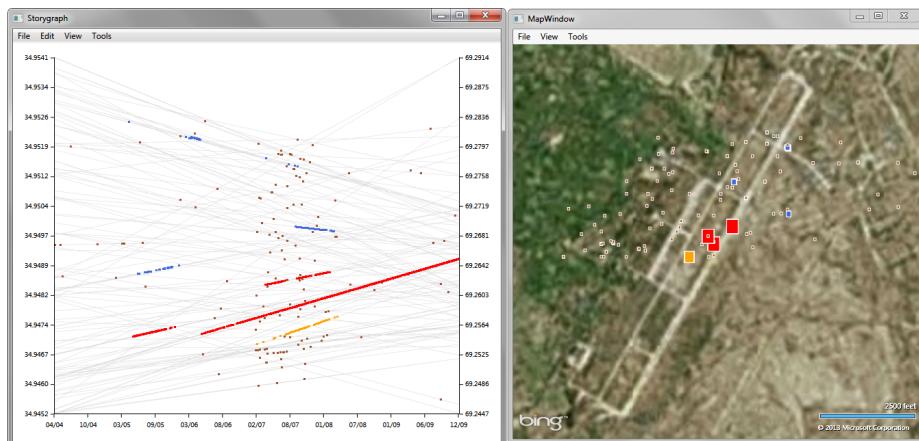


Fig. 4. Storygraph and the synchronized aerial map view showing a subset of events that took place during the Afghanistan war. When a user clicks on a data point in the map view, the corresponding data point in Storygraph is highlighted, and vice versa.

the dataset is dense and clusters of data points are easier to identify without the location lines. On the other hand, location lines may help identify trends in a sparse data plot as in Figure 4 (left).

In Storygraph, users can zoom in or out along the time line to examine the data on different time scales. Users can also filter the data by range of latitude, longitude, and time. If a user wants to find the details about an event, he or she can right click on a data point to open a pop-up window that shows the event details.

4 Case Studies

We tested our technique using two datasets: Wikileaks Afghanistan War Diary (War Diary) and data related to Laos from NARA's Southeast Asia Data Base of Records About Air Sorties Flown in Southeast Asia (Laos UXO). Our visualizations revealed meaningful patterns in the data for which we were able to generate hypotheses.

4.1 Afghanistan War Log (2004-2010)

The War Diary comprises U.S. military significant activity reports from Afghanistan during the period 2004-2010 [20]. Consisting of approximately 60K highly structured records, the data provides a rigorously categorized and unprecedented look at the daily conduct of war. This dataset consists more than twenty dimensions besides latitude, longitude and event date out of which we plotted the

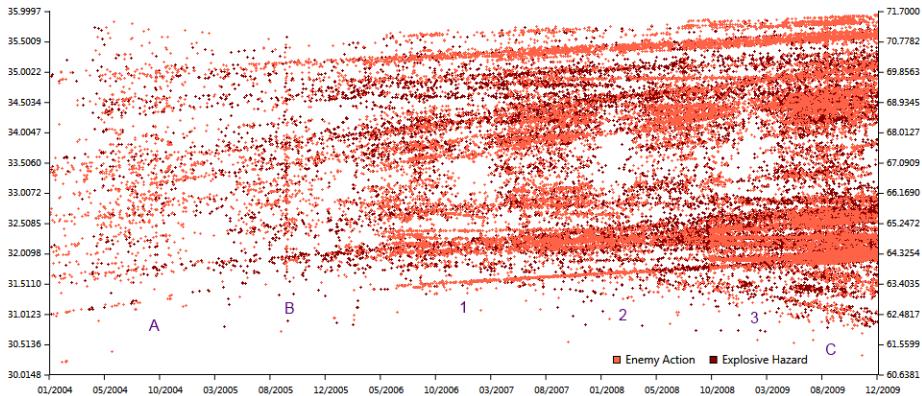


Fig. 5. Storygraph showing all the events during Afghanistan war from 2004 to 2009 across different regions of the country. More obvious patterns are marked by A-C and 1-3.

"Type of action" limiting to "Enemy Action" and "Explosive Hazard" in Figure 5. Figure 6 shows the "Aggregate number of deaths" along with events. We also present storylines of seven Afghanistan based combat units in Figure 3

For Figure 5, we observed two sets of distinct patterns marked by A-C and 1-3. The vertical bands marked by A-C are formed due to the events clustering between Aug-Nov 2004, Jul-Oct 2005 and near Sept 2009. They indicate widespread and coordinated "Enemy Action" and "Explosive hazards" events at a very short period of time. By correlating these dates to broader events in Afghanistan, we discovered that these clusters were proximal to elections; our hypothesis is that the incidence of violence, and therefore the number of activity reports, increases during elections.

Numbers 1-3 in the figure shows a periodicity of voids, or a lack of reports in those areas. These "holes" seem to appear around the end of the year. From location lines, we found that the geographic location by these holes correspond to a section of the Kabul-Kandahar highway, a key portion of Afghanistan's national road system and a main target of attacks. The Storygraph visualization shows that there have been regular quiet periods for that section of the highway around the end of 2005, 2006, 2007, and 2008. Since the data set does not extend beyond December 2009, we are unable to confirm if the pattern repeated in 2009. To our knowledge, this pattern has not been identified or discussed in any published report. Figure 6 shows the death toll as the war progressed. We observed that the frequency of death rose with time. A more interesting observation however, as in case of Figure 5, is the periodicity in the number of deaths. With this we can directly correlate the number of deaths with the number of explosive hazards and enemy actions. Although this is not new information, it demonstrates Storygraphs ability to help identify significant event patterns.

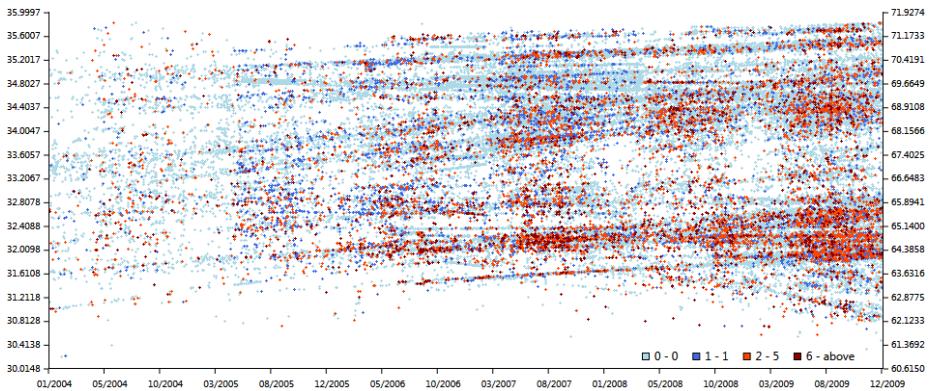


Fig. 6. Storygraph of Afghanistan war showing periodicity in the total number of deaths. The lighter shades denote events with less than one death reported while darker shades denote events with more than one death.

4.2 Unexploded Ordnance Laos (1964-1973)

An ongoing problem in war torn regions is the persistence of unexploded ordnance. Established by the Lao government with support by various NGOs, the Lao National Unexploded Ordnance Programme (UXO Lao) addresses this ongoing problem so as to reduce the number of new casualties and increase the amount of land available for agriculture. This data set details bombings in Laos by the USAF during the period of the war, 1968-1975 and is also a military, descriptive log of the bombings. It consists of 60,000 reports documenting approximately 2 million tons of ordnance [21]. Figure 7 shows the graph obtained from this dataset.

The Storygraph obtained from this dataset shows three distinct patterns.

1. The bombings intensified during October 1969 - Mar 1970.
2. From the start of the data to June, 1966 show cluster of events that form distinct lines. These lines signify that the bombings were focused at certain locations at almost regular intervals.
3. The bombings reduced drastically after March 1972 when most US troops left Vietnam.
4. The vertical bands above the 03/1970, 03/1971 and 02/1972 show the periodicity in the bombings.
5. The patterns of bombing data are interspersed with periodic bands of white.

We have two hypothesis for those voids: either they could mean that bombings were paused during that range of time, it could also mean that the data correlated to those raids during that period was redacted from the set. Like much military data, classified operations such as those by special forces are frequently not contained within general operation activity reports. It is beyond the scope of this paper to test these hypotheses. However, it demonstrates the ability of Storygraph to discover patterns and form hypotheses.

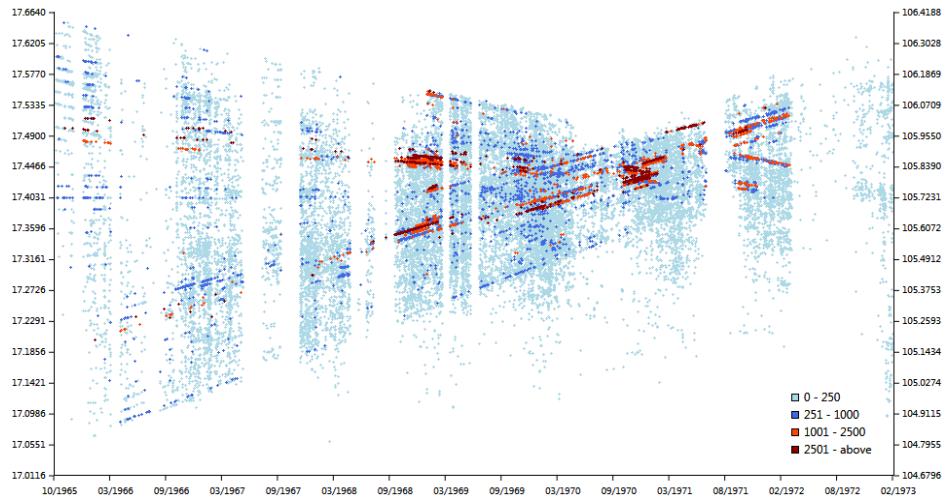


Fig. 7. Storygraph generated from the Laos dataset. Each event corresponds to a bombing. The color signifies the number of bombs dropped as shown in the legend.

5 Conclusion

In this paper, we have presented a novel visualization technique, Storygraph, that provides an integrated view containing locations and time. We also presented storylines based on Storygraph. Storygraph addresses the problems in previous 3D and integrated 2D spatio-temporal data visualization to minimize glyph occlusion and reduce glyph cluttering. This improved design help users better correlate events on both the spatial and temporal dimensions. Storylines help track the movement of characters and the interactions between them. In the future we plan to extend the visualization to incorporate uncertainty in location and time.

References

1. Hägerstrand, T., et al.: Time-geography: focus on the corporeality of man, society, and environment. *The Science and Praxis of Complexity*, 193–216 (1985)
2. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visualizing time-oriented dataa systematic view. *Computers and Graphics* 31, 401–409 (2007)
3. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics* 14, 47–60 (2008)
4. Fisher, D., Hoff, A., Robertson, G., Hurst, M.: Narratives: A visualization to track narrative events as they develop. In: *Proceedings of IEEE Symposium on Visual Analytics Science and Technology*, pp. 115–122 (2008)

5. Vrotsou, K., Johansson, J., Cooper, M.: Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Transactions on Visualization and Computer Graphics* 15, 945–952 (2009)
6. Javed, W., McDonnel, B., Elmqvist, N.: Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics* 16, 927–934 (2010)
7. Gschwandtner, T., Aigner, W., Kaiser, K., Miksch, S., Seyfang, A.: Carecruiser: Exploring and visualizing plans, events, and effects interactively. In: *Proceedings of IEEE Pacific Visualization Symposium (PacificVis)*, pp. 43–50 (2011)
8. Geng, Z., Peng, Z., Laramee, R., Walker, R., Roberts, J.: Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics* 17, 2572–2580 (2011)
9. Zhao, J., Chevalier, F., Pietriga, E., Balakrishnan, R.: Exploratory analysis of time-series with chronolenses. *IEEE Transactions on Visualization and Computer Graphics* 17, 2422–2431 (2011)
10. Krstajic, M., Bertini, E., Keim, D.: Cloudlines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics* 17, 2432–2439 (2011)
11. Asgary, A., Ghaffari, A., Levy, J.: Spatial and temporal analyses of structural fire incidents and their causes: A case of toronto, canada. *Fire Safety Journal* 45, 44–57 (2010)
12. Plug, C., Xia, J.C., Caulfield, C.: Spatial and temporal visualisation techniques for crash analysis. *Accident Analysis and Prevention* 43, 1937–1946 (2011)
13. Jern, M., Franzen, J.: “Geoanalytics” - exploring spatio-temporal and multivariate data. In: *Proceedings of Tenth International Conference on Information Visualisation*, pp. 25–31 (2006)
14. Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., Cleveland, W., Grannis, S., Ebert, D.: A visual analytics approach to understanding spatiotemporal hotspots. *IEEE Transactions on Visualization and Computer Graphics* 16, 205–220 (2010)
15. Gatalsky, P., Andrienko, N., Andrienko, G.: Interactive analysis of event data using space-time cube. In: *Proceedings of the Eighth International Conference on Information Visualisation*, pp. 145–152 (2004)
16. Tominski, C., Schulze-Wollgast, P., Schumann, H.: 3D information visualization for time dependent data on maps. In: *Proceedings of the Ninth International Conference on Information Visualisation*, pp. 175–181 (2005)
17. Adrienko, G., Adrienko, N., Mladenov, M., Mock, M., Politz, C.: Identifying place histories from activity traces with an eye to parameter impact. *IEEE Transactions on Visualization and Computer Graphics* 18, 675–688 (2012)
18. Landesberger, T.V., Bremm, S., Andrienko, N., Adrienko, G., Tekusova, M.: Visual analytics for categoric spatio-temporal data. In: *Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST 2012)*, pp. 183–192 (2012)
19. Shaw, S.L., Yu, H.: A gis-based time-geographic approach of studying individual activities and interactions in a hybrid physical virtual space. *Journal of Transport Geography* 17, 141–149 (2009) *ICT and the Shaping of Access, Mobility and Everyday Life*
20. Guardian.co.uk: Afghanistan war logs 2003-2010 (2010)
21. UXO-LAO: Lao national unexploded ordnance programme annual report 2007 (2008)

Organizing Visual Data in Structured Layout by Maximizing Similarity-Proximity Correlation

Grant Strong¹, Rune Jensen², Minglun Gong¹, and Anne C. Elster²

¹ Dept. of Computer Sci., Memorial Univ. of Newfoundland, St. John's, NL, Canada

² Dept. of Computer and Info. Sci., Norwegian Univ. of Sci. and Tech., Trondheim, Norway

Abstract. The goal of this work is to organize visual data into structured layouts such that proximity reflects similarity. The problem is formulated as maximizing the correlation between the dissimilarities among the data and their placement distances in the structured layouts. An efficient greedy-based coarse-to-fine algorithm is proposed to compute near optimal data placements. The qualities of such placements are verified using an enumeration-based algorithm that is capable of computing the exact solution for small-scale problems. Results on different datasets show that data items with low dissimilarity being placed near one another whereas those with high dissimilarity are placed apart. This facilitates users to understand the relations among the data items and to locate the desired ones during the browsing.

1 Introduction

The sheer volume and varying types of data available today drives the need for the ability to automatically arrange data like articles, images, videos, webpages, etc., for visual presentation. In this paper, we focus on organizing such data in a structured way so that the proximity of data items reveals their similarities, facilitating users to understand the relationships between the data and to locate desired elements from it.

Here structured layouts refer to multi-dimensional groups of discrete cell locations. 2D rectangular grid is a useful example. Its two degrees of dimensional freedom allow for multi-dimensional data to be flexibly arranged and it is very straightforward to present on computer screen. 3D cubes are a logical extension that is harder to visualize since occlusion must be dealt with. Trees and graphs are also possible alternatives when particular cell structures are desired.

This paper formally defines the problem of mapping data into a structured layout as a problem of maximizing the proximity-similarity correlation of data items. The mapping solution that maximizes the correlation is referred as the Max Correlation Map (MCM). An efficient coarse-to-fine approach is proposed to compute close approximations for the MCM of a given problem.

2 Related Work

How to visualize a set of data items based on their relative similarity has been widely studied. Many approaches show similarity or connectedness using graphs or networks

[2, 3, 18, 21]. In a graph, each node represents an item or a cluster of data and edges represent the connections between them. This paper focuses on structured grids with cells which can only contain one item. Similar items are forced into the surrounding cells so that proximity reflects similarity.

In the past, relative positions for data items have been generated by using dimensionality reduction techniques [1, 8, 11, 13, 20]. For example, being a set of manifold learning techniques, Multidimensional Scaling (MDS) [1, 11] is often used to map data onto 2D or 3D visual space. The classical version attempts to minimize a linear error function computing the total difference between the dissimilarities in the high-dimensional space and the respective distances in the reduced space. There are a host of variants to the MDS, for instance Sammon’s mapping [13] which applies the same strategy to a non-linear error function. Locally Linear Embedding (LLE) [12] and Stochastic Neighbor Embedding (SNE) [4] are two other popular methods of high to low dimensional embedding. The basic idea is that a set of weights (LLE) or neighborhood probabilities (SNE) can be derived from the high dimensional data points to guide a low dimensional embedding. The assumption is that locally, in the high dimensional space, the data points can be described by those in their neighborhoods. These notable dimension reduction techniques, along with others, have been established for many years and have been reviewed extensively [19].

A Self-Organizing Map (SOM) is a network of interconnected weight vector units. The weight vectors are trained by repeatedly modifying them under the influence of similar input vectors [9, 10]. In the end, each input vector is mapped to the position of the unit with the closest weight vector [15, 17]. A downside to this technique is that it requires input data to have a vector representation. The possibility also exists for units to be shared, so post-processing must be done to generate an occlusion free mapping.

Differing from the approaches just described, our approach organizes data into cell-based structured layouts. Instead of allowing a data item to be placed at an arbitrary location, the proposed approach assigns it to a unique cell in the output structure. This transforms what is normally a continuous optimization problem into a discrete one. The most closely related technique is the Self-Sorting Map (SSM) [14]. It is a multi-dimensional pseudo-sorting algorithm. Like the proposed approach, the SSM can run on a set of input data items as long as their dissimilarities can be measured. Nevertheless, the SSM maximizes correlation indirectly by minimizing local dissimilarity. This makes it faster but not always globally accurate if a strong layout is not established in the early stages.

The approach being proposed diverges from the SSM in that it optimizes on the correlation coefficient directly, leading to organizations of higher global correlation. It has the ability to handle all types of data and layouts that the SSM can, with the addition of being able to place items in structured layouts with some cells fixed, enlarged, or removed without changes to the core algorithm.

3 Problem Formulation

Given a dataset Ω , assume there is a dissimilarity function δ defined such that for any two data items s and t in Ω , if $s = t$ then $\delta(s, t) = 0$, otherwise $\delta(s, t) \geq 0$. Now assume there is a structured layout Γ that contains n cells, where $n = |\Omega|$.

We define a mapping $M: \Omega \rightarrow \Gamma$ as a function that assigns each data item in the dataset Ω to a cell in the structured layout Γ . M is occlusion-free if no more than one data item can be assigned to any given cell in Γ .

The objective is to find an occlusion-free map, i.e., the MCM, where the proximity of data items in the structured layout correlates with the similarities among them. More formally, such a map is one that attempts to maximize the following Pearson correlation coefficient:

$$\rho(M) = \frac{1}{\sigma_\psi \sigma_\delta} \cdot \frac{1}{|\Omega|^2} \sum_{\forall s, t \in \Omega} (\psi(M(s), M(t)) - \bar{\psi})(\delta(s, t) - \bar{\delta}) \quad (1)$$

where $\psi(\cdot, \cdot)$ is the distance between two cells, $M(\cdot)$ returns the cell to which a given item is mapped, and where $\bar{\psi}$ and $\bar{\delta}$ are the means and σ_ψ and σ_δ are the standard deviations of the distance and dissimilarity measures, respectively.

Different distance functions, such as Euclidean distance or geodesic distance, can be used to define $\psi(\cdot, \cdot)$. If we fix the structured layout Γ and the dataset Ω , then the means and standard deviations are constant. Equation (1) can be simplified by defining normalized cell distance and data dissimilarity measures as:

$$\Psi(u, v) = \frac{\psi(u, v) - \bar{\psi}}{\sigma_\psi}, \Delta(s, t) = \frac{\delta(s, t) - \bar{\delta}}{\sigma_\delta}$$

Thus, in simplified form, Equation (1) becomes:

$$\rho(M) = \frac{1}{|\Omega|^2} \sum_{\forall s, t \in \Omega} \Psi(M(s), M(t)) \cdot \Delta(s, t) \quad (2)$$

Since the mapping function $M: \Omega \rightarrow \Gamma$ is one-to-one for an occlusion-free map, its inverse function $M^{-1}: \Gamma \rightarrow \Omega$ exists. Equation (1) can also be expressed as:

$$\rho(M) = \frac{1}{|\Gamma|^2} \sum_{\forall u, v \in \Gamma} \Psi(u, v) \cdot \Delta(M^{-1}(u), M^{-1}(v)) = \frac{1}{|\Gamma|} \sum_{u \in \Gamma} \rho_u \quad (3)$$

where ρ_u evaluates how well the data item stored in a given cell u correlates with the remaining cells. Here ρ_u is referred to as the cell correlation score and is computed as:

$$\rho_u(M) = \frac{1}{|\Gamma|} \sum_{v \in \Gamma} \Psi(u, v) \cdot \Delta(M^{-1}(u), M^{-1}(v)) \quad (4)$$

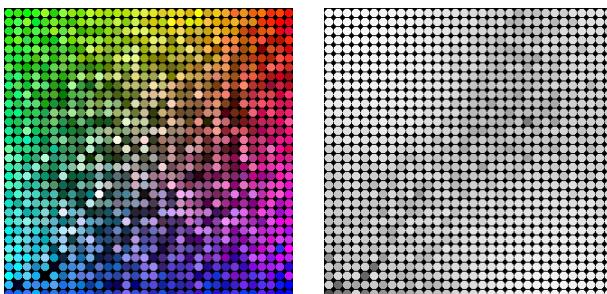


Fig. 1. Left: An organization of 1024 random Lab color vectors. Right: Visualization of the correlation scores at different cells, where high intensity represents high correlation score. In this instance the lower left corner is the worst correlated area.

Figure 1 visualizes the correlation scores for different cells. By definition (3), the overall correlation of a given map is equal to the average of all cell correlation scores.

4 Mapping Search Algorithms

In this section we describe our two algorithms for searching for MCMs or *near MCMs*. The first algorithm computes *near MCMs* efficiently using a greedy method under a coarse-to-find scheme, the second is an enumeration-based MCM. How to compute the *true MCMs* for small-scale problems to aid in the evaluation of the performance of the greedy method is also discussed.

4.1 Greedy-Based MCM Search (G-MCM)

Our goal here is to maximize correlation directly without any intermediate structures. The greedy algorithm starts with generating an initial map by randomly assigning data to empty cells. It then tries to improve the mapping through swapping data items in two selected cells. If such a swap increases the total correlation of the map, it will be preserved; otherwise, it will be reverted. However, directly applying data swapping in such a greedy manner can lead to local optimal solutions. To overcome this, a coarse-to-fine scheme is used which splits the map into large blocks in the beginning. Data is swapped between different blocks guided by increasing correlation. No regard is given to the exact cell position into which the data item is being moved to in the blocks. Starting with large blocks limits the number of possible solutions so that a greedy search has a good chance of finding a near-optimal solution at that block level. Once such a solution at a coarse level has been obtained, the process moves to a finer level to refine the result. In this way, the coarse-to-fine search method builds a solution incrementally, avoiding the combinatorial explosion of possible solutions that the enumerated approach contends with, while mitigating the potential for local minima that greedy algorithms are known for by not imposing tight restrictions on cell positions at coarse block levels.

Correlation Update after Data Swap

Given a mapping M with its correlation $\rho(M)$ evaluated, we first discuss how to efficiently compute the correlation $\rho(M')$ for the updated map M' , which is obtained by swapping two data items in M . The key to partially updating ρ lies in the fact that it can be described as the sum of the cell correlations, as shown in Equation (3). If two items, s in cell u and t in cell v , are swapped then every cell correlation needs to be updated to account for the change. This is done in two phases: First, the differences in correlation between the old item placement and the new item placement must be applied to all other cells, and secondly, the cell correlations for u and v need to be recomputed; see Equation (5) below. Equation (3) can then be used as usual to compute the updated correlation $\rho(M')$ after the swap. By updating this way, only the necessary changes to cell correlations are carried out and the recalculation of all cell correlations on every swap is avoided.

$$\rho'_w = \begin{cases} \rho_w + \Psi(w, u)(\Delta(M^{-1}(w), t) - \Delta(M^{-1}(w), s)) \\ + \Psi(w, v)(\Delta(M^{-1}(w), s) - \Delta(M^{-1}(w), t)) & \text{If } w \in \Gamma \setminus \{u, v\} \\ \sum_{i \in \Gamma} \Psi(w, i) \cdot \Delta(M^{-1}(w), M^{-1}(i)) & \text{If } w \in \{u, v\} \end{cases} \quad (5)$$

where ρ'_w is the new cell correlation score for cell w .

Coarse-to-Fine Processing Scheme

As shown in Figure 2, to perform data swapping in a coarse-to-fine manner, the cells in the map are initially grouped into a few of large blocks. Each of the blocks is further split into smaller blocks in the next block level, until the finest block level where each block contains only one cell. At a given block level, the cell distance function is adjusted as: $\psi'(u, v) = \psi(C(u), C(v))$, where function $C(\cdot)$ returns the center cell for the block that u belongs to. Once the distance measures are adjusted, the corresponding mean $\bar{\psi}'$, standard derivation $\sigma_{\psi'}$, and normalized distance $\Psi'(u, v)$ are updated as well.

Such a change in the distance measure ensures that $\Psi'(u, v) = 0$ as long as u and v belonging to the same block. Consequently, the exact cell position where a data item is placed inside a given block does not matter, allowing the swap-based optimization to focus on moving data into the proper block.

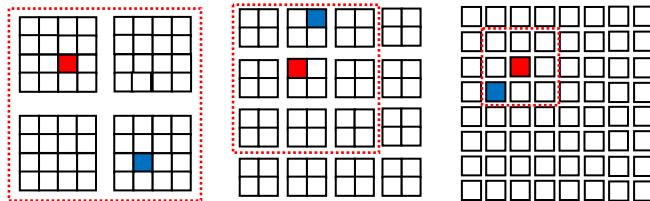


Fig. 2. Illustration on how data are swapped in three levels: 2×2 blocks (left), 4×4 blocks (middle), and 8×8 individual cells (right). The neighborhood for the same red cell is marked with red dashed box, within which its swap partner (shown in blue) is randomly selected.

To move data at a given block level, the algorithm cycles through every cell u in the map and looking for a random cell in u 's block neighborhood to swap the data with. This neighborhood is a 3×3 block window with the block of u at the center, as seen in Figure 2. After the swap the correlation is updated using Equation (5). If the correlation increases the change is kept, otherwise it is reverted.

Fixed Item Conditions

The design of the MCM algorithm makes the implementation of fixed items (or alternatively removed cells in the layout) straightforward. The most significant consideration is that dissimilarities of comparisons with fixed items are weighted. This prioritizes the relationships of the fixed items with other non-fixed ones. Fixed items can also occupy a larger region than standard items without modification to the fundamental design. Figure 3 shows an example of how a relatively large fixed item can affect the organization of colors.

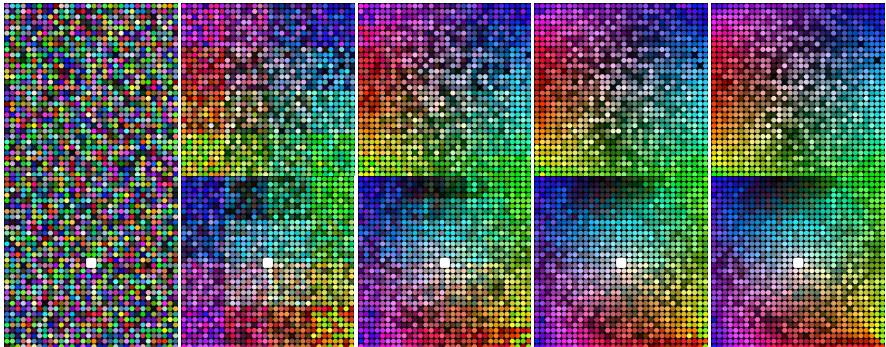


Fig. 3. Top row: An organization of Lab color vectors. Bottom row: A fixed item is introduced.

4.2 Enumeration-Based MCM Search (E-MCM)

Here we discuss an approach that enumerates all possible mappings $M: \Omega \rightarrow \Gamma$ by “brute force”. This is useful evaluation of the G-MCM. To enumerate the mappings we start with an empty map and fill the cells one by one recursively. Once all the cells are filled, we have an occlusion-free mapping solution and its correlation is evaluated. The recursive process backtracks and continues forward again with a different items until all possible mappings are evaluated and the one with maximum correlation, i.e., the MCM, is found. Even though it is possible to find the true MCM with this approach, the computational cost is too high for practical use, as shown in Table 1. In the rest of this section we discuss a novel way of reducing the computational cost by pruning the recursive tree of possible mappings being searched.

Table 1. The number of solutions needs to be evaluated with and without pruning. Without pruning, the solution space becomes impractical to search for maps as small as 4×4 .

Map Size	# of Solutions Evaluated w/o Pruning	# of Solutions Evaluated w/ Pruning
3×3	362,880	6,043
4×3	479,001,600	66,099
4×4	20,922,789,888,000	21,380,134

Pruning

If we consider the above brute force search process as depth-first traversal of a decision tree with all possible mapping solutions represented as leaves in the tree, we can prune a branch during the traversal if we know none of the leaves in the branch has higher correlation than a known solution. To apply pruning, we need a way to compute an upper bound correlation score $\bar{\rho}$ for a given partially filled map. The upper bound is computed by assuming that each unfilled cell u is occupied by an ideal data item, whose dissimilarity to any other cell v is identical to the distance between u and v . Hence, the upper bound of the cell correlation score for a given cell u is:

$$\bar{\rho}_u(M) = \begin{cases} \sum_{v \in F} \Psi(u, v) \cdot \Delta(M^{-1}(u), M^{-1}(v)) + \sum_{v \in \Gamma \setminus F} \Psi(u, v)^2 & \text{if } u \in F \\ \sum_{v \in \Gamma} \Psi(u, v)^2 & \text{if } u \in \Gamma \setminus F \end{cases} \quad (6)$$

where F is a subset of Γ containing occupied cells.

Summing together $\bar{\rho}_u(M)$ gives us the upper bound correlation score $\bar{\rho}(N)$ for a partially filled map N . Intuitively, placing a data item s into a given cell u means replacing the imaginary perfect data in u with the real data s , which will leave or lower the correlation. Hence, if the upper bound $\bar{\rho}(N)$ is already smaller than the correlation $\rho(M)$ of an already found full map M , there will be no point in conducting any further search beyond the partial solution N . This allows the corresponding subtree to be pruned during the depth first traversal.

In practice, we first compute a solution M using the G-MCM discussed in Section 4.1 and use the value $\rho(M)$ as the pruning threshold. Once a better solution M' ($\rho(M') > \rho(M)$) is found during the traversal, the value $\rho(M')$ is used instead. As shown in Table 1, pruning can greatly reduce the number of solutions needs to be evaluated. It is also worth noting that during the traversal when a data item is filled into an existing map N , the upper bound $\bar{\rho}(N')$ for the new map N' can be efficiently updated based on the original $\bar{\rho}(N)$ using the approach discussed in Section 4.1.

5 Results

In this section, the G-MCM is compared against its closest alternative the SSM in terms of quality, speed and consistency using both artificial and real data. Both implementations are in Java, with the E-MCM utilizing multi-threading. An Intel Xeon E5540 CPU with 4 cores at 2.53 GHz was used.

5.1 Comparison among G-MCM, E-MCM, and SSM

We start the tests using small datasets, which the E-MCM can solve practically. Figure 4 shows the mappings found by the E-MCM (the true MCM), G-MCM (near optimal solutions), and SSM algorithms. The results confirm that near optimal solutions found by the greedy approach are very close to the true MCM. It is obvious that the E-MCM approach is impractical as it takes a staggering amount of time to search for the best 4×4 map. Overall the G-MCM is the winner, with respect to the speed/correlation tradeoff in this case.

To evaluate the performances of different algorithms over large datasets, a set of pseudo-color vectors is constructed, whose color values are derived from grid coordinates. As shown in Figure 5, at the initial layout, the dissimilarity between any two color vectors is identical to the distance between their corresponding cells, and hence, the correlation value is “1”. While this problem is simpler than true 3D color vectors, it still demonstrates the better performance of the G-MCM over the SSM.

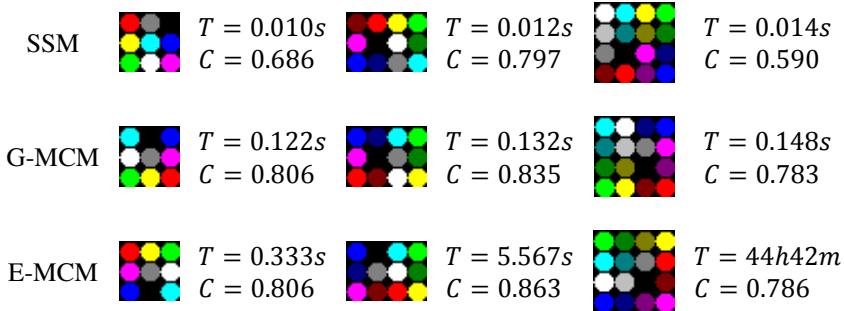


Fig. 4. A comparison between the E-MCM and G-MCM. Times and correlations are given.

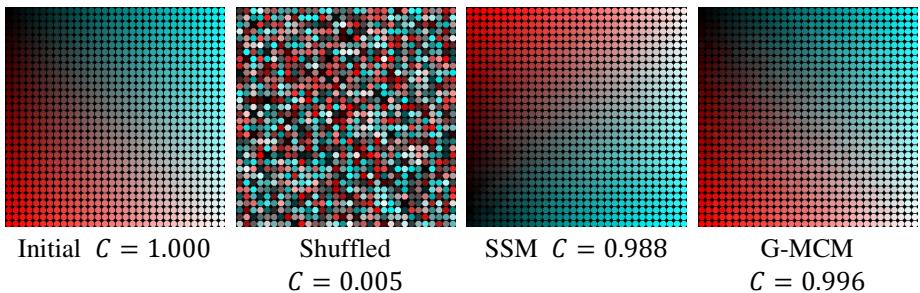


Fig. 5. Organizations for a set of color vectors whose color channels are set based on the grid coordinates (left). Using shuffled vectors (mid-left) as input, the results of SSM and G-MCM are shown on the right.

Since in both the G-MCM and SSM there is an element of randomness in the choice of swap positions, we study the effects of this randomness on layout quality here. Figure 6 shows that the G-MCM gives a correlation range of 0.944 to 0.996 (excluding a single 0.737 outlier) after 1000 runs, whereas the SSM yields 0.947 to 0.988 after 1000 runs. The median correlation of the G-MCM distribution (0.990) is higher than the comparable SSM (0.975).

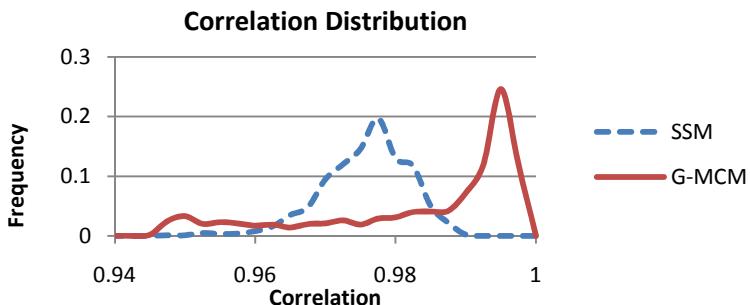


Fig. 6. In both plots the same 32×32 map of pseudo-color vector sets in Figure 5 are organized 1000 times and the frequency of the correlation scores is shown. Different runs start with different initial layouts and use different item swap orders. Left: G-MCM. Right: SSM.

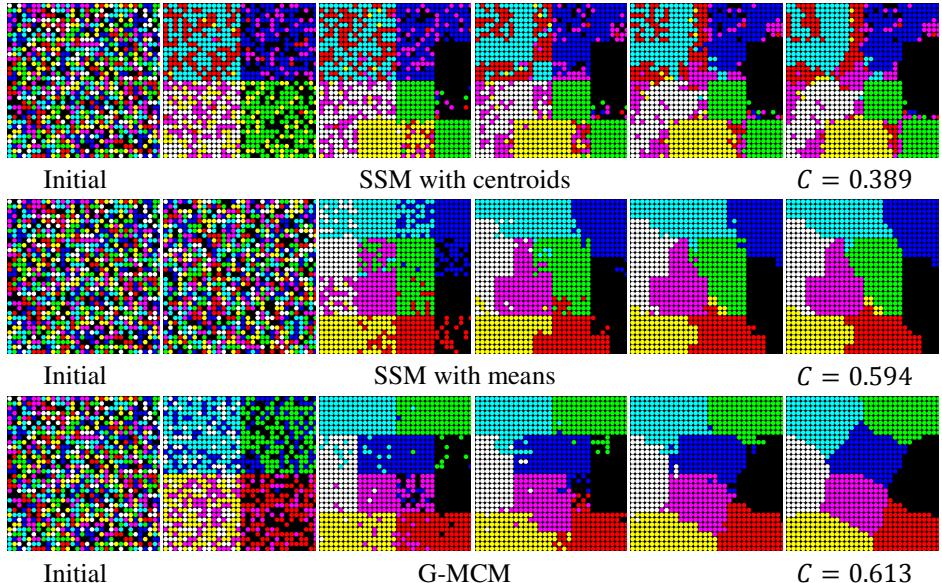


Fig. 7. SSM and G-MCM running on binary RGB colors swapping items as follows: Top row: by SSM based on centroids. Middle row: by SSM based on means. Bottom row: by G-MCM based on overall correlation.

Figure 7 further compares the performance of the G-MCM and SSM over a set of polarized colors. It shows that the deficiency of the SSM comes from its having to rely on block representatives, either means or centroids. When the SSM tries to organize these binary vectors, its representatives cannot capture the important facets of the data at the top level, resulting poor global layouts at coarse levels. The subsequent levels of the organizational hierarchy do not recover because the SSM focuses on local sorting. This undesirable layout is most noticeable when centroids, which are chosen items rather than means, are used. The G-MCM rectifies this issue by eliminating the need for representatives altogether and organizing things globally. Its arrangement is more correlated, more visually appealing, and more symmetrical. The locations of different colors' blocks are logical given the relationships between the primary and secondary colors.

5.2 Image Organization

In this section we show how the G-MCM can be used to organize a set of images. The images are acquired through Google image search using query “Washington”. Each image is tagged with a concept which can be compared with the concepts of other images using a dissimilarity matrix. Likewise, images also carry a visual content component from which dissimilarity can be derived. Details on describing images in this way can be found in [6, 7]. Since the G-MCM relies on item-item dissimilarity alone, it can directly arrange these images, whereas alternative approaches [17] require pre-processing of the concept matrix into vector form.

As shown in Figure 8, G-MCM can effectively place conceptually or visually similar images together. Existing user studies have shown that such a layout can facilitate users to find the desired images [5, 16]. The figure also demonstrates the use of fixed items. There is no change to the algorithm other than weighing the dissimilarity of fixed items more. This shows the flexibility of the G-MCM in dealing with “obstacles” in the layout.

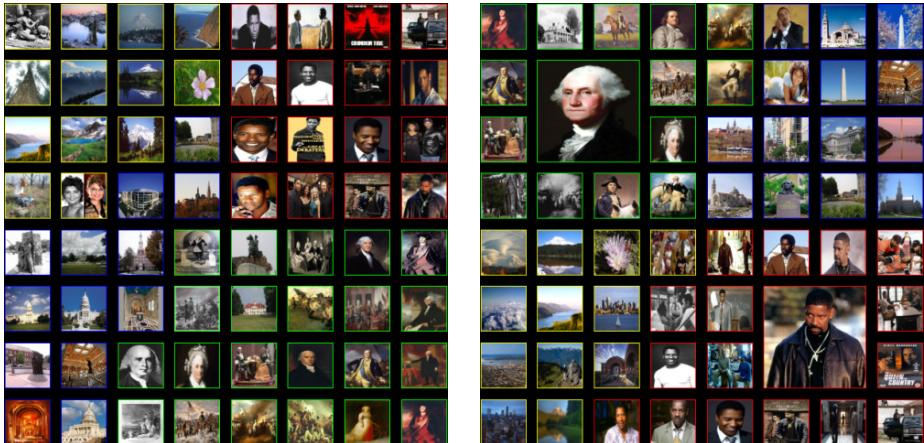


Fig. 8. The MCM layout of a collection of images expanded out of the query “Washington”. Color-coded borders relative to their category. “Denzel Washington” (red), “Washington State” (yellow), “Washington DC” (blue), and “George Washington” (green) are clearly visible. Within the regions, visually similar images also neighbor one another. Left is the standard arrangement while the right shows one with enlarged fixed items.

6 Conclusion

In this work, the problem of organizing data into structured layouts is formulated as maximizing the correlation between data similarity and their placements’ proximity. As demonstrated above, the presented G-MCM approach can consistently generate structured layouts of high global correlation. Although not as quick as the SSM, the correlations of its results are consistently higher than the SSM’s on the same data and it handles polarized datasets much better than SSM does. Furthermore, the G-MCM is more flexible in that the same basic algorithm can be adapted to handle either fixed or movable items of varying sizes. In terms of implementation, the G-MCM is significantly simpler than the SSM. In essence, it only consists of a single straightforward swapping stage that uses a block-level correlation score for guidance.

The main weakness of the algorithm is its high computational cost because of the correlation calculation requirement. Since it has been decomposed into a partial correlation map, it is possible to parallelize this calculation to speed up the process. As for future work, it would be worth investigating a heuristic rather than the current random neighborhood swap, which might lead to faster convergence, higher overall quality, and less deviation from the mean result.

References

- [1] Borg, I., Groenen, P.: Modern Multidimensional Scaling: theory and applications, 2nd edn. Springer, New York (2005)
- [2] Chen, T.T., Hsieh, L.C.: The Visualization of Relatedness. In: Proc. International Conference on Information Visualisation, pp. 415–420 (2008)
- [3] di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall (1999)
- [4] Hinton, G., Roweis, S.: Stochastic neighbor embedding. Advances in Neural Information Processing Systems 15(1), 833–840 (2002)
- [5] Hoque, E., Hoeber, O., Gong, M.: CIDER: Concept-based image diversification, exploration, and retrieval. In: IP&M (in press, 2013)
- [6] Hoque, E., Hoeber, O., Strong, G., Gong, M.: Combining conceptual query expansion and visual search results exploration for web image retrieval. JAIHC (in press, 2013)
- [7] Hoque, E., Strong, G., Hoeber, O., Gong, M.: Conceptual query expansion and visual search results exploration for Web image retrieval. In: Mugellini, E., Szczepaniak, P.S., Pettenati, M.C., Sokhn, M. (eds.) AWIC 2011. AISC, vol. 86, pp. 73–82. Springer, Heidelberg (2011)
- [8] Kaski, S., Peltonen, J.: Dimensionality Reduction for Data Visualization. IEEE Signal Processing Magazine 28(2), 100–104 (2011)
- [9] Kohonen, T.: Self-Organization Maps. Springer (1995)
- [10] Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Paatero, V., Saarela, A.: Self Organization of a Massive Document Collection. IEEE Transactions on Neural Networks 11(3), 574–585 (2000)
- [11] Morrison, A., Ross, G., Chalmers, M.: Fast multidimensional scaling through sampling, springs and interpolation. Information Visualization 2(1), 68–77 (2003)
- [12] Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
- [13] Sammon, J.W.: A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computing 18(5), 401–409 (1969)
- [14] Strong, G., Gong, M.: Data organization and visualization using self-sorting map. In: GI, St. John's, NL, Canada, pp. 199–206 (2011)
- [15] Strong, G., Gong, M.: Similarity-based image organization and browsing using multi-resolution self organizing map. IVC 29(11), 774–786 (2011)
- [16] Strong, G., Hoeber, O., Gong, M.: Visual image browsing and exploration (vibe): User evaluations of image search tasks. In: An, A., Lingras, P., Petty, S., Huang, R. (eds.) AMT 2010. LNCS, vol. 6335, pp. 424–435. Springer, Heidelberg (2010)
- [17] Strong, G., Hoque, E., Gong, M., Hoeber, O.: Organizing and browsing image search results based on conceptual and visual similarities. In: Bebis, G., et al. (eds.) ISVC 2010, Part II. LNCS, vol. 6454, pp. 481–490. Springer, Heidelberg (2010)
- [18] Tikhonova, A., Ma, K.-L.: A Scalable Parallel Force-Directed Graph Layout Algorithm. In: Proc. Eurographics Parallel Graphics and Visualization Symposium, pp. 25–32 (2008)
- [19] Van der Maaten, L., Postma, E., Van den Herik, H.: Dimensionality reduction: A comparative review. Technical Report TiCC TR 2009-005 (2009)
- [20] Venna, J., Peltonen, J., Nybo, K., Aidoo, H., Kaski, S.: Information retrieval perspective to nonlinear dimensionality reduction for data visualization. Journal of Machine Learning Research 11, 451–490 (2010)
- [21] Zhang, J., Chen, C., Li, J.: Visualizing the Intellectual Structure with Paper-Reference Matrices. IEEE TVCG 15(6), 1153–1160 (2009)

Mixing Geometrically Diverse Window Managers

Anthony Savidis^{1,2} and Andreas Maragudakis¹

¹ Institute of Computer Science, Foundation for Research and Technology – Hellas

² Department of Computer Science, University of Crete

Abstract. Compositing is currently the prevalent rendering paradigm for window managers. It applies off-screen drawing of managed windows with final image composition by the window manager itself. In this context, a compositing system is presented, enabling the concurrent presence of multiple window managers, being arbitrarily nested while facilitating switch managers on-the-fly. Two distinct managers are implemented, 2d desktop and custom 3d book, that can be freely combined into nested hierarchies. To allow such nesting two extensions are introduced. Firstly, the compositing process is turned to a rendering pipeline with window managers directly in-the-loop, with an imaging model combining diverse geometries. Secondly, to facilitate focus control in such geometric spaces, a cascaded pointing translation process is implemented, enabling geometric mapping of pointing events across nested window managers. The entire compositing system is implemented in a custom widget toolkit named sprint (in C++ with OpenGL and shaders) that is publicly available.

1 Introduction

Currently, the imaging model of most window managers reflects the compositing paradigm. The latter relies on the drawing of managed windows into off-screen buffers (normal rendering phase), with the window manager responsible to eventually compose the final picture from such buffers (compositing rendering phase). Also, compositing does not radically affect individual window rendering, since it is still performed as before, however, with output redirected in off-screen-buffers. Effectively, it brought two changes: (i) the final rendering stage; and (ii) initial pointing filtering from window manager space to local (planar) window space. While the amendments are overall simple, they are powerful in terms of the visual scenarios they support, and radical regarding the underlying implementation rework they require (i.e., GPU rendering).

We discuss two novel extensions along the standard compositing features. Firstly, we extend the imaging model to support nested window managers, thus enabling arbitrarily nested interactive spaces. In this context, nesting is possible on different window managers, while enabling users switch managers on-the-fly. Secondly, we extend the initial pointing filtering process towards a cascaded translation process in order to support focus control across nested window managers and their custom geometries.

We with the novel features of the compositing system. Then, we discuss the primary implementation aspects and patterns to accommodate them. Finally, we compare with related work, draw key conclusions and outline future steps.

2 Related Work

Window managers (Myers, 1988) with compositing implementations appeared originally under X windows more than a decade ago, following early pioneering work with first 3d window managers such as the Task Gallery (Robertson et al., 2000). Event today compositing managers like Compiz (Canonical, 2013), KWin (KDE, 2013) and Quartz (Apple, 2013) do not have interoperating variations and always work in a standalone fashion. Currently, for users to alternate across different spaces the notion of virtual desktops is offered (Ringel, 2011), but always with a similar window management style.

Additionally, they still operate outside the toolkit loop: after the toolkit in terms of display composition, and before the toolkit in terms of the initial input translation. While not related to compositing, the work on facades (Stürzlinger et al., 2006) revealed the need for compositional user-interfaces, something that inspired us towards dynamic window manager switching.

Display improvements for various scenarios have been proposed, like optimal space exploitation (Bell & Feiner, 2000), improved desktop rendering for importance-driven compositing (Waldner et al., 2011), optimal display usage for different monitor sizes (Hutchings et al., 2004; Hutchings & Stasko, 2004). In general, display improvements could be modeled in window managers as modular layout add-ons implementing different policies; however, we did not focus on this particular problem in the reported work. A constraint-based approach to model such policies is proposed in (Badros, 2001).

Various research efforts on compositing systems have been carried out, but little progress is made in extending the compositing pipeline with radical refinements such multiple and nested window managers. Metisse is a flexible compositing system (Chapuis et al., 2005) relying on FvwmCompositor which allows 3d transformations on window rendering, thus offering a framework for rendering windows beyond typical desktop topologies. However, it still cannot combine different window managers, while pick translation is typical single-stage preprocessing involving the window 3d transformation matrix.

3 Features

We implemented a compositing toolkit with two distinct window managers: (i) desktop window manager with a 2d geometric space and rectangular planar geometry; and (ii) custom book window manager with a 3d geometric space with a typical perspective view frustum. A snapshot with nested desktop and book managers, showing texturing and triangulation involved in compositing is provided in Figure 1, while cross nesting with book and desktop managers is provided under Figure 2.

Typical live taskbars with window miniatures are implemented, as in most compositing window managers, relying on the off-screen-buffers for the contents of managed windows. In our implementation they are currently included in desktops, while we are working on a geometric model suited to the book window manager.



Fig. 1. Mixed window managers with windows as textured (two triangles) rectangles

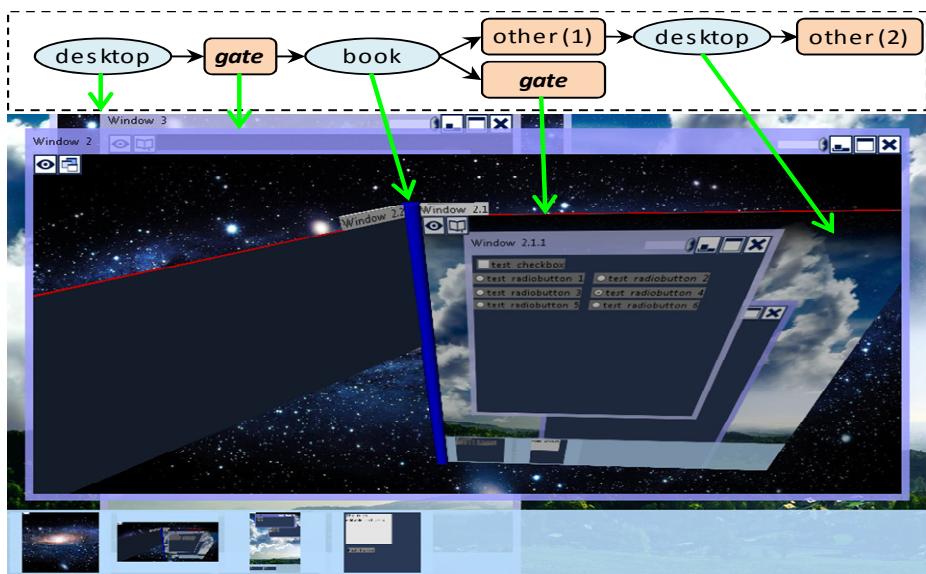


Fig. 2. Desktop encompassing a book encompassing a desktop

Switching from one window manager to another is facilitated on-the-fly, and, as we discuss later, in an automated implementation manner. Since switching is also an operation on window managers it is also included in their special local toolbar (see Figure 2, top-left window areas, right to eye icons, showing a desktop or book icons).

Because of nesting, window maximization can have a dual meaning. The common meaning, as observed in (standalone) desktop managers, is making windows occupy the full-screen space. However, in managers with non-desktop rendering plugins, like the cubic renderers of Compiz and KWin, the behavior is different with maximization adjusted to the rendering space were windows geometrically ‘live’ - in the previous cases the actual cube sides. Since the two interpretations are different, we decided to separate them as follows.

Manager maximization / restoration, offered to window managers only, not to windows, allowing them exploit full screen space. It enables give the impression that a window manager is standalone, (see Figure 3, right part). In our implementation this mode is interactively offered by a local toolbar at the top-left of the window-manager rendering area, and concerns to the ‘eye’ icon (see Figure 2 and Figure 3, toolbars at top-left window areas). Additionally, successive manager maximize requests are allowed on nested managers, with the restore operation always returning to the full-screen state of the hierarchically-closest parent manager.

Window maximization / restoration, an operation on managed windows, is optionally offered by individual window managers depending on their individual rendering models. In our case it is only supplied by desktops and is included in the standard window frame toolbar. Through this operation windows take the full space in area of their parent window manager (see Figure 3, ‘window maximize’). In our context, if a desktop is the root manager or is maximized, this operation maximize behaves exactly as in existing desktops and makes the window occupy full-screen space.



Fig. 3. Dual maximization operation applying on window managers (eye icon) and on managed windows (standard icon)

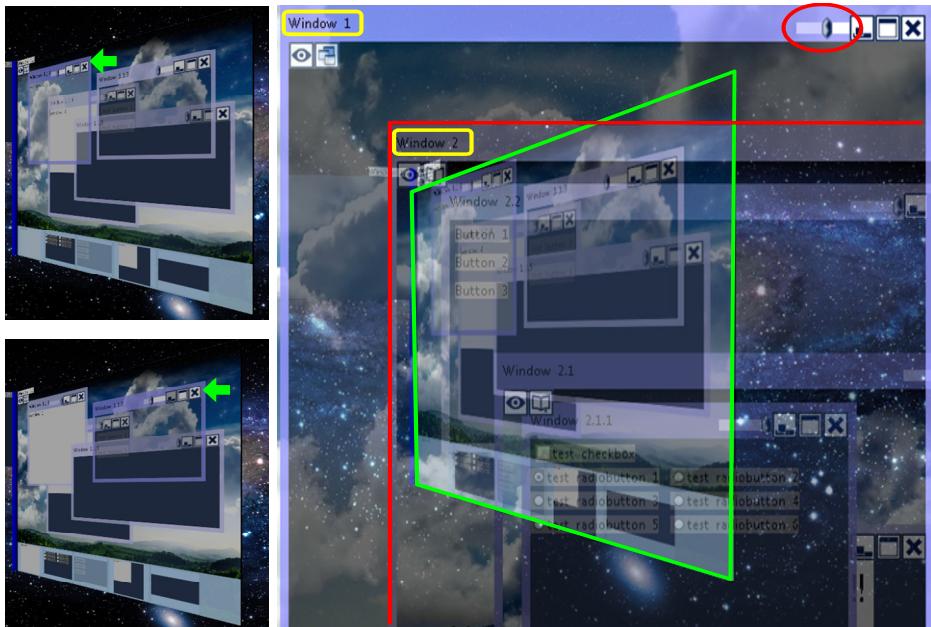


Fig. 4. Transparency control; Left: on windows of a desktop nested in a book; Right: book nested in ‘Window 1’ of desktop (‘Window 2’ is transparent and in-front of ‘Window 1’)

When supporting nested window managers the resulting interactive spaces not only become visually rich, but also sometimes visually complex. In particular, the desktop with its overlapped windows may cause nested window managers to be obscured. To allow users quickly inspect what is behind we introduced an extra item in the desktop toolbar for interactive transparency control (see Figure 3). Our implementation concerns appearance and does not allow obscured content to be interactive without the focus, as in (Robertson et al., 2000) were interaction with hidden content is enabled.

4 Implementation

In compositing toolkits, the classes regarding managed windows have always a local off-screen-buffer. Under OpenGL implementations the latter is typically a frame buffer object with a texture rendering target. In this context, to support dynamically nested window managers we introduced a managed window subclass named *gate*, with a dynamically associated window manager instance (optional, can be null). The root of the entire window hierarchy is always a *gate*, while *gates* may be freely nested within any container window. However, only instances of managed windows can have a *gate* instance as a parent.

The class design to accommodate our requirements is outlined under Figure 4 (many details omitted for clarity). Overall, window managers rely on decorators, one decorator added per managed window. The latter concerns the *WindowManager*

(WM) abstract class, its Decorate abstract method, and the *WindowDecorator* (WD) abstract class. This way, managed windows are added / removed to / from window managers as follows:

```
static WM::Add (ManagedWindow win)
    { decorators[win] = Decorate(win); }
static WM::Remove (ManagedWindow win)
    { decorators[win].Destroy(); decorators.erase(win); }
```

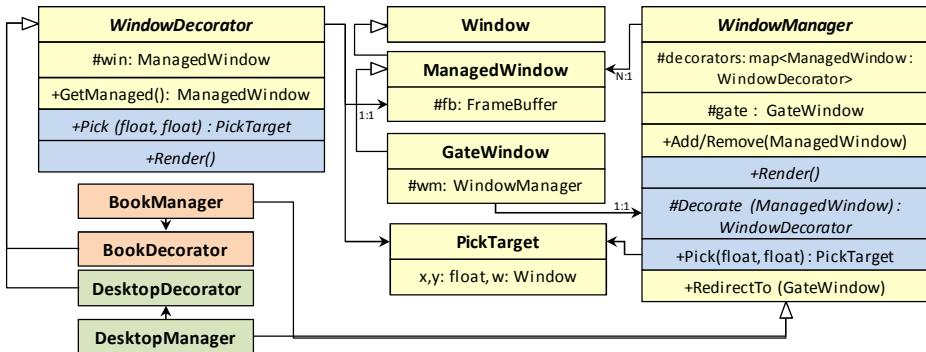


Fig. 5. Class schema to support dynamic and hybrid nesting of compositing window managers - key methods and members are only shown for clarity; it relies on dynamic decorators with augmented rendering and cascaded pick translation

As also shown under Figure 5, subclasses of window manager and decorator are defined in pairs, something reflected in the desktop and book window managers we implemented. One role of the decorator is to optionally introduce extra widgets necessary for interactive window control. The latter is a standard approach on desktops, originally introduced by window managers for X windows, with frames added as extra windows. We continue with the implementation details of the rendering pipeline, dynamic switching and cascaded input translation.

The rendering process is recursive as in all toolkits. When a gate window is asked to render itself it simply delegates the rendering work to its window manager instance:

```
GateWindow::Render() {
    Set rendering target to this.GetFrameBuffer()
    Draw any background image
    Get.WindowManager().Render()
}
```

We continue with the Render methods at the level of superclasses. There are also three non-abstract methods, two for rendering managed windows and decorators, and one to render the local toolbar (this is discussed later).

```

WM::RenderWindows() ←non-abstract method
{ foreach x in decorators do x.GetKey().Render() }
WM::RenderDecorators() ←non-abstract methods
{ foreach x in decorators do x.GetValue().Render() }
WM::Render()
{ RenderWindows() RenderDecorators() RenderLocalToolbar() }
WD::Render(){ /* default implementation is empty */ }

```

Then, in subclasses we may refine the abstract Render methods as required. We outline the implementation case regarding the book window manager:

```

BookWM::Render() {
    RenderWindows()
    Prepare all 3d polygons and refresh all modified window textures
    Set display camera and draw the entire scene
    RenderLocalToolbar()
}

```

As observed, in the book implementation the decorator subclass has no particular role in rendering. This is no general rule, but reflects our choice to gather all triangles for managed windows into one rendering batch (it is much faster than having separate batches per window). As mentioned, the entry points for dynamically attaching window managers are gate window instances. Interactively, the dynamic switching options are included in the local toolbar of window managers. In our implementation, the construction of this toolbar is automatic and can accommodate any future window manager subclasses that may be possibly implemented.

In other words, if an extra window manager is implemented, it will directly appear with a reserved entry in this toolbar. For this to work, a specific design pattern needs to be adopted, as outlined under Figure 6. In particular, a factory has to be implemented and a factory instance should be initially (at startup) registered to a factory directory with a suitable unique class identifier.

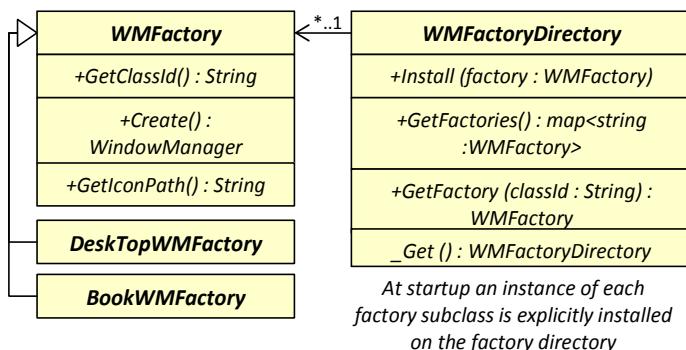


Fig. 6. Abstract superclass for window manager factories and the singleton registry with all factories indexed using their unique class ids

The local window-manager toolbar is placed on top of the rendering output that is drawn in the gate frame buffer, while it is interactively a standard toolbar with clickable behavior. It is prepared on-demand during rendering (i.e., when changed, it is internally cached) as follows:

```
WM::RenderLocalToolbar() {
    Get a copy of all factories from the factory directory
    Remove the entry for the wm class id of the caller, i.e. this
    Draw manager maximization entry and set its click handler
    foreach x in factories do {
        Draw the texture corresponding to x.GetIconPath()
        Set the click handler of the toolbar entry to invoke WM::Switch()
    }
}
```

Once a toolbar item is selected, the associated window manager is instantiated and set as the current in its respective gate instance, destroying the previous manager. This approach enables any number of window managers to be interactively activated and combined. The previous behavior is possible through the Switch method below:

```
static WM::Switch (GateWindow gate, string wmClassId) {
    gate.Get.WindowManager().Destroy()
    gate.Set.WindowManager(Create(wmClassId))
}

static WM WM::Create (string wmClassId) {
    factory = get factory entry from directory for wmClassId
    return factory.Create()
}
```

5 Cascaded Pick Translation

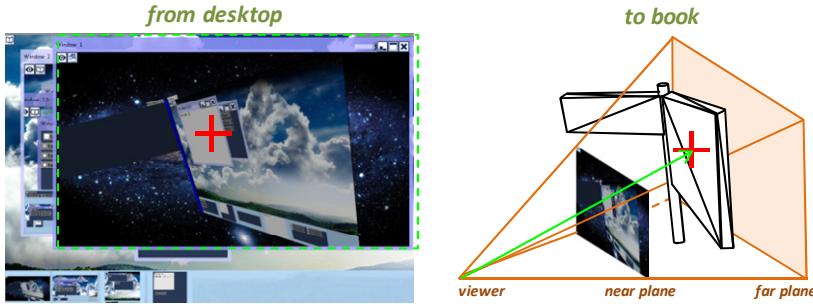
Picking through pointing is of key importance in toolkits and windows managers since it is used others for focus control. In our context, pointing is translated from the top window manager to the nested ones via a cascaded translation process (Figure 7).

More specifically, every window manager is able to translate from a planar position, whose coordinates are given relative to its gate, to a pair carrying the actual picked managed window and a relative position inside it. Then, the managed window invokes pick recursively to child windows. As with rendering, gates always delegate pick requests directly to their window manager:

```
PickTarget GateWindow::Pick (float x, float y)
{ return Get.WindowManager().Pick(x,y) }
```

The initial pick request is always in screen coordinates and is supplied to the root gate instance. The implementation of the default pick logic at the level of the window

manager superclass is to iteratively test managed windows. The latter is directly sufficient for the desktop manager subclass.



The *desktop* propagates hierarchically pick to *managed* windows which apply point-in-rectangle tests by default, except *gates* delegating window managers to pick

The *gate window* passes pick to its *book* which performs typical 3d ray-object intersection to find the *managed window* that will further process the pick

Fig. 7. Cascaded pick processing enabling to support precise pointing interaction and focus window shifting across the diverse geometries of nested window managers

```
PickTarget WM::Pick (float x, float y) {
    foreach x in decorators do {
        PickTarget target = x.GetKey().Pick()
        if target.GetWindow() ≠ null then
            return target
    }
    return null      ← fallback means no pick target was found
}
```

However, when it comes to the book manager subclass, the pick logic is more comprehensive and is provided below. Initially, the planar pick event is translated to the geometry of the book by producing a ray on the perspective view frustum using the current camera. Then, translation proceeds by recursively invoking the *Pick* method on the managed window corresponding to hit book page. The latter is done after translating the ray intersection point to the local planar space of the window.

```
PickTarget BookWM::Pick (float x, float y) {
    Translate pick coordinates x,y to an 3d ray
    Determine closest intersecting book page and its hit point
    if a book page is intersected then {
        Let win be the managed window for this book page
        Translate hit point to page plane coordinates x',y'
        return win.Pick(x',y')
    }
}
```

6 Conclusions and Future Work

We have implemented a low-fidelity experimental toolkit named *sprint*, supporting compositing rendering and hosting of arbitrarily nested window managers. We enabled users switch window managers on-the-fly to any registered alternative window manager. To experiment with different geometries we implemented a desktop and book window managers and tested various dynamic nesting configurations between them. In this context, we mainly focused on the technical amendments and implementation for nested compositing window managers, rather than on the human factors and interaction quality inherent in hybrid and nested workspaces. In summary, we propose the involvement of window managers into the loop with a cascaded rendering and pick-translation pipeline.

Our implementation also indicated that a few extensions on the toolkit side are required, the most prominent being the need for a special window class, called *gate* in our work, to offer hosting of embedded window managers. The tests also showed that switching is extremely fast, since it only involves the initial creation of the window manager custom geometry. The contained windows and window managers are by default cached due to compositing rendering.

The entire implementation of *sprint* and the two window managers is available with anonymous checkout from <https://139.91.186.186/svn/sprint>

References

1. Myers, B.: A taxonomy of window manager user interfaces. *IEEE Computer Graphics and Applications* 8(5), 65–84 (1988)
2. Robertson, G., Van Dantzich, M., Robbins, D., Czerwinski, M., Hinckley, K., Risdan, K., Thiel, D., Gorokhovsky, V.: The Task Gallery: a 3D window manager. In: Proc. CHI 2000, pp. 494–501. ACM (2000)
3. Bell, B.A., Feiner, S.K.: Dynamic space management for user interfaces. In: Proc. UIST 2000, pp. 239–248. ACM (2000)
4. Badros, G.J., Nichols, J., Borning, A.: Scwm: An Extensible Constraint-Enabled Window Manager. In: USENIX Annual Technical Conference, FREENIX Track 2001, pp. 225–234 (2001)
5. Ishak, E.W., Feiner, S.K.: Interacting with hidden content using content-aware free-space transparency. In: Proc. UIST 2004, pp. 189–192. ACM (2004)
6. Waldner, M., Steinberger, M., Grasset, R., Schmalstieg, D.: Importance-driven compositing window management. In: Proc. CHI 2011, pp. 959–968. ACM (2011)
7. Stürzlinger, W., Chapuis, O., Phillips, D., Roussel, N.: User interface façades: towards fully adaptable user interfaces. In: Proc. UIST 2006, pp. 309–318. ACM (2006)
8. Chapuis, O., Roussel, N.: Metisse is not a 3D desktop! In: Proc. UIST 2005, pp. 13–22. ACM (2005)
9. Hutchings, D.R., Smith, G., Meyers, B., Czerwinski, M., Robertson, G.: Display space usage and window management operation comparisons between single monitor and multiple monitor users. In: Proc. AVI 2004, pp. 32–39. ACM (2004)
10. Hutchings, D.R., Stasko, J.: Shrinking window operations for expanding display space. In: Proc. AVI 2004, pp. 350–353. ACM (2004)

11. Ringel, M.: When one isn't enough: an analysis of virtual desktop usage strategies and their implications for design. In: CHI Extended Abstracts, pp. 762–763. ACM (2003)
12. Apple. Quartz Compositor,
http://apple.wikia.com/wiki/Quartz_Compositor (accessed, April 2013)
13. Canonical LTD. Compiz, <https://launchpad.net/compiz> (accessed April 2013)
14. KDE. KWin, <http://techbase.kde.org/Projects/KWin>
(accessed April 2013)

Classifier Comparison for Repeating Motion Based Video Classification

Kahraman Ayyildiz and Stefan Conrad

Department of Databases and Information Systems,
Institute of Computer Science, Heinrich Heine University Duesseldorf, Germany
`kahraman.ayyildiz@uni-duesseldorf.de, conrad@cs.uni-duesseldorf.de`

Abstract. In this paper we introduce a repeating motion based video classification system. Videos from certain topical areas like sports, home improvement, or mechanical motion often show specific repeating movements. Main and side frequencies of these repetitions can be considered as motion features. We receive these features by the Fourier transform of spatio-temporal motion trajectories and use them during classification phase. Our experiments focus on various classifiers in order to find the most accurate classifier for motion frequency related features.

1 Introduction

In computer vision research on video surveillance, video retrieval, or object tracking is encouraged by obvious demand. Face tracking for example is useful for videoconferencing as the camera follows the speaker's head during a speech. Furthermore major corporations and online video portals maintain video databases. Thus motion and video analysis is relevant for industry, technique and practical life.

This research work illustrates how frequency features from cyclic motion can be utilized for video classification. Our previous work [1] supplies the basic idea of this approach. Now the feature extraction phase works more efficiently by using frequency averages of the whole spectrum instead of one to three frequency peaks. Moreover in the experimental stage we focus on the accuracy and runtime of different classifiers in order to determine the best classifier for our system. As even frequency spectra of videos with same class labels can vary clearly, we need a classifier that is able to handle such differences.

In figure 1 we depict the different stages of our approach. The flow diagram starts with video data input containing repeating movements like hammering, planing, or filing for instance (home improvement). At first regions of movement are detected in every clip framewise. Region detection takes place by measuring the color difference of pixels in two frames following each other. Based on regions we can compute image moments and accordingly centroids. The chronological order of image moments or centroids is considered as a 1D-function and represents the motion in a video sequence. The transform of a 1D-function reveals its frequency domain. By partitioning the frequency axis into intervals of same

length, average amplitudes for each interval are computed. Henceforth we refer to these averages as *AAFIs* (Average Amplitudes of Frequency Intervals). AAFIs constitute the final feature vectors for each clip with respect to its motion. These feature vectors represent the input data for classifiers, which then compute the nearest class for a video scene.

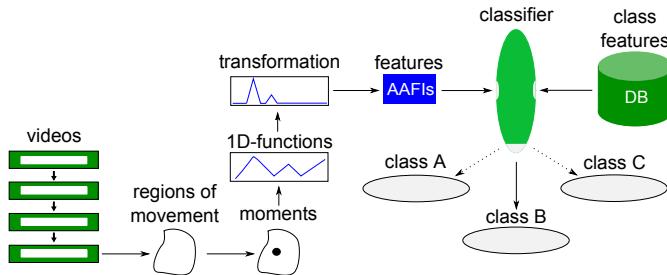


Fig. 1. Flow diagram of whole classification process

2 Image Moments and 1D-Functions

Once the motion in a video is detected image moments and accordingly 1D-functions can be determined. So now we explain regions of motion and define image moments formally.

2.1 Regions of Motion

In figure 2 a person painting a wall is depicted. By analyzing two consecutive frames of this activity we detect regions with motion. Color differences between the first and the second frame are measured for each pixel. The color difference of a pixel exceeding a predefined threshold combined with a minimum number of neighbor pixels with a color difference beyond the same threshold defines a pixel to be a part of a movement. Thus a region of motion is represented by the conflation of pixels with motion.

In figure 2 the pixel differences of these two frames show up regions with movement, which again are plotted as a binary image. We can see that the most active areas are the paint roller, the hand, the forearm and the upper arm. Hence the centroid of regions with motion follows exactly the right forearm. As a result the painting action sets a specific motion track as it evolves in time.

2.2 Image Moments

An image moment is defined as the weighted average of pixel intensities of a picture. It can describe the area, the bias, or the centroid of segmented image parts. The two main image moment types are raw moments and central moments.

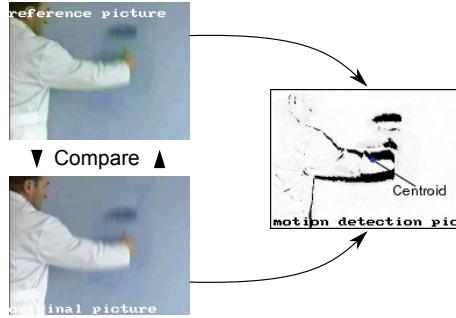


Fig. 2. Regions with pixel activity and centroid

Raw moments are sensitive to translation, whereas central moments are translation invariant. Next equation defines a raw moment M_{ij} for a two dimensional binary image $b(x, y)$ and $i, j \in \mathbb{N}$ [2]:

$$M_{ij} = \sum_x \sum_y x^i \cdot y^j \cdot b(x, y) \quad (1)$$

The order of M_{ij} is always $(i + j)$. M_{00} is the area of segmented parts. Consequently $(\bar{x}, \bar{y}) = (M_{10}/M_{00}, M_{01}/M_{00})$ determines the centroid of segmented parts. In addition central moment computations involve centroid coordinates [2].

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i \cdot (y - \bar{y})^j \cdot b(x, y) \quad (2)$$

Here μ_{20} and μ_{02} represent the variances of pixels with regard to x and y coordinates, respectively. We calculate image moments for each frame.

2.3 Deriving 1D-Functions

Video frames have a chronological order. Hence a series of moment values is also depending on time t . Now we define a 1D-function $f(t)$ as a series of these moment values by considering only one dimension. For centroid coordinates $(\bar{x}_t, \bar{y}_t) = (M_{10_t}/M_{00_t}, M_{01_t}/M_{00_t})$ we decompose function $f_c(t) = (\bar{x}_t, \bar{y}_t)$:

$$f_{c_x}(t) = \bar{x}_t \wedge f_{c_y}(t) = \bar{y}_t \quad (3)$$

Experiments in section 5 use only $f_{c_x}(t)$ and $f_{c_y}(t)$ instead of $f_c(t)$, because the 1D-function transforms result in more accurate frequency domains than 2D-function transforms. For any 1D-function $f(t)$ the direction of a moment at time t is defined by equation 4.

$$f_d(t) = \begin{cases} +1, & \text{if } f(t) - f(t-1) > 0 \\ 0, & \text{if } f(t) - f(t-1) = 0 \\ -1, & \text{if } f(t) - f(t-1) < 0 \end{cases} \quad (4)$$

3 AAFIs as Feature Vectors

As already mentioned each 1D-function can be transformed to its frequency spectrum. After partitioning this spectrum into intervals of same length, we compute an average amplitude for each interval. So the whole frequency spectrum is captured by AAFIs as feature vector.

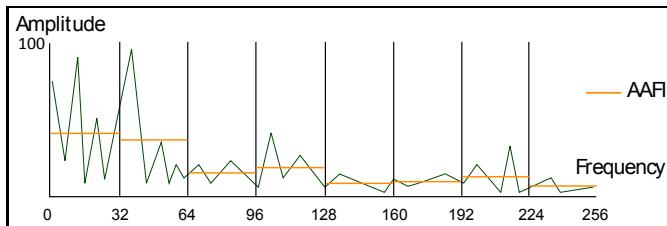


Fig. 3. Average amplitudes of frequency intervals (AAFIs)

In figure 3 we illustrate the AAFI approach by partitioning a frequency domain with a length of $m = 256$ units to $n = 8$ intervals. Since we use the fast Fourier transform variables m and n have to be a power of 2, where $m \geq n$. Further the horizontal, orange lines mark the average amplitude of each interval. Hence with regard to figure 3 one 1D-function leads to 8 average amplitudes and one 8-dimensional feature vector, respectively. Due to the fact that videos produce two 1D-functions, each video is described by two 8-dimensional feature vectors in this example. Thus a partitioning of the frequency spectrum into n intervals results in a $(2 \cdot n)$ -dimensional feature vector for each video.

4 Defining Classifiers

As the main focus of this work is on the evaluation of classifiers, we now introduce several classifiers utilized for the classification process.

First of all we define $C = \{C_1, \dots, C_m\}$ as our set of classes. Each class $C_i \in C$ contains a set of objects, so we define $C_i = \{o_{i_1}, \dots, o_{i_{n_i}}\}$, $C_i \neq \{\}$ and $C_i \cap C_j = \{\}$ for $i \neq j$. The total of all objects in classes is $A = C_1 \cup \dots \cup C_m$. Now a set of objects $B = \{o_1, \dots, o_l\} \neq \{\}$ without class labels are considered for classification.

4.1 K-Nearest Neighbors

The k-Nearest Neighbors classifier (KNN) [3] searches k neighbors with a minimal distance to a given object $o_b \in B$ among all objects $o_i \in A$. Object o_b is then assigned to the class with maximal amount of neighbors. The metric and the parameter k used play a central role for the rate of correct classifications. A large k can reduce noise, but it can also involve a wrong neighbor class.

$$class_{knn}(o_b, C) = \arg \max_{C_i \in \mathbf{C}} |C_i \cap n_k(o_b)| \quad (5)$$

Equation 5 shows the selection method of our KNN classifier. Let $n_k(o_b)$ the set of k nearest neighbors for o_b . Then $class_{knn}(o_b, C)$ leads to the class with the most elements in $n_k(o_b)$. If several classes have the same maximal number of elements in $n_k(o_b)$, one class is chosen by chance. The KNN classifier's time complexity is of the order $\mathcal{O}(e \cdot d)$, since e objects out of all classes need Euclidian distance calculations and each distance calculation amounts to d iterations - one iteration for each dimension.

4.2 Average Linkage Classifier

Hierarchical clustering algorithms often use the *average linkage* distance for distance calculations between clusters [4]. Now we apply this distance for our classifier by assigning an object to the class with the minimal average linkage distance. Therefore we call this classifier Average Linkage Classifier (ALC). More precise the ALC computes all Euclidian distances between an object $o_b \in B$ and objects $o_i \in C_i$. Dividing the sum of these distances by the class size $|C_i|$ yields the average linkage distance δ_i between object o_b and class C_i .

$$\delta_i(o_b) = \frac{1}{|C_i|} \sum_{o_i \in C_i} \|o_b - o_i\| \quad (6)$$

$$class_{alc}(o_b, C) = \arg \min_{C_i \in \mathbf{C}} (\delta_i(o_b)) \quad (7)$$

The ALC assigns object o_b to the class with the minimal distance δ_i among all classes $C_i \in C$. If several classes have the same minimal distance to o_b , one class is chosen at random. ALC's main advantage is that it works without any parameter input. Object classes are given and average linkage distances are normalized by the class size. Its time complexity is of the order $\mathcal{O}(m \cdot e \cdot d)$. The implementation is a nested loop, where Euclidian distance calculations need d iterations. This calculations take place for e objects of one class and m classes have to be compared.

4.3 Nearest Centroid Classifier

In machine learning a nearest centroid classifier (NCC) assigns an object $o_b \in B$ to the class with the closest centroid μ_i [5]. Just as ALC NCC's main advantage is its independence of any input parameter.

$$\mu_i = \frac{1}{|C_i|} \sum_{o_i \in C_i} o_i \quad (8)$$

$$class_{ncc}(o_b, C) = \arg \min_{C_i \in \mathbf{C}} \|\mu_i - o_b\| \quad (9)$$

Equation 9 predicts class C_i with the nearest centroid to o_b . Centroids for m classes are calculated. Moreover each centroid calculation depends on e objects in each class and on the dimensionality d of the objects. In addition each centroid-object distance computation depends again on the dimensionality. So the time complexity order is $\mathcal{O}(m \cdot e \cdot d) + \mathcal{O}(m \cdot d) = \mathcal{O}(m \cdot e \cdot d)$.

4.4 Radius Based Classifier

Now we explain our *Radius Based Classifier* (RBC) [1]. Let object $o_b \in B$ and class $C_i \in C$, then radius ε determines the ε -neighborhood $N_\varepsilon(o_b, C_i)$. This ε -neighborhood encloses all objects of class C_i inside the predefined radius around o_b . RBC is sensitive to the compactness of a class. So the choice of parameter ε is decisive. The distance between objects is measured by Euclidian distance.

$$N_\varepsilon(o_b, C_i) = \{o_s | o_s \in C_i \wedge \|o_b - o_s\| < \varepsilon\} \quad (10)$$

Based upon $N_\varepsilon(o_b, C_i)$ we define the distance between an object o_b and a class C_i .

$$\nu_i(o_b) = 1 - \frac{|N_\varepsilon(o_b, C_i)|}{|C_i|} \quad (11)$$

$$class_{rbc}(o_b, C) = \arg \min_{C_i \in C} (\nu_i(o_b)) \quad (12)$$

Equation 12 gives the class with the minimal distance to o_b among all classes. For $|class_{rbc}(o_b, C)| = 1$ the RBC designates o_b to the next class C_i . For $|class_{rbc}(o_b, C)| > 1$ one class is chosen at random. Now here again the time complexity order is $\mathcal{O}(m \cdot e \cdot d)$.

4.5 Support Vector Machine

A Support Vector Machine (SVM) is a classifier, that separates the vector space for a given set of labeled training objects into class spaces [6]. Afterwards a new object can easily be assigned to a class. So class borders have to be determined. These borders are called *hyperplanes*. A valid separation by hyperplanes is only possible, if the object set is linearly separable. If not kernel functions are applied in order to describe the hyperplanes in higher dimensional vector spaces. We receive SVM by minimizing error function

$$\frac{1}{2} w^T w + C \sum_{i=1}^m \xi_i \quad (13)$$

with restrictions:

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, \dots, m \quad (14)$$

Variable w is the normal vector to the hyperplane and index i labels the m training classes. Parameter ξ_i handles inseparable input data. It measures the

degree of misclassification of the training data x_i . The objective function 13 is then increased by a function which penalizes non-zero ξ_i , and the optimization becomes a tradeoff between a large margin and a small error penalty. Additionally capacity constant C regulates that tradeoff. Consider that $y_i \in \pm 1$ are class labels for training objects x_i . Bias b is a further constant and kernel function ϕ transforms input data to the feature space. Now for multi-class classification we use *one-against-one* approach: If m is the number of classes, we generate $m(m - 1)/2$ models. Each model involves only two classes of training data and assigns an instance to one of these two classes. The class with the maximal amount of assignments determines the final classification. SVM's training phase and finding the best kernel or best parameters take much time. In worst case training complexity order is $\mathcal{O}(N_S^3)$ with N_S as the number of support vectors. Test phase takes $\mathcal{O}(M \cdot N_S \cdot m^2)$ operations, where M is the number of kernel operations. Using RBF kernel complexity order is $\mathcal{O}(d \cdot N_S \cdot m^2)$.

4.6 Artificial Neural Network

An artificial neural network (ANN) is composed of artificial neurons with connections or edges among each other [7]. Commonly its topology consists of three layers: input, hidden and output layers. Hidden layers are not visible from the outside, but they improve the abstraction of the ANN model. ANN is an adaptive system and can change its structure during training phase. This means building and deleting connections, adjusting edge weights, or adjusting a neuron's activation function. Each neuron has a network function $f(x)$ which is a composition of functions $g_i(x)$, which again can be a composition of functions. By adding arrows between these functions or neurons, it is possible to visualize a network structure. An established composition type is the *nonlinear weighted sum*:

$$f(x) = K \left(\sum_i w_i g_i(x) \right) \quad (15)$$

Here K represents the activation function and w_i is the weight for each function. Finding optimal parameters and training can be an effort. Let q the number of layers, r_q the number of neurons in each layer and S the cost for multiplying the weight with the input and adding it to the sum. Then the complexity order for executing ANN is $\mathcal{O}(q \cdot r_q^2 \cdot S)$.

5 Experiments

This section focuses on accuracy and runtime performance of our system with respect to introduced classifiers.

5.1 Motion Transformation

Figure 4 depicts an example 1D-function, which stems from a person's motion while using a wrench. Particularly the figure plots x-axis coordinates of centroids

and captures the main motion. It is obvious that the 1D-function corresponds to the left-right and right-left movements. Transforming this 1D-function by fast Fourier transform (FFT) results in a frequency domain with peaks at 13 and 27.

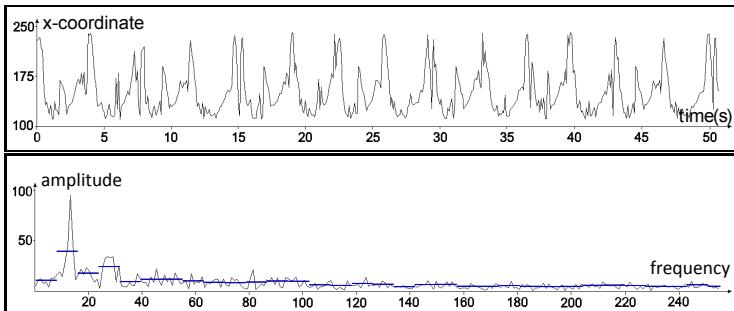


Fig. 4. Transform of a wrench handling 1D-function

5.2 Classifier Accuracies

In total we assign 200 own and 102 external video scenes [8] to one out of ten home improvement classes. For own video data we use m-fold cross validation, whereas external videos are assigned directly to own video classes, because cross validation was not possible due to external classes with just few clips. Our ANN is a multilayer perceptron neural network, which utilizes supervised learning technique *backpropagation*. Neuron weights w_i are randomized and by adjusting the number of neurons in layers, the number of iterations and the learning rate, we optimize the neural network. SVM's supervised learning algorithm is based on C-SVC (C-Support Vector Classification) and the radial basis function as kernel. Here classes are not weighted. Table 1 shows resulting accuracies. *Overall accuracies* are not the average of *class accuracies*, but the ratio of correct classifications to the total number of classifications. For own videos direction information and for external videos location information of image moments is applied, since external videos contain more irregular movements. Moreover table 1 depicts that the classifier with the maximum accuracy 0.92 is KNN classifier. Classifiers NCC, RBC and SVM also achieve high accuracies, whereas ANN classifier yields a minimum accuracy at 0.78. Considering single class classifications home improvement activity "Putty Knife" achieves extraordinary high accuracies. Almost all classifiers can assign all test videos correctly, because its movements and by association its features are clearly different from other classes. Furthermore the entire system works so accurate for own videos, that there no single class with low accuracies for each classifier. Now for external video classification ANN is the strongest approach. Its accuracy marks the maximum at 0.45. SVM falls down to 0.08, thus it is not reliable for external videos.

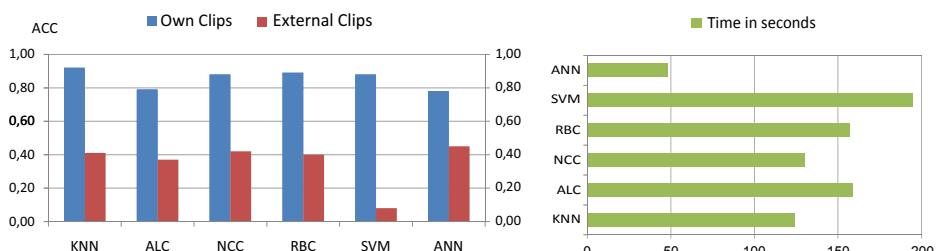
Table 1. Classifier accuracies for own and external videos

Class ACC	Own Videos						External Videos					
	KNN	ALC	NCC	RBC	SVM	ANN	KNN	ALC	NCC	RBC	SVM	ANN
File	0.85	0.40	0.80	0.85	0.80	0.85	0.00	0.00	0.00	0.00	0.00	0.00
Hammer	0.85	0.80	0.75	0.95	0.85	0.75	0.00	0.00	0.00	0.00	0.00	0.00
Paint Roller	1.00	0.80	1.00	1.00	1.00	0.45	0.79	0.64	0.86	0.69	0.00	0.57
Paste Brush	0.95	0.90	0.95	1.00	0.90	0.85	0.25	0.00	0.00	0.25	0.00	0.25
Plane	0.80	0.65	0.55	0.50	0.80	0.85	0.09	0.00	0.09	0.00	0.00	0.00
Putty Knife	1.00	1.00	1.00	1.00	1.00	0.85	0.65	0.57	0.57	0.57	0.00	0.83
Sandpaper	0.90	0.40	0.90	0.94	0.65	0.80	0.36	0.09	0.55	0.18	0.00	0.64
Saw	0.90	0.95	0.90	0.75	0.90	0.60	0.18	0.45	0.27	0.33	0.00	0.45
Screwdriver	1.00	0.95	0.95	0.90	0.90	0.75	0.86	0.71	0.86	0.57	0.00	0.57
Wrench	0.95	1.00	0.95	1.00	1.00	1.00	0.25	0.63	0.25	0.50	1.00	0.25
Overall ACC	0.92	0.79	0.88	0.89	0.88	0.78	0.41	0.37	0.42	0.40	0.08	0.45

Accuracy values for KNN, ALC, NCC and RBC are located around 0.40. Videos of the classes "File" and "Hammer" are completely misclassified, because of feature similarities to videos of the classes "Paint Roller" and "Paste Brush". Figure 5 shows that own videos are assigned twice as accurate as external videos. This is due to irregular motions and recordings in external clips. For clear and smooth video motions the KNN classifier works most accurate. Otherwise ANN classifier works best. A neural network can handle irregularities better than space model based classifiers. SVM is not reliable for irregular motion data, because widespread feature vectors with high dimensions make proper space partitioning difficult. As a result often one space partition dominates the whole vector space and videos with varying feature vectors mostly are assigned to this dominating partition. On the whole KNN classifier seems to be the most effective classifier.

5.3 Runtime Analysis

Figure 6 shows the runtime for each classifier, when the amount of test videos is set to 1000. The classification process relies on a database with 20 videos for each of the ten classes. Test series with SVM and ANN classifier are based on previously trained test model and neural network, respectively.

**Fig. 5.** Overall accuracies for own and external clips**Fig. 6.** System's runtime with different classifiers

Otherwise runtime would exceed plotted time range. The bar chart shows that RBC and ALC classifier both need about 160 seconds, since both algorithms compute distances between test object and all objects of each class. NCC and KNN classifier operate much faster. For NCC only distance calculations to class centroids are necessary. KNN classifier is fast, because for each assignment there is only one object set to analyze. Nevertheless ANN classifier outperforms all other classifiers by assigning 1000 videos in 48 seconds. As we use *Neuroph Framework* the weighted sum input function is optimized and the neural network is a matrix based implementation [9]. Hence decisions by the ANN classifier are highly performant. On the other side SVM classifier shows up low performance. In this case we utilize *LIBSVM Framework* [10]. The one-against-one approach explained before needs much more operations to find the next class. Following line chart 7 depicts system's runtime, when the number of classified videos is enlarged. Since each video classification adds a fixed time value to the runtime, its growth is linear for each classifier. At first glance it becomes apparent that ANN classifier has the slightest runtime growth and handles large scale data sets most effectively. Moreover the mean runtime for each video assignment is 0.048 seconds.

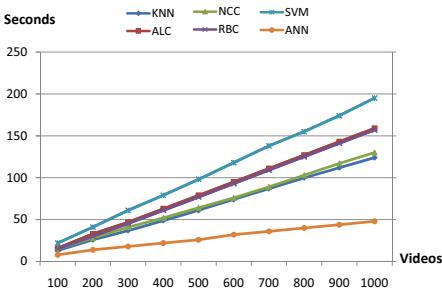


Fig. 7. System's runtime with increasing amount of classified videos

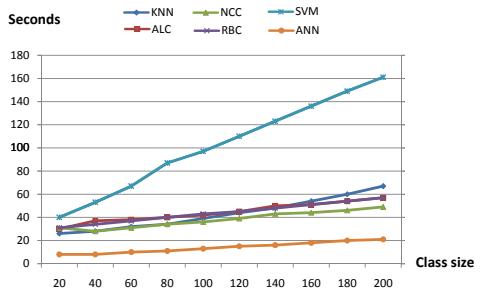


Fig. 8. System's runtime with increasing referenced class sizes

In figure 8 runtime for 200 classified clips with up to 200 referenced videos per class in database is presented. SVM classifier shows up the strongest runtime increase. The bigger the SVM model becomes the more resources are needed. Furthermore involving more support vectors multiplies the one-against-one operations. Increasing the number of classes would increase runtime even more. The class size rise has a minor effect on NCC. As we increase referenced class size with factor 10 its runtime increases with factor 1.58. KNN has a clear upward movement, since we use a sorting algorithm for KNN in order to find the k-nearest neighbors. Also ANN classifier rises strongly, because the more video features are involved the more difficult it becomes to calculate the decision via neural network. Test results mark up that the number of classified videos has a stronger impact on runtime than the size of referenced classes.

6 Related Work

Videos reveal a huge amount of information. Hence video annotation and classification is realized in various manners. Main techniques base on key-frames [11], texts in frames [12], audio signals [13] and motions. In [14] repeating motion of human body parts is analyzed by tracking Moving Light Displays (MLD). The frequency peaks of Fourier transformed MLD curves are considered as features of cyclic motion. The classification of different motion types results in high accuracies from 0.84 to 0.96. Unfortunately authors miss to give rise about the size of tested dataset. Cheng et al. analyze sports videos by using a neural network based classifier in [15]. Series of horizontal and vertical pixel motion vectors are transformed by a modified FFT and result in two main frequencies for each clip. Here authors state accuracies up to 1.0 for five analyzed sports activities, but the average class size is three and therefore not convincing.

Some research in computer vision considers direct comparison of classifiers based on motion features. Sobral et al. provide a system for highway traffic video classification [16]. Capturing crowd density and crowd speed of vehicles the system assigns traffic videos to one of three congestion classes. Test series with classifiers KNN, Naive Bayes, SVM, and ANN yield accuracies of 0.93, 0.93, 0.93 and 0.95, respectively. The number of classified highway traffic videos is 254. Research paper [17] is about an outdoor personal navigation system, which works computer vision aided. Feature selection takes place by analyzing video images and position information provided by GPS. After comparing SVM, ANN and KNN classifiers authors state that SVM works best for their purposes. Accuracies range from 0.50 to 1.00 for different activities like driving, walking, running, using stairs or an elevator. Another work concerns hand signal classification by muscle contraction EMGs [18]. Here again SVM performs best, KNN shows high accuracies and DT (decision tree) shows the lowest performance. Authors Rocamora and Herrera use features from transformed audio signals to detect the singing voice part in music audio files [19]. In this case classifiers SVM, ANN, KNN and DT achieve accuracies of 0.85, 0.83, 0.77 and 0.73, respectively.

7 Conclusion

In this paper we have shown a video classification method based on the frequency of repeating movements. Frequency spectra are computed by transforming spatio-temporal image moment trajectories (1D-functions). We explained how partitioned frequency spectra can be utilized for feature extraction and video classification.

The experimental stage of our work focused on different classifiers in order to find the most efficient one. In literature SVM and ANN classifiers show up the best results. Our approach performs best with KNN and ANN classifiers. Although KNN classifier is extremely simple, for videos with clear recording conditions it works best. With KNN classifier we could assign 184 videos out of 200 videos properly just relying on repeating motion data. ANN classifier

achieves higher accuracies than other classifiers when recording conditions are not homogenous. A neural network is capable of handling irregularities concerning the frequency domain features better than spatial distance or spatial partitioning based approaches. Adding weights and activation functions to neurons leads to correct classifications even if AAIs of some frequency areas vary. Hence a video classification system relying on frequency features will work more accurately with ANN for most real world databases.

Real time action recognition and evaluation with publicly available datasets like KTH and HOHA remain open issues for our repeating motion based approach.

References

1. Ayyildiz, K., Conrad, S.: Video classification by main frequencies of repeating movements. In: Int. Workshop on Image Analysis for Multimedia Interact. Serv. (2011)
2. Wong, W., Siu, W., Lam, K.: Generation of moment invariants and their uses for character recognition. Pattern Recognition Letters, 115–123 (1995)
3. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 21–27 (1967)
4. Voorhees, E.M.: Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Inf. Processing and Management, 465–476 (1986)
5. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Inf. Retrieval (2008)
6. Hill, T., Lewicki, P.: Statistics: Methods and Applications (2006)
7. Foody, G.M.: Relating the land-cover composition of mixed pixels to ann classification output. Photogrammetric Engineering and Remote Sensing, 491–499 (1996)
8. LLC, Y.: Youtube: Broadcast yourself, youtube.com (2013)
9. Neuroph: Java neural network framework, neuroph.sourceforge.net (2013)
10. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines, csie.ntu.edu.tw/~cjlin/libsvm (2013)
11. Pei, S., Chen, F.: Semantic scenes detection and classification in sports videos. In: Conference on Computer Vision, Graphics and Image Processing, pp. 210–217 (2003)
12. Lienhart, R.: Indexing and retrieval of digital video sequences based on automatic text recognition. In: 4th ACM International Conference on Multimedia, pp. 419–420 (1996)
13. Patel, N., Sethi, I.: Audio characterization for video indexing. In: SPIE on Storage and Retrieval for Still Image and Video Databases, pp. 373–384 (1996)
14. Meng, Q., Li, B., Holstein, H.: Recognition of human periodic movements from unstructured information using a motion-based frequency domain approach. Image and Vision Computing, 795–809 (2006)
15. Cheng, F., Christmas, W., Kittler, J.: Periodic human motion description for sports video databases. In: Int. Conference on Pattern Recognition, pp. 870–873 (2004)
16. Sobral, A., Oliveira, L., Schnitman, L., Souza, F.D.: Highway traffic congestion classification using holistic properties. In: 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications (2013)
17. Saeedi, S., Moussa, A., El Sheimy, N.: Vision-aided context-aware framework for personal navigation services. International Society for Photogrammetry and Remote Sensing, 231–236 (2012)
18. Glette, K., Gruber, T., Kaufmann, P., Torresen, J., Sick, B., Platzner, M.: Comparing evolvable hardware to conventional classifiers for electromyographic prosthetic hand control. In: NASA/ESA Conf. on Adaptive Hardware and Systems, pp. 32–39 (2008)
19. Rocamora, M., Herrera, P.: Comparing audio descriptors for singing voice detection in music audio files. In: 11th Brazilian Symposium on Computer Music (2007)

Implementation of Source Engine for Virtual Tours in Manufacturing Factories

Petr Horejsi and Jiri Polcar

University of West Bohemia, Univerzitni 8, 306 14 Plzen

Abstract. When we discuss virtual reality as a medium, there are two main genres: virtual tour and virtual training. This paper deals with the use of virtual reality for interactive virtual tours in manufacturing factories. We developed a brand new software **package/library DIGITOV** which serves as an aid for constructing new virtual interactive manufacturing factory models using Source Engine. This graphic engine is known for its use in the Half-Life 2 computer game [1]. One of the general advantages of adapting this engine is the public availability of the powerful development software tools included in the Source SDK. While adapting the engine to a new environment it is necessary to prepare a library consisting of many new 3D models, textures, sounds, choreographed scenes etc. The package includes for instance: machines, products, parts of a manufacturing line, specific sounds etc.

1 Introduction

We can observe a visible trend of penetration of virtual reality into everyday praxis. Many industrial corporations have their own concept of a digital factory: all the aspects of manufacturing are digitally verified on digital mock-ups before physical manufacturing. This approach brings significant cost savings. Virtual reality is used mainly in the validation phase; for example, verifying the design and functionality of a digital prototype, the driving properties of a physically non-existing car, ergonomics of a workplace etc. We will focus on the validation of manufacturing companies' layout and employee training. Using the further described virtual tours it is possible to visualize the whole factory including administration space, validate the perspective for working activities, have a virtual discussion with guides, etc. Generally it is possible to perform many more interactive actions in a customizable environment. All the virtual tours can be practically made using stereoscopic projection in a CAVE (Computer Aided Virtual Environment) using a haptic controller (see Fig. 1). There is a possibility of using supported DirectX stereoscopy with the Razor Hydra controller.

There are several software tools for modeling digital factories with the possibility of adding models from a 3D library to user composition. Two of today's major solutions are the Dassault Systeme DELMIA and Siemens Tecnomatix. These can be used to perform a large number of analyses [2]. From these two packs the full virtual tour

in CAVE can be provided by Teamcenter VisMockUp software but with limited interaction. The other suitable tool is VisTable which natively supports a virtual tour but without the possibility of interacting with the virtual world.

All the universal software tools for virtual environment development are relatively expensive. This is why we have been searching for a way to use a widely spread environment for enterprise and factory design. From the practical side: we were also looking for a modular possibility in order to develop a universal learning tool for a virtual world design course for industrial engineers. After researching more possibilities we have chosen to modify Source Engine. Its single non-commercial license costs about €10.

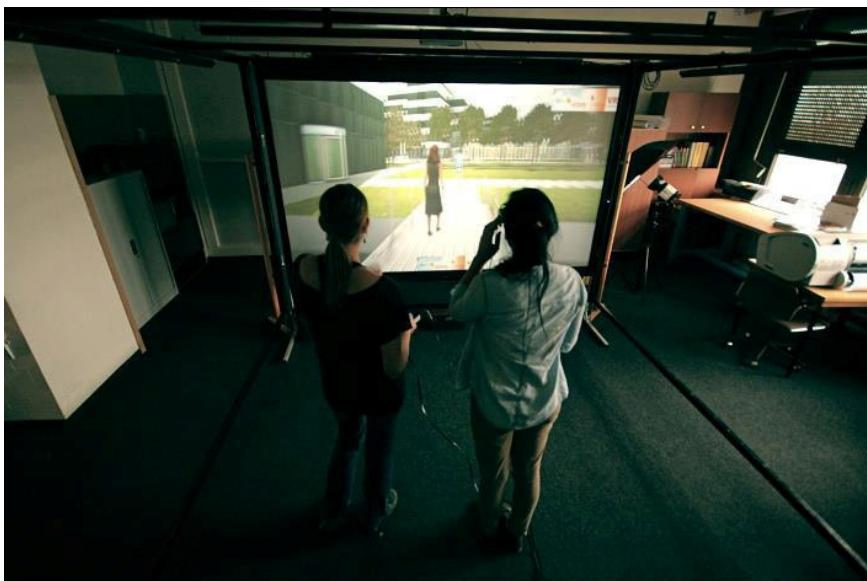


Fig. 1. Target single-screen CAVE with IS-900 tracking device

2 Serious Games

Our system (see below) uses the principles of serious gaming. This term relates to specific interactive simulations where the main reason is not only to entertain, but also to educate the user or engage him/her in solving particular problems. A certain amount of fun while playing is required, though. One of the first applications of serious gaming came from the army. There is for example a free game by America's Army for training soldiers and advertising [3]. Another technically simpler game is for example Food Force. It was developed by the United Nations and informs about famine in the world [4].

Serious games, like any other computer game, can be developed from scratch using conventional programming environments or with the aid of so called ‘sandbox’

programs. These programs can be used to accelerate the development of virtual environments with the possibility of implementing interactive elements. 2D or 3D objects can usually be imported from standard formats, the interactive elements are implemented by block programming or scripting. Some examples of stand-alone sandbox solutions are Thinking Worlds, WinterMute Engine, Unity3D or VirTools (with native supports of multicluster visualization in CAVE). Another possibility is the use of game editors made by game developers simultaneously with the engine (which is the case described in this paper).

Adaptation of standard game engines for serious games is very frequent. For example, CryEngine has been successfully adapted for training Dubai's S.W.A.T police commandos [5], the US Army [6] and even as a surgery simulator [7]. Source Engine was used for example for a restaurant hygiene simulator [8] and other serious games.

3 Source Engine and Source SDK

Another implementation described here uses **Source Engine**, which is a modular game engine for PC, Linux, Mac OS, PS3 and Xbox. Source Engine was released in 2004 for the computer game Counter-Strike: Source, and then for Half-Life 2 one year later. Source is a high-quality graphics core with simulated physical systems support using the Havok engine. Source engine is based on the DirectX architecture with the possibility of High Dynamic Range. More features such as multicore rendering support, Hardware Facial Animation, "Soft" particles, etc. have recently been added [9].

The virtual environment is built by a **map** (or level). The virtual world can be comprised of more interconnected maps. All particular environments are limited to a user-defined enclosed volume which is composed of **brushes**. Brushes are basic 3D objects that represent walls, floors, ceilings, cliffs, terrain, etc. The world details: e.g. furniture, humans, trees or sometimes even whole buildings (with esthetic functionality only) are represented by **3D models** (static, dynamic and physical props). Brushes are objects of lower detail than 3D models and are modeled and edited directly in Source SDK enclosed tools. On the other hand, 3D models are modeled using complex 3D editing tools like 3DS MAX, Blender or XSI SoftImage and then imported to a Source based map. All items and NPCs (Non-Player Characters) are represented by detailed 3D models. Functional parts of maps are implemented by so-called **entities** (doors, lifts, switches, light control, any mathematical or physical logic, etc.).

Every virtual environment developed in Source Engine is physically stored in a specific directory structure which includes for instance these components (directories):

- Maps – all maps are stored here (BSP - Binary Space Partitioning format)
- Materials – contains all textures (VTF – Valve texture File) including a material description file in text format (VMF – Valve Material File)
- Models – all the detailed 3D models (MDL format). Only special textures from Materials directory can be used for models.

- Scenes – scripted scenes data (choreography) created by the FacePoser tool which is included in the Source SDK
- Sound – sounds in WAV format

This structure can be compiled into a single .gfc archive. There is also a possibility to inherit data from other modifications (adapted Source Engine packs) or Source powered games. In other words, it is possible to use all previously designed models, textures, NPCs, etc.

Components of the virtual environments in Source Engine are created and edited using the Source SDK which is a powerful development package tool which can be downloaded via the Steam distribution system by the owner of a valid Source engine game (like Half Life 2). This environment includes a possibility of new modification (so called ‘mods’) development. Within or without this modification a new map can be developed. There is also a tool for new map creation called **Hammer Editor**. Software tool FacePoser can be used for creating choreographed scenes including face mimics and lips to sound synchronization. There are a lot of other tools included, usually file format converters, such as image to VTF (Valve Texture File) and many more. The source code of the engine is included, so the possibilities are virtually unlimited.

4 Reference Model

First a reference model was created in order to validate the possibility of developing such complex models in the given engine.



Fig. 2. Manufacturing layout of a reference model (in Siemens NX)

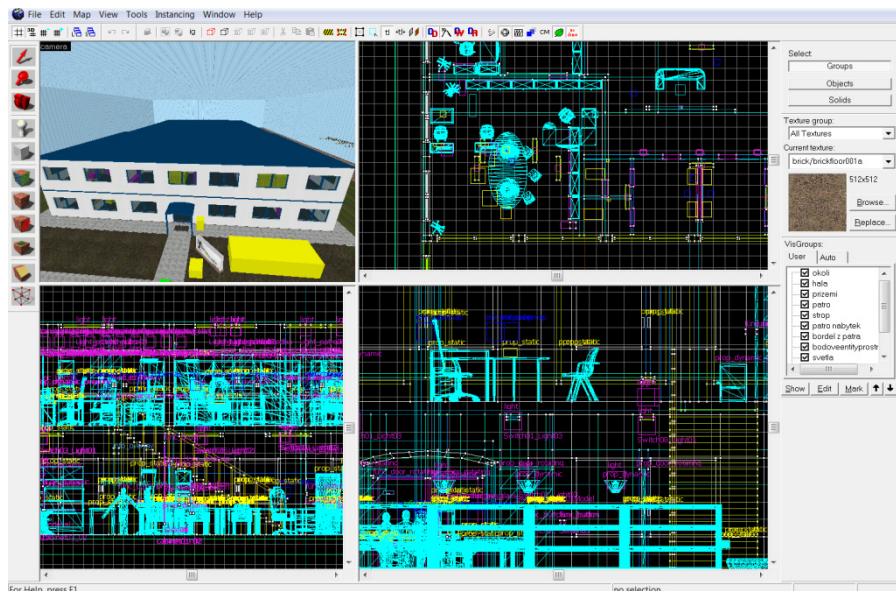


Fig. 3. Example of the development of the first reference model

An extensive virtual re-engineering project was used as an ideological base. In this project each step of the referential product (RC car) production planning was digitally composed starting with a raw design and ending with assembly line simulation optimization. All the production data including 2D and 3D layout from the Technomatix software package were available (such as 3D layout in Fig. 2). The assembly line was re-created using the Source SDK tools. The project output data also helped with construction of other non-manufacturing facilities (offices, canteen, toilets, exterior etc.). Simple interactive features were implemented, such as switchable lights, doors, ladders, etc. More complex interactivity was also added: e.g. the conveyor belts control, interaction with the operators and choreographed scenes with the virtual actors (mentors) in general.

During the development (see Fig. 3) it was revealed that the number and type of models and textures included in the original core was not sufficient. 3D models of workshop and office furniture, machines, IT equipment, etc., were missing. Based on the original gaming purpose, it is clear that the original includes models and textures which are supposed to immerse the user in a thrilling atmosphere, for example the furniture is sometimes half-broken and the walls are dirty, ergo they were not suitable to be used for our case.

5 DIGITOV Package

The evaluation of the development, implementation and validation process of the referential model raised the need for new content. So as to be able to effectively create new virtual models of manufacturing factories, it was necessary to produce a

brand new library comprising 3D models, textures, interactive features and so called in Source SDK **prefabs** (logically grouped brushes, entities and 3D models). Some of these were created while working on the reference model. In order to maximize the modularity of the “building set”, a lot more of such components were needed to be added. The DIGITOV modification for the computer game Half-Life 2: Episode Two started to take form. This is not a standard game modification, like a new story, but a collection of components for developers of virtual enterprise environments. The package includes an automatic installer developed in Delphi. There is just one requirement for the DIGITOV package to work: to own a user license for the Half-Life 2: Episode Two computer game, preferably in the Orange Box edition.

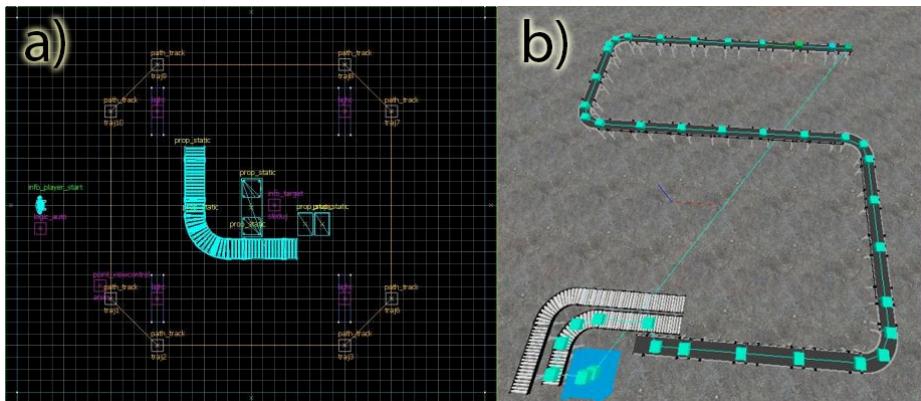


Fig. 4. a) conveyor belt curve prefab, b) conveyor belt constructed from prefabs in Hammer Editor

The DIGITOV currently contains approximately 150 new 3D models, more than 50 textures for models, more than 50 textures for maps and more than 25 static and dynamic prefabs. Static prefabs represent usually concrete or brick blocks with/without windows/doors, from which a raw factory layout could be composed. Dynamic prefabs are sets of prepared interactive aggregations. One example of dynamic prefabs from the DIGITOV package could be particular active parts of a conveyor belt (start/end, straight, curve, T-shape part, etc.), where besides the model and track path are also defined points where the product model is transformed (assembled), where the belt can be connected to another part of the belt, where the operator performs interactions etc. (see Fig. 4)

The package also contains various predefined choreographed scenes with virtual employees like various types of greetings, mentoring, presentations and more. These can be assembled into arbitrary sequences in the level editor. The original virtual actors' clothes were “virtually cleaned and ironed” (see Fig. 5-a). All these choreographic scenes were prepared using the FacePoser tool, where the particular face mimics and speech is synchronized and composed into replicas in a timeline (see Fig. 5-b). For English language Microsoft Speech SDK can be used for automatic mimic and voice synchronization. For other languages time consuming manual synchronization has to be done.

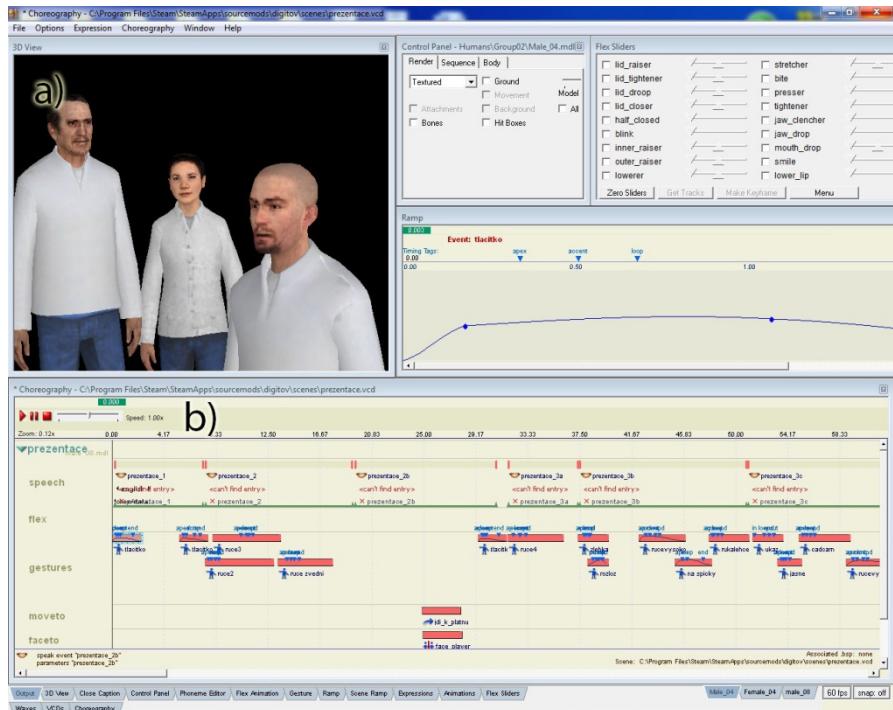


Fig. 5. a) Re-skinned NPC employees in Face Poser, b) Action editor

The DIGITOV package is installed in the modifications folder. While launching the Source SDK tool, the DIGITOV mod can be simply selected from the menu instead of the default Source Engine 2009, then all the additional options are available in the Valve Hammer Editor.

6 An Example of DIGITOV Package Implementation

Let us show one of the virtual enterprise models which was designed with the aid of the DIGITOV package from a user point of view.

The first person camera perspective view of the model is shown just after the virtual environment has been loaded. The control system help is displayed immediately after loading the model. The user has a complete mouse and keyboard controlled freedom of movement. The right mouse key serves for displaying labels or hints - mainly for interactive elements (like offices, persons, products, etc.). This feature is very useful for instance for key elements of conveyor belt inspection.

While taking a tour in one of the models the user follows a storyboard (see Fig. 6):

- External view of the manufacturing enterprise. It is possible to examine the parking lot, climb up on to the roof. Let us take a look inside,

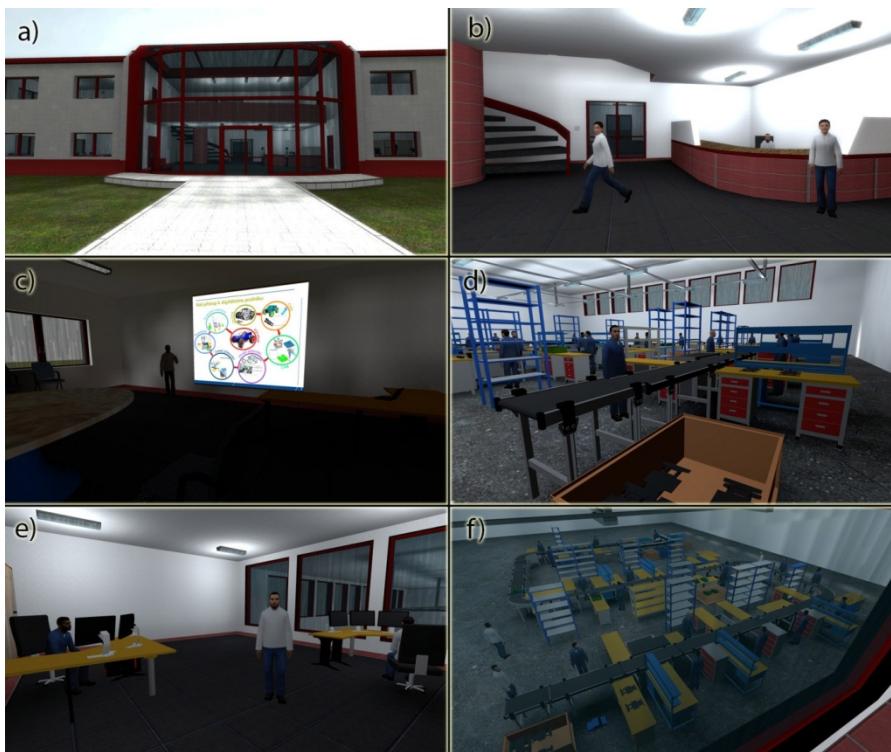


Fig. 6. Storyboard

- b) where the user is welcomed. A virtual lady will take us to the presentation room,
- c) where the lecturer will show us some basics of the production in this factory (this presentation is more than 10 minutes long, but can be skipped)
- d) then the user is brought to the manufacturing section. Each particular section of the manufacturing process can be inspected.
- e) The user can also be familiarized with the functions of every single administrative office in the administrative part of building.
- f) The manufacturing section can be observed from a bird's-eye view.

7 Validation, Conclusion and Results

The developed package is an alternative to other enterprise design software. It is a convenient tool not only for students, but can be used also for commercial use (a valid commercial license needs to be purchased by an enterprise). It can be used for instance to show the factory to new employees, who are recruited in another part of the world or for common visualization of not yet existing enterprise buildings for many reasons.

A manual and an education e-book (with more than 200 pages) were written alongside development. 14 authors spent 3 years developing DIGITOV and the e-book.

The ready-to-use package has been used for making 15 complex virtual enterprise models; two of them were models of real factories, validated by its staff; two more models of residential houses and a partial model of a university building. More reusable assets were designed and added to the package during the development of the models.

In comparison with large commercial tools for factory design like Delmia, Tecnomatix or Autodesk Factory CAD the proposed package does not offer the option to perform analyses of material flow, capacity tasks, ergonomics task, etc. but the DIGITOV package has advantages in better visualization, solution of training tasks and user interactivity implementation with low-cost. Industrial engineers who were approached reacted very positively to the presentation power and high fidelity of simulation. DIGITOV offers an assets library which can compete with factory design software packages.

Right now with the DIGITOV package and a little experience a fully interactive factory design can be produced in a few hours. For example, in comparison with DELMIA, the same factory layout was developed in half the time.

The DIGITOV package successfully supports the lessons of the “Digital enterprise and virtual reality” subject (so far for two years). The students are shown the basics of developing virtual worlds. The package is also being used for lessons at the “Summer school of virtual reality” for high school students.

The environments can be viewed stereoscopically on a 3D monitor or in CAVE. The virtual world can be controlled using a keyboard and a mouse or with a special Razor Hydra controller 9, where a sensor monitors the position of the special controllers in both hands via a magnetic field. The right location has to be found while using this controller in CAVE because of possible interference issues.

Although the DIGITOV package represents a powerful alternative tool for making mock-ups of virtual enterprises, there are still some ideas for improvement, such as:

- Dynamic animations in some models are missing.
- The map must be run from the console using the “map” command – a GUI launcher would be suitable.
- Interactive height selection of the player in order to verify the working positions perspective for different operator heights.

One of these recently corrected imperfections was the low count of textures. Volunteers searched the internet for usable free textures. Then a texture service pack was released containing more than 1000. Another recent validated potency is the possibility of adding more remote host users into a map and then to arrange off-distance briefings in a virtual factory.

The main point of the subsequent research is to develop a convertor, which will automatically convert a factory model from VisTable software into DIGITOV ready maps.

Acknowledgements. This paper was prepared with support of the Internal Science Foundation of the University of West Bohemia SGS–2012-063.

References

1. Source Engine games (June 2013),
<http://www.giantbomb.com/source-engine/3015-751/games/>
2. Kurkin, O., Januska, M.: Product Life Cycle in Digital factory. In: Knowledge Management and Innovation: A Business Competitive Edge Perspective, 15th International-Business-Information-Management-Association, Cairo, Egypt, November 6-7, vol. 1-3, pp. 1881–1886 (2010) ISBN: 978-0-9821489-4-5
3. America's Army (June 2013), <http://www.americasarmy.com/>
4. Food Force 2 (June 2013), <https://code.google.com/p/foodforce/>
5. Dubai police training (June 2013),
<http://tbreak.com/megamers/16428/features/dubai-police-uses-cry-engine-3-for-swat-training/>
6. Intelligence Electronic Warfare Tactical Proficiency Trainer (June 2013),
<http://www.peostri.army.mil/PRODUCTS/IEWTPT>
7. Virtual Surgery Far Cry Demo (June 2013),
<http://www.youtube.com/watch?v=QF0yyfhchvg>
8. Mac Namee, B., et al.: Serious Gordon: using serious games to teach food safety in the kitchen. In: 9th International Conference on Computer Games: AI, Animation, Mobile, Educational and Serious Games CGAMES 2006 (2006)
9. Wikipedia: Source(game engine) (June 2013),
[http://en.wikipedia.org/wiki/Source_\(game_engine\)](http://en.wikipedia.org/wiki/Source_(game_engine))
10. Razer Hydra (June 2013),
<http://www.razerzone.com/gaming-controllers/razer-hydra>
11. Kurkin, O., Simon, M.: Optimization of Layout Using Discrete Event Simulation. In: Business Transformation Through Innovation and Knowledge Management: An Academic Perspective, 14th International-Business-Information-Management-Association Conference, Istanbul, Turkey, June 23-24, vol. 1-4 (2010) ISBN: 978-0-9821489-3-8

Evaluating 3D Vision for Command and Control Applications

Britton Wolfe, Beomjin Kim, Benjamin Aeschliman, and Robert Sedlmeyer

Dept. of Computer Science
Indiana University-Purdue University Fort Wayne (IPFW)
Fort Wayne, IN 46805, USA
`{wolfeb,kimb,aescbd01,sedlmeye}@ipfw.edu`

Abstract. 3D stereoscopic vision is used in many applications, but the level of benefit to the user differs depending on the particular application. We studied its benefits for command and control applications such as battlefield visualization or disaster response. We conducted experiments where the subjects completed some simple military planning exercises both with and without 3D vision. 3D users had lower error when judging line of sight between two points. Furthermore, survey results show that subjects preferred 3D. We also compared two ways of rendering symbols in the environment. Billboard symbols were more efficient than draping the symbol on the terrain.

1 Introduction

Command and control (C2) software displays information to commanders about the units under their control, the locations of those units, and any other information that may be relevant for the current situation. This information is typically displayed on 2-dimensional screens or monitors. However, the locations of military units, points of interest, and even the terrain of the battlefield¹ itself is naturally 3-dimensional data. One method of displaying 3D information on a 2D device is to project the 3D information into the 2D plane. This is called a 2.5D display. However, the projection into 2D loses depth information, which can make it difficult for the user to estimate depth from the 2.5D display (e.g., determining the slopes of the mountains in Figure 1).

In contrast, a 3D vision system renders the data so that a user actually perceives objects in three dimensions by using stereoscopic vision. Stereoscopic displays are a natural fit for geospatial information such as battlefield locations. Our work studies the strengths and weaknesses of stereoscopic vision specifically for completing military planning tasks with C2 systems.

2 Background and Motivation

3D stereoscopic devices have become increasingly prevalent in the consumer market and benefit users in various ways. They have been utilized in a variety of

¹ While our study examines military C2, the results can also inform other C2 uses like disaster response or border control.

areas such as simulation, training, entertainment, education, physical sciences, geography, and medicine [1–4].

Human beings perceive depth by obtaining a different view of the world from each eye and reconstructing a 3D view from them [5]. Stereoscopic 3D devices display two different viewpoints in a similar manner. By seeing two different images from two different perspectives, the user is able to perceive the depth of a rendered environment. Stereoscopic 3D images provide a binocular view point that gives additional depth information not available with monocular viewpoints.

3D stereoscopic environments provide depth perception advantages, but there are known limitations that prevent users from accurately perceiving depth. The lack of natural depth cues and differences between the user's actual convergence in reality and the viewer's convergence on the screen can make depth perception difficult [6]. Researchers have studied methods for enhancing depth perception and reducing visual fatigue when using 3D vision technology [7]. Previous experimental studies have produced mixed results, showing a general trend of underestimation in depth perception in 3D environments [8, 1, 9, 10]. However, another study showed 3D stereoscopic visualization to be helpful for egocentric distance estimation during robot teleoperation [11]. Another study showed that users' depth perception accuracy varied not only when using different graphics effects, but also when using different zooming levels of virtual spaces [12].

When compared with a standard 2D interface, 3D interfaces projected into 2D are sometimes beneficial [13] and are sometimes detrimental [14] to the usability of the system [15]. The existing mixed results on 3D and 2.5D interfaces underscore the importance of systematically studying the strengths and weaknesses of 3D and 2.5D displays in the specific case of C2 systems.

3 Software Description

We built a basic C2 application that visualizes real geographic regions (Figure 1), allowing us to compare 2D, 2.5D, and 3D displays [16]. We call our software C2 in a Virtual Environment (C2VE). We developed C2VE using OpenGL.

3.1 Rendering the Environment

C2VE combines a satellite image with an associated height map to display the terrain of a battlefield (Figure 1). One of the primary features of C2VE is the sense of depth that the user experiences when using the 3D system. Of course, this 3D experience cannot be adequately conveyed in a paper. For example, from the 2.5D images in Figure 1, it is difficult to discern the slope of the mountains. Moving the camera in a 2.5D environment can help with perceiving the slopes, but excessive movement could be disorienting. Instead, 3D lets users perceive the slope (and other depth information) even without moving the camera.

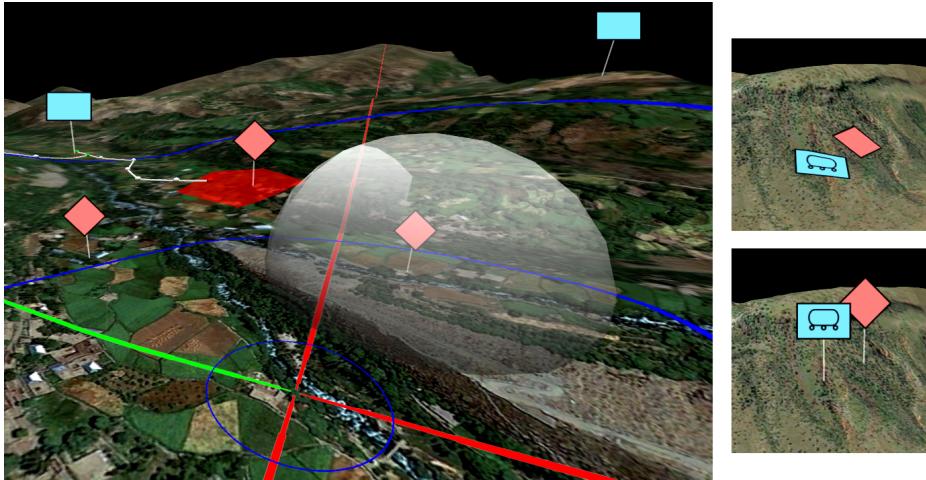


Fig. 1. Screen shot of C2VE software (left). Draped (top right) and billboard (bottom right) symbol modes.

We acquired several aerial orthophoto image files from the U.S. Geological Survey's Seamless Data Warehouse [17]. The image has the resolution of about 7000 by 7000 pixels and represents 7 by 7 km of land. This is sufficient to give the user a good sense of buildings, rivers, roads, farms, trees, and mountains. The elevation data was retrieved from the USGS database in the Digital Terrain Elevation Data (DTED) format. The DTED data was then processed in MICRODEM mapping software to generate the height map image file that incorporates elevation data of the target region [18]. Compared to the image, the height map has lower resolution (about 512 by 512 pixels), but it is still sufficient to capture important terrain features.

We used the open source 3D graphics software, Blender, to generate terrain mesh models from the height map and orthographic image [19]. The aerial image is texture mapped to the surface of the terrain mesh model through a bilinear interpolation. Terrain models are transformed to a pictorial image by passing through rendering pipelines in the order of vertex, geometry, and fragment shaders. The generated terrain view displays a pictorial scene with proper illumination that shows different levels of height changes. When the scene is rendered using the 3D vision technology, it presents a virtual space mimicking a real landscape view.

3.2 Symbol Drawing

The C2VE system allows the user to interactively place and manipulate military symbols. It uses a subset of standard military symbols [20] which are stored

as textures. Because the symbols are 2D images, C2VE supports two different methods of displaying them in the 3D environment: as a billboard or as a draped mesh (Figure 1).

The billboard objects float a little ways above the terrain, dynamically changing orientation as the camera moves to ensure that the symbol is always facing the user. For the draped objects, C2VE generates a mesh that is located just above the terrain with the texture of the symbol applied to the visible (i.e., skyward-facing) side. This mesh object is fabricated based on the shape of the terrain under the symbol, so the symbol closely adheres to the terrain.

3.3 Interacting with the System

The user interacts with the environment through the keyboard, mouse, and a 3D Space Navigator (a 6-degree of freedom joystick about the size of a computer mouse). The keyboard is used to toggle between two modes: cursor movement or camera movement.

In the cursor movement mode, the Space Navigator moves the cursor along the terrain. The cursor lets the user place new symbols, delete or move existing symbols, and draw areas of interest in the battlefield. The location of the cursor is shown with crosshairs projected on the terrain to help the user easily locate it in the environment (Figure 1). The cursor also indicates scale and direction (a highlighted part of the crosshairs always points north).

In camera movement mode, the user can move the camera around the environment using either the Space Navigator or the keyboard and mouse. The Space Navigator provides 6 degrees of freedom, so it is sufficient to move the camera in any direction. However, preliminary tests showed that some users had difficulty controlling the Space Navigator, so we added the keyboard/mouse option for our final experiments. The mouse is used to rotate the camera angle and the arrow keys on the keyboard move the camera forward, backward, left, or right, relative to the camera's current orientation. Both the Space Navigator and the keyboard/mouse are active in camera movement mode, so the user can choose whichever they prefer or switch between them.

3.4 2D Mode

For comparison purposes, C2VE can also operate in 2D mode. That mode is similar to a flat map, because the terrain is not rendered in 3D. The user simply sees an overhead or bird's eye view of the aerial imagery. Elevation information is indicated by contour lines, which were generated from the `contourc` function of GNU Octave [21]. The user can still zoom and pan the map using the Space Navigator, but the camera is limited to a bird's eye vantage point.

4 Experiment Details

Once C2VE was developed, we designed and conducted a study to measure if and to what extent 3D vision was helpful in C2 applications. The study included two kinds of experimental tasks:

- Line of sight: The subject is asked if certain points on the map are visible from several observation posts that are marked on the map.
- Check routes: The subject is shown a map with several routes and asked to determine if each route is viable or not. They also mark down their certainty in their answers and highlight parts of the routes that they think are problematic. These tasks were developed in consultation with a captain in the U. S. Army who recently served as a convoy commander.

The experiment session is organized into 3 sets of tasks. The first set is a training task presented in 2.5D, which lets the user develop familiarity with the system. The second and third sets each use a different display depth (e.g., 2.5D then 3D), determined by the subject's randomly assigned group. Sets 2 and 3 each include a training task, a Line of Sight part, and a Check Routes part. To make the sets as independent as possible, each set uses different maps. Thus, the specific tasks differ between the sets, but they are matched in number and approximate difficulty.

There were 31 subjects. Each subject was randomly assigned into one of twelve groups. The groups differ by which display depth they used for set 2 tasks (2D, 2.5D, or 3D), which depth they used for set 3 tasks, and whether they used draped or billboard symbols throughout the experiments.

We used NVIDIA 3D Vision glasses with a 22 inch 3D-ready computer monitor (1920x1080 resolution). The subjects began the experiment by viewing training videos that described how to use the software and some factors to consider when choosing a convoy route (e.g., steepness of the terrain, sharpness of turns).

Each subject could earn a small monetary bonus (up to \$10) for correctly completing the tasks, which provided incentive to complete the tasks to the best of their ability. After finishing the tasks, each subject completed a survey about the software.

5 Analysis

We recorded how long it took the subjects to complete each task, the amount of camera and cursor movement, and the number of correct answers on the Line of Sight and Check Routes tasks. In addition, the subjects were asked to rank the display depths they used as part of their survey. We analyzed this data to determine the effects of symbol type (billboard or draped) and display depth (2D, 2.5D, or 3D) on users' performance.

5.1 Individuals' Performance Change with Different Display Depths

Each subject tested two display depths. This section examines the changes in their scores from one depth to the other. Before doing the comparison, we controlled for differences in the difficulty of the different task sets (i.e., set 2 and set 3) by subtracting the mean score for each part from the users' scores on that part. After this adjustment, the average score on each part is 0, with negative scores representing worse-than-average performance on a given task set.

Table 1. 2.5D - 3D Scores. Mean values and standard deviations. Positive values indicate an advantage of 3D over 2.5D

	Error	Time (sec.)	Movement
Line of Sight	7.8 ± 18.6	-60.5 ± 117.4	-0.090 ± 0.992
Routes	-1.1 ± 10.1	-115.4 ± 197.0	0.126 ± 3.080

The first comparison is between the 2D view and the other views (2.5D and 3D) for the Line of Sight tasks. Not surprisingly, the 2D view led to significantly worse accuracy than the other views. To come to this conclusion, we computed an error score for each Line of Sight question, taking into account the user's answer (yes or no) and their confidence in the answer (none, little, some, or a lot). The error score is 0 for a high-confidence correct answer, medium for a low-confidence incorrect answer, and high for a high-confidence incorrect answer.² Each subject completed two sets of Line of Sight tasks, each with a different display depth. For each subject who used 2D, we took their Line of Sight error in 2D and subtracted their Line of Sight error from their other display depth (2.5D or 3D). The resulting differences measure the improvement each person showed when using 2.5D or 3D instead of 2D. The average improvement of the 15 subjects was 16.0, and the standard deviation of the improvements was 16.3. A one-sample, two-tailed t-test shows that this is a significant difference ($p = 0.002$).³

The more interesting question is whether or not 3D leads to better performance than 2.5D. Thus, we analyzed the difference in each of 16 subjects' scores on 3D and 2.5D task sets. We looked at their error scores, the amount of camera and cursor movement,⁴ and the time it took the users to complete their tasks. Table 1 shows the mean and standard deviation for each task type and evaluation measure. These numbers are close to 0 relative to the standard deviation. The only statistically significant difference between 3D and 2.5D on any of these measures was the time it took to evaluate routes ($p = 0.033$ using a one-sample, two-tailed t-test), which was about two minutes higher for 3D. However, the standard deviation is fairly high, so the difference is not overwhelming.

² Each of the 8 possible responses was mapped to an integer score from -4 to 4. "No" answers were negative and "yes" answers were positive. The level of confidence determines the magnitude of the score, with a lot of confidence being ± 4 and no confidence being ± 1 . The correct answers had a lot of confidence, except for a few borderline routes where the best answer is to indicate lower confidence. The user's error for a question is the distance between the correct score and the user's score.

³ Statistical calculations were performed using R [22].

⁴ The overall camera/cursor movement measure was obtained by combining three quantities: linear camera movement (Euclidean distance), camera rotation (the angular movement in pitch and yaw), and cursor movement (Euclidean distance, discounting elevation). Each quantity was standardized so that it had mean 0 and standard deviation of 1 across all subjects and tasks. Then the standardized quantities were added together to get the overall movement score.

Table 2. Comparing 2.5D and 3D Scores on Set 3 Tasks. Medians (Q1–Q3) are reported. Significant differences are highlighted with italics.

	Error	Time (sec.)	Movement
Line of Sight, 3D	<i>1.0 (0.0–2.0)</i>	204.9 (165.9–252.0)	-2.77 (-2.94– -2.40)
Line of Sight, 2.5D	<i>8.0 (6.0–14.0)</i>	259.5 (235.6–282.1)	-2.29 (-2.51– -2.15)
Line of Sight, Billboard	3.5 (0.0–8.0)	<i>183.9 (144.7–243.7)</i>	-2.84 (-3.03– -2.51)
Line of Sight, Draped	1.5 (0.3–6.0)	254.2 (245.5–308.3)	-2.26 (-2.46– -1.67)
Routes, 3D	17 (11–22)	544 (395–580)	1.25 (0.90–2.15)
Routes, 2.5D	22 (10–27)	348 (301–491)	1.25 (0.07–2.58)

Much of the variance in these scores could be due to the fact that they include performance measures from task sets 2 and 3. From talking with the subjects after the experiments, we suspect that many subjects were still getting accustomed to the system during set 2, which would introduce variance into those scores. Thus, the next section looks at set 3 scores alone.

5.2 Comparing Groups: Display Depth and Symbol Type

While the previous section looked at the differences in each user's performance on two display depths, this section examines differences among the groups of subjects. We looked at their scores on the last set of tasks (i.e., Set 3), since that will give the best picture of users' performance after they have become accustomed to the system.

Since the 2D display depth was inferior for line of sight tasks and the users rated it the least useful on the survey, we excluded it from further analysis. This left 23 subjects for analysis. For the Check Routes tasks, there were no significant differences between 2.5D and 3D for the error score, time, or camera movement (Table 2).

For the Line of Sight tasks, we ran MANOVA for the three dependent variables (error, time, and movement) with the display depth (2.5D or 3D) and symbol type (billboard or draped) as the two factors. The MANOVA results indicated that each factor has a significant impact on at least one of the dependent variables (display depth $p = 0.028$ and symbol type $p = 0.019$). Figure 2 shows the data for each dependent variable, and Table 2 lists quartile numbers.

When we ran ANOVA for each dependent variable separately, we found that 3D leads to significantly lower error than 2.5D ($p = 0.002$), as seen in Table 2. The symbol type (billboard or draped) was significant for determining the time and the camera movement ($p = 0.013$ and $p = 0.004$, respectively), with billboard symbols taking less time and movement (Table 2).

5.3 User Survey

At the end of the experiment session, each user filled out a survey with several questions about the software. One question asked the user to indicate which of

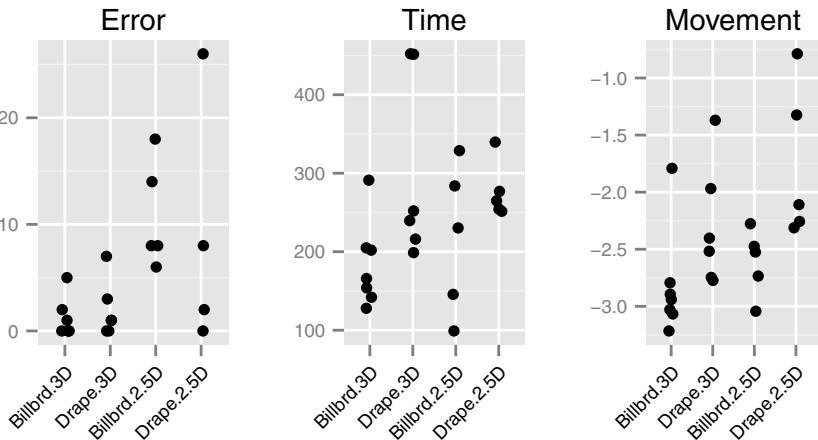


Fig. 2. Users' Scores for Set 3 Line of Sight tasks. Points are perturbed slightly along the x axis to avoid overlap.

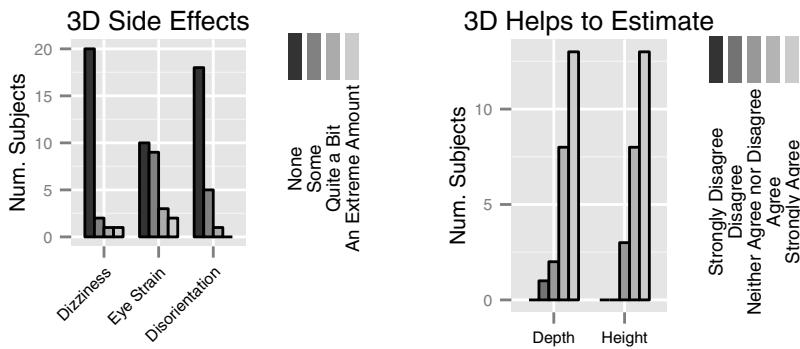


Fig. 3. Survey Results

the two display depths they used was better for completing their tasks. Everyone who used 2D said that it was the worse option. The remaining subjects used 2.5D and 3D, with 12 out of 16 subjects (75%) saying that 3D was better.

The 24 subjects who used 3D were asked to rate their levels of dizziness, eye strain, and disorientation (Figure 3). Only 2 subjects (8%) reported “Quite a Bit” or “An Extreme Amount” of dizziness or disorientation. An additional 3 subjects reported “Quite a Bit” or “An Extreme Amount” of eye strain. Those 5 subjects (21%) were the only ones to report more than “Some” of any side effect. In fact, most of the subjects (17 out of 24, 71%) reported no dizziness or disorientation. Of those, 8 subjects reported some eye strain, with the other 9 (38% overall) reporting no side effects at all.

The 3D users also rated the degree to which 3D helped them perceive the height and depth of the environment. A large majority (19 out of 24, 79%) agreed that 3D helped with both height and depth, a few subjects expressed neutral opinions, and only one user disagreed that 3D helped with distance estimation (Figure 3).

6 Conclusions

Our results provide evidence that 3D stereoscopic vision is beneficial for C2 line of sight tasks, and it does not inhibit performance on the route tasks. Furthermore, users generally preferred 3D to 2.5D and reported little discomfort from using 3D. Billboard style symbols led to more efficient decisions than draped symbols, without sacrificing accuracy.

Acknowledgments: This research is supported by the National Science Foundation under Grant No. 0968959. The Raytheon Company served as our corporate affiliate through NSF's Security and Software Engineering Research Center.

References

1. Grechkin, T.Y., Nguyen, T.D., Plumert, J.M., Cremer, J.F., Kearney, J.K.: How does presentation method and measurement protocol affect distance estimation in real and virtual environments? *ACM Transactions on Applied Perception* 7, Article No. 26 (2010)
2. Yang, M., McMullen, D.P., Schwartz-Bloom, R.D., Brady, R.: Dive into alcohol: A biochemical immersive experience. In: *IEEE Virtual Reality Conference*, pp. 281–282 (2009)
3. Chittaro, L., Ranon, R., Ieronutti, L.: Vu-flow: A visualization tool for analyzing navigation in virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 12, 1475–1485 (2006)
4. Schwartz, R.J., Fleming, G.A.: Real-time aerodynamic flow and data visualization in an interactive virtual environment. In: *IEEE Instrumentation and Measurement Technology Conference*, vol. 3, pp. 2210–2215 (2005)
5. Zelle, Z.M., Figura, C.: Simple, low-cost stereographics: VR for everyone. In: *The 35th SIGCSE Technical Symposium on Computer Science Education*, pp. 348–352 (2004)
6. Elmqvist, N., Philippas, T.: A taxonomy of 3D occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 1095–1109 (2008)
7. Cipiloglu, Z., Bulbul, A., Capin, T.: A framework for enhancing depth perception in computer graphics. In: *7th Symposium on Applied Perception in Graphics and Visualization*, pp. 141–148 (2010)
8. Jones, J.A., Swan II, J.E., Singh, G., Kolstad, E., Ellis, S.R.: The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception. In: *7th Symposium on Applied Perception in Graphics and Visualization*, pp. 9–14 (2008)

9. Wartell, Z., Hodges, L.F., Ribarsky, W.: A geometric comparison of algorithms for fusion control in stereoscopic HTDs. *IEEE Transactions on Visualization and Computer Graphics* 8, 129–143 (2002)
10. Livingston, M.A., Zhuming, A., Swan, J.E., Smallman, H.S.: Indoor vs. outdoor depth perception of mobile augmented reality. In: *IEEE Virtual Reality Conference*, pp. 55–62 (2009)
11. Livatino, S., Privitera, F.: 3D visualization technologies for teleguided robots. In: *ACM Symposium on Virtual Reality Software and Technology*, pp. 240–243 (2006)
12. Hartzell, T., Thompson, M., Kim, B.: The influence of graphics effects on perceiving depth in minimized virtual environments. In: Kim, T.-h., Ko, D.-s., Vasilakos, T., Stoica, A., Abawajy, J. (eds.) *FGCN/DCA 2012. CCIS*, vol. 350, pp. 235–242. Springer, Heidelberg (2012)
13. Sun, Y., Ding, N., Hao, G., Shi, X.: The research and application of 2D and 3D interactive system. In: *Second International Conference on Information and Computing Science*, pp. 252–254 (2009)
14. Cockburn, A., McKenzie, B.J.: 3D or not 3D? Evaluating the effect of the third dimension in a document management system. In: *SIGCHI Conference on Human Factors in Computing Systems*, pp. 434–441 (2001)
15. St. John, M., Cowen, M., Smallman, H., Oonk, H.: The use of 2D and 3D displays for shape understanding versus relative position tasks. *Human Factors* 43, 79–98 (2001)
16. Wolfe, B., Podgorniy, D., Kim, B., Sedlmeyer, R.: C2VE: A software platform for evaluating the use of 3D vision technology for C2 operations. In: *Proceedings of the 17th International Command and Control Research and Technology Symposium. CD-ROM* (2012)
17. U.S. Geological Survey: Seamless database (2012) (accessed March 2012)
18. Guth, P.: Microdem home page (2010) (accessed March 2012)
19. Blender Foundation: Blender (2013) (accessed March 2013)
20. Dept. of Defense: MIL-STD-2525C (2008) (accessed March 2012)
21. Octave community: GNU/Octave (2012)
22. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013)

Author Index

- Ababsa, Fakhr-Eddine I-562
Ababsa, Fakhreddine II-603
Abe, Toru II-308
Abeles, Peter II-454, II-552
Aboalsamh, Hatim II-493
Aehnelt, Mario II-76
Aeschliman, Benjamin II-747
Agranovsky, Alexy II-349
Ahmad, Touqeer I-181, II-663
Ait-Aoudia, Samy I-539
Alathari, Thamer S. I-238
Al-Hammadi, Muneer H. II-503
Almuzaini, Huda II-493
Alsam, Ali I-79
AlTarawneh, Ragaad 1
Angelakis, D. II-138
Apostolopoulos, Ilias II-254
Argyros, Antonis A. I-216, II-118
Atsumi, Masayasu I-416
Ayala, Orlando II-48
Ayyildiz, Kahraman II-725
Azary, Sherif II-189
- Barata, Catarina I-1, I-40
Bauer, Jens II-1
Bebis, George I-181, II-254, II-416, II-493, II-503, II-663
Behnke, Sven I-517
Belongie, Serge I-550, II-513
Belyaev, Alexander I-119, II-523
Bengherabi, Messaoud I-539
Bennett, Michael J. I-343
Ben Salah, Mohamed I-30
Bergamaschi, Sonia II-58
Bernard, Jürgen II-13, II-361
Bharthavarapu, Harika II-210
Bhojani, Faraz II-681
Bhowmick, Partha I-69
Bothorel, Gwenael II-396
Boudon, Frédéric I-322
Boulanger, Pierre I-30
Boutellaa, Elhocine I-539
Brazil, Emilio Vital II-384
Brimkov, Boris I-476
- Brimkov, Valentin E. I-246
Brun, Anders II-98
Burkhardt, Dirk II-86
- Cai, Caixia I-373
Camps, Octavia II-266
Carpendale, Sheelagh II-384
Catarci, Tiziana II-58
Ceron, Alexander II-484
Chanda, Bhabatosh I-507
Chang, Michael Ming-Yuen II-298
Chang, Yu-Han II-673
Charbit, Maurice I-562
Charvillat, Vincent I-322
Chen, Baowen I-333
Chen, Xiao I-437
Cheng, Jun I-333
Cheng, Zhi-Quan I-459
Cho, Sang-Hyun I-290
Choi, Jin Sung I-301
Chollet, Gérard I-562
Chotrov, Dimo II-288
Christiansen, Eric I-550, II-513
Cohrs, Moritz II-25
Conrad, Stefan II-725
Cox, Simon I-255
- Damer, Naser II-68
Dang, Tuan Nhon I-280
Dasgupta, Prithviraj I-160
Davies, I. Tudur I-255
De, Sourav I-69
Dean-León, Emmanuel I-373
Dee, Hannah I-572
Deufemia, Vincenzo II-128
Diamantas, Sotirios Ch. I-160
Dias, Miguel Sales I-385
Diaz, Idanis I-30
Dober, Johannes I-20
Doretto, Gianfranco II-210
Doulamis, Anastasios II-148
Doulamis, Nikolaos II-108
- Eaton, David II-384
Ebert, Achim II-1

- Economou, George I-496
 Eikel, Benjamin I-108, I-448
 Eleftheriadis, Stefanos I-527
 Elfarargy, Mohammed I-353
 Elster, Anne C. II-703
 Erdogan, Hakan II-178
 Ertl, Thomas II-373
 Fabian, Tomas II-222
 Feldmar, Jacques I-562
 Fellner, Dieter II-13
 Feng, Ke II-210
 Figueiredo, Mário A.T. II-475
 Fischer, Matthias I-108, I-448
 Flores, Arturo II-513
 Folmer, Eelke II-254
 Fonseca, Jose A.S I-129
 Fotopoulos, Spiros I-496
 Frey, Steffen II-373
 Fukuda, Hisato I-395
 Fukushi, Masaru II-308
 Fusek, Radovan II-199, II-425
 Gadermayr, Michael I-50, II-465
 Gaspar, Filipe I-385
 Gava, Christiano Couto I-363
 Gladisch, Stefan II-36
 Gong, Minglun II-703
 Grammatikopoulos, Lazaros II-148
 Greiner, Russell I-30
 Großmann, Bjarne II-318
 Guan, Hao I-407
 Gubbi, Jayavardhana II-613
 Guénard, Jérôme I-322
 Hadid, Abdenour I-539
 Hafiz, Rehan II-339
 Han, Li I-171
 Hao, Hongwei II-168
 Harizi, Farid I-539
 Havelock, Tom I-343
 Hegenscheid, Katrin I-20
 Hengen, Jean-Marc I-363
 Hirose, Naoto II-562
 Hoehn, Bret I-30
 Holuša, Michael I-228
 Holzapfel, Andre II-118
 Horejsi, Petr II-737
 Hu, Jiangyue I-171
 Huber-Mörk, Reinhold II-278
 Humayoun, Shah Rukh II-1
 Hurter, Christophe II-396
 Hussain, Muhammad II-416, II-493, II-503
 Ibrahim, Alaa Eldin M. II-623
 Ivanovska, Tatyana I-20
 Jähn, Claudius I-108, I-448
 Jensen, Rune II-703
 Jeong, Ilkwon II-446
 Jiang, Guannan II-435
 Jiang, Jun I-333
 Joy, Kenneth I. II-349
 Kachi, Daisuke I-395
 Kambhamettu, Chandra I-312, II-48
 Kán, Peter II-328
 Kang, Hang-Bong I-290
 Kaufmann, Hannes II-328
 Kawanaka, Akira II-562
 Keller, Patric II-1
 Ki, Kai Lee II-298
 Kim, Beomjin II-747
 Kim, Jae-Hean I-301
 Kim, Jaehwan II-446
 Kimani, Stephen II-58
 Klimke, Stefan II-25
 Knoll, Alois I-373
 Kobayashi, Yoshinori I-395
 Kohlhammer, Jörn II-13, II-86, II-361
 Kolagunda, Abhishek I-312
 Komodakis, Nikos I-194
 Kong, Dexing II-653
 Konstantaras, A. II-138
 Koo, Bon-Ki I-301
 Kormann, M. II-244
 Koutlemanis, P. I-216
 Kriegman, David I-550, II-513
 Kritsotaki, A. II-138
 Krüger, Volker II-318
 Kuijper, Arjan II-86, II-361
 Kuno, Yoshinori I-395
 Kuo, Chia-Ning II-406
 Kuznetsova, Alina II-592
 Kwak, Iljung S. I-550
 Kwok, Ngai Ming II-435
 Laqua, René I-20
 Laramee, Robert S. I-255
 Lee, Chulhee II-532
 Lee, Jaehoon II-532

- Lemos, João M. II-140, II-475
 Leva, Mariano II-58
 Li, Chang-Tsun II-542
 Li, Lin I-171
 Liao, Shenghui I-60
 Lin, Juncong I-60
 Lin, Stephen Ching-Feng II-435
 Linder, Thorsten I-517
 Lipşa, Dan R. I-255
 Liu, Wanmu I-343
 Livanos, Georgios II-148
 Lou, Qiong II-653
 Lu, Guoyu I-312, II-48
 Ly, Vincent I-312, II-48
- Mahdavi, Amiri Ali II-681
 Maher, Andrew II-613
 Maheswaran, Rajiv II-673
 Mahmoodi, Sasan I-343
 Malagoli, Alberto II-58
 Maleshkov, Stoyan II-288
 Mandava, Ajay Kumar II-663
 Maragudakis, Andreas II-714
 Maravelakis, E. II-138
 Marques, Jorge S. I-1, I-40, I-140, II-475
 Marusic, Slaven II-613
 Mecella, Massimo II-58
 Metaxas, Dimitris I-486
 Meyer auf der Heide, Friedhelm I-108, I-448
 Miaoulis, George II-108
 Michel, Damien II-118
 Mignot, Cyrille II-603
 Miller, Ben II-693
 Mirza, Anwar M. II-493
 Mori, Satoshi I-395
 Morin, Géraldine I-322
 Morioka, Kotaro I-427
 Mostafa, Ahmed E. II-384
 Motiian, Saeid II-210
 Mozdřeň, Karel II-199, II-425
 Muhammad, Ghulam II-416, II-493, II-503
 Murtha, Albert I-30
 Mussadiq, Shafiq II-339
 M.V., Rohith II-48
- Nagaoka, Tomoaki II-633
 Nait-Charif, Hammadi I-129
 Nalpantidis, Lazaros II-318
- Nazemi, Kawa II-13, II-86
 Nefian, Ara I-181
 Nguyen, Khoa Tan I-266
 Nixon, Mark S. I-238
 Nouak, Alexander II-68
 Ntelidakis, A. I-216
- Obermaier, Harald II-349
 Ohtake, Yutaka I-427
 Oikonomopoulos, Antonios I-150
 Opel, Alexander II-68
- Pacheco, Sandra II-475
 Palágyi, Kálmán I-87
 Palaniswami, Marimuthu II-613
 Panagiotakis, Costas II-118
 Pantic, Maja I-150, I-527, II-234
 Paolino, Luca II-128
 Paragios, Nikos I-194
 Pavlovic, Vladimir II-234
 Peng, Jialin II-653
 Petring, Ralf I-108, I-448
 Petrovska-Delacrétaz, Dijana I-562
 Polcar, Jiri II-737
 Porikli, Fatih II-178
 Portelo, Ana II-475
 Porter-Sobieraj, Joanna I-99
 Poudel, Rudra P.K. I-129
 Prieto, Flavio II-484
 Proença, Pedro F. I-385
 Protopapadakis, Eftychios II-108, II-148
- Ramakrishna, Anil II-673
 Rao, Aravinda S. II-613
 Rashwan, Marwa I-353
 Rasmussen, A. I-206
 Regentova, Emma E. I-181, II-663
 Ren, Peng I-407
 Retz, Reimond II-13
 Riaz, Zahid I-517
 Ribeiro, Ricardo A. I-140
 Rivertz, Hans Jakob I-79
 Rizq, Amr I-353
 Robertson, Neil M. I-119, II-523
 Rodrigues, M. II-244
 Ropinski, Timo I-266
 Rosenhahn, Bodo II-592
 Rozeira, Jorge I-40
 Rudovic, Ognjen I-527, II-234

- Ruela, Margarida I-1
 Ruppert, Tobias II-361
 Russo, Alessandro II-58
 Ryciuk, Marcin I-99
- Sadlo, Filip II-373
 Saleh, Sahar Q. II-416
 Samanta, Soumitra I-507
 Samavati, Faramarz II-681
 Savakis, Andreas II-189
 Savidis, Anthony II-714
 Schindler, Alexander II-278
 Schuhler, C. II-244
 Schulz, Hans-Jörg II-76
 Schumann, Heidrun II-36
 Seck, Alassane I-572
 Sedlmeyer, Robert II-747
 Seo, Guiwon II-532
 Serrurier, Mathieu II-396
 Sharlemin, Sajid II-210
 Sharlin, Ehud II-384
 Shen, Haoquan I-312
 Shrestha, Ayush II-693
 Sluzek, Andrzej II-643
 Smith, William A.P. I-407
 Sørensen, T.S. I-206
 Sojka, Eduard I-228, II-199, II-425
 Somanı, Nikhil I-373
 Soudani, Amira II-158
 Sousa, Mario Costa II-384
 Stab, Christian II-86
 Steiger, Martin II-86
 Stentoumis, Christos II-148
 Stricker, Didier I-363
 Strong, Grant II-703
 Šurkala, Milan II-199, II-425
 Surmann, Hartmut I-517
 Suzuki, Hiromasa I-427
- Taetz, Bertram I-363
 Tang, Chu II-168
 Theodorakopoulos, Ilias I-496
 Tian, Shu II-168
 Tiddeman, Bernard I-572
 Tomek, P. II-244
 Tominski, Christian II-36
 Topkaya, Ibrahim Saygin II-178
 Tran, Ngoc-Trung I-562
 Triki, Olfa II-572
- Ueda, Daisuke II-308
 Uhl, Andreas I-50, II-465
 Ulmer, Alex II-361
 Urban, Bodo II-76
- Vasile, Alexandru N. II-266
 Vécsei, Andreas I-50, II-465
 Völzke, Henry I-20
- Wahlberg, Fredrik II-98
 Wang, Lian-Ping II-48
 Wang, Xiaolong II-48
 Whytock, Tenika P. I-119, II-523
 Wilkinson, Leland I-280
 Wolfe, Britton II-747
 Wong, Chin Yeow II-435
 Wong, Kin-Hong II-298, II-582
 Worst, Rainer I-517
 Wu, Fa II-653
 Wu, Xing I-11
- Xia, Jiazhi I-60, I-459, I-468
 Xiang, Bo I-194
 Xinogalos, M. II-138
- Yang, Chuan-Kai II-406
 Yang, Fei I-486
 Yao, Yi II-542
 Yasunobe, Tatsuki II-562
 Yazdanpanah, Ali Pour II-663
 Yiakoumettis, Christos II-108
 Ynnerman, Anders I-266
 Yu, Yang I-486
 Yu, Ying-Kin II-298
 Yuan, Fei II-168
- Zabulis, X. I-216
 Zachmann, Gabriel II-25
 Zagrouba, Ezzeddine II-158
 Zéraï, Mourad II-572
 Zervakis, Michael II-148
 Zhang, Jian J. I-129
 Zhang, Shaoting I-486
 Zhang, Wu I-11
 Zhao, Yi II-693
 Zhu, Lei II-582
 Zhu, Ying I-437, II-693
 Zhuo, Shaojian I-11
 Ziegler, G. I-206