

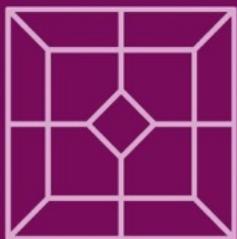
Tomáš Pajdla
Jiří Matas (Eds.)

LNCS 3023

Computer Vision – ECCV 2004

8th European Conference on Computer Vision
Prague, Czech Republic, May 2004
Proceedings, Part III

3
Part III



ECCV 2004



Springer

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

New York University, NY, USA

Doug Tygar

University of California, Berkeley, CA, USA

Moshe Y. Vardi

Rice University, Houston, TX, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Springer

Berlin

Heidelberg

New York

Hong Kong

London

Milan

Paris

Tokyo

Tomáš Pajdla Jiří Matas (Eds.)

Computer Vision – ECCV 2004

8th European Conference on Computer Vision
Prague, Czech Republic, May 11-14, 2004
Proceedings, Part III



Springer

Volume Editors

Tomáš Pajdla

Jiří Matas

Czech Technical University in Prague, Department of Cybernetics

Center for Machine Perception

121-35 Prague 2, Czech Republic

E-mail: {pajdla,matas}@cmp.felk.cvut.cz

Library of Congress Control Number: 2004104846

CR Subject Classification (1998): I.4, I.3.5, I.5, I.2.9-10

ISSN 0302-9743

ISBN 3-540-21982-X Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11007753 06/3142 5 4 3 2 1 0

Preface

Welcome to the proceedings of the 8th European Conference on Computer Vision!

Following a very successful ECCV 2002, the response to our call for papers was almost equally strong – 555 papers were submitted. We accepted 41 papers for oral and 149 papers for poster presentation.

Several innovations were introduced into the review process. First, the number of program committee members was increased to reduce their review load. We managed to assign to program committee members no more than 12 papers. Second, we adopted a paper ranking system. Program committee members were asked to rank all the papers assigned to them, even those that were reviewed by additional reviewers. Third, we allowed authors to respond to the reviews consolidated in a discussion involving the area chair and the reviewers. Fourth, the reports, the reviews, and the responses were made available to the authors as well as to the program committee members. Our aim was to provide the authors with maximal feedback and to let the program committee members know how authors reacted to their reviews and how their reviews were or were not reflected in the final decision. Finally, we reduced the length of reviewed papers from 15 to 12 pages.

The preparation of ECCV 2004 went smoothly thanks to the efforts of the organizing committee, the area chairs, the program committee, and the reviewers. We are indebted to Anders Heyden, Mads Nielsen, and Henrik J. Nielsen for passing on ECCV traditions and to Dominique Asselineau from ENST/TSI who kindly provided his GestRFIA conference software. We thank Jan-Olof Eklundh and Andrew Zisserman for encouraging us to organize ECCV 2004 in Prague. Andrew Zisserman also contributed many useful ideas concerning the organization of the review process. Olivier Faugeras represented the ECCV Board and helped us with the selection of conference topics. Kyros Kutulakos provided helpful information about the CVPR 2003 organization. David Vernon helped to secure ECVision support.

This conference would never have happened without the support of the Centre for Machine Perception of the Czech Technical University in Prague. We would like to thank Radim Šára for his help with the review process and the proceedings organization. We thank Daniel Večerka and Martin Matoušek who made numerous improvements to the conference software. Petr Pohl helped to put the proceedings together. Martina Budošová helped with administrative tasks. Hynek Bakstein, Ondřej Chum, Jana Kostková, Branislav Mičušík, Štěpán Obdržálek, Jan Šochman, and Vít Zýka helped with the organization.

Organization

Conference Chair

Václav Hlaváč

CTU Prague, Czech Republic

Program Chairs

Tomáš Pajdla

Jiří Matas

CTU Prague, Czech Republic

CTU Prague, Czech Republic

Organization Committee

Tomáš Pajdla

Radim Šára

Vladimír Smutný

Eva Matysková

Jiří Matas

Václav Hlaváč

Workshops, Tutorials

Budget, Exhibition

Local Arrangements

CTU Prague, Czech Republic

Conference Board

Hans Burkhardt

Bernard Buxton

Roberto Cipolla

Jan-Olof Eklundh

Olivier Faugeras

Anders Heyden

Bernd Neumann

Mads Nielsen

Giulio Sandini

David Vernon

University of Freiburg, Germany

University College London, UK

University of Cambridge, UK

Royal Institute of Technology, Sweden

INRIA, Sophia Antipolis, France

Lund University, Sweden

University of Hamburg, Germany

IT University of Copenhagen, Denmark

University of Genoa, Italy

Trinity College, Ireland

Area Chairs

Dmitry Chetverikov

Kostas Daniilidis

Rachid Deriche

Jan-Olof Eklundh

Luc Van Gool

Richard Hartley

MTA SZTAKI, Hungary

University of Pennsylvania, USA

INRIA Sophia Antipolis, France

KTH Stockholm, Sweden

KU Leuven, Belgium & ETH Zürich, Switzerland

Australian National University, Australia

Michal Irani	Weizmann Institute of Science, Israel
Sing Bing Kang	Microsoft Research, USA
Aleš Leonardis	University of Ljubljana, Slovenia
Stan Li	Microsoft Research China, Beijing, China
David Lowe	University of British Columbia, Canada
Mads Nielsen	IT University of Copenhagen, Denmark
Long Quan	HKUST, Hong Kong, China
Jose Santos-Victor	Instituto Superior Tecnico, Portugal
Cordelia Schmid	INRIA Rhône-Alpes, France
Steven Seitz	University of Washington, USA
Amnon Shashua	Hebrew University of Jerusalem, Israel
Stefano Soatto	UCLA, Los Angeles, USA
Joachim Weickert	Saarland University, Germany
Andrew Zisserman	University of Oxford, UK

Program Committee

Jorgen Ahlberg	Joachim Buhmann	Alexei Efros
Narendra Ahuja	Hans Burkhardt	Irfan Essa
Yiannis Aloimonos	Aurelio Campilho	Michael Felsberg
Arnon Amir	Octavia Camps	Cornelia Fermueller
Elli Angelopoulou	Stefan Carlsson	Mario Figueiredo
Helder Araujo	Yaron Caspi	Bob Fisher
Tal Arbel	Tat-Jen Cham	Andrew Fitzgibbon
Karl Astrom	Mike Chantler	David Fleet
Shai Avidan	Francois Chaumette	Wolfgang Foerstner
Simon Baker	Santanu Choudhury	David Forsyth
Subhashis Banerjee	Laurent Cohen	Pascal Fua
Kobus Barnard	Michael Cohen	Dariu Gavrila
Ronen Basri	Bob Collins	Jan-Mark Geusebroek
Serge Belongie	Dorin Comaniciu	Christopher Geyer
Marie-Odile Berger	Tim Cootes	Georgy Gimelfarb
Horst Bischof	Joao Costeira	Frederic Guichard
Michael J. Black	Daniel Cremers	Gregory Hager
Andrew Blake	Antonio Criminisi	Allan Hanbury
Laure Blanc-Feraud	James Crowley	Edwin Hancock
Aaron Bobick	Kristin Dana	Horst Haussecker
Rein van den Boomgaard	Trevor Darrell	Eric Hayman
Terrance Boult	Larry Davis	Martial Hebert
Richard Bowden	Fernando De la Torre	Bernd Heisele
Edmond Boyer	Frank Dellaert	Anders Heyden
Mike Brooks	Joachim Denzler	Adrian Hilton
Michael Brown	Greg Dudek	David Hogg
Alfred Bruckstein	Chuck Dyer	Atsushi Imiya

Michael Isard	Nassir Navab	Jon Sporring
Yuri Ivanov	Shree Nayar	Charles Stewart
David Jacobs	Ko Nishino	Peter Sturm
Allan D. Jepson	David Nister	Changming Sun
Peter Johansen	Ole Fogh Olsen	Tomas Svoboda
Nebojsa Jojic	Theodore Papadopoulos	Rahul Swaminathan
Frederic Jurie	Nikos Paragios	Richard Szeliski
Fredrik Kahl	Shmuel Peleg	Tamas Sziranyi
Daniel Keren	Francisco Perales	Chi-keung Tang
Benjamin Kimia	Nicolas Perez	Hai Tao
Ron Kimmel	de la Blanca	Sibel Tari
Nahum Kiryati	Pietro Perona	Chris Taylor
Georges Koepfler	Matti Pietikainen	C.J. Taylor
Pierre Kornprobst	Filiberto Pla	Bart ter Haar Romeny
David Kriegman	Robert Pless	Phil Torr
Walter Kropatsch	Marc Pollefeys	Antonio Torralba
Rakesh Kumar	Jean Ponce	Panos Trahanias
David Liebowitz	Ravi Ramamoorthi	Bill Triggs
Tony Lindeberg	James Rehg	Emanuele Trucco
Jim Little	Ian Reid	Dimitris Tsakiris
Yanxi Liu	Tammy Riklin-Raviv	Yanghai Tsin
Yi Ma	Ehud Rivlin	Matthew Turk
Claus Madsen	Nicolas Rougon	Tinne Tuytelaars
Tom Malzbender	Yong Rui	Nuno Vasconcelos
Jorge Marques	Javier Sanchez	Baba C. Vemuri
David Marshall	Guillermo Sapiro	David Vernon
Bogdan Matei	Yoichi Sato	Alessandro Verri
Steve Maybank	Eric Saund	Rene Vidal
Gerard Medioni	Otmar Scherzer	Jordi Vitria
Etienne Memin	Bernt Schiele	Yair Weiss
Rudolf Mester	Mikhail Schlesinger	Tomas Werner
Krystian Mikolajczyk	Christoph Schnoerr	Carl-Fredrik Westin
J.M.M. Montiel	Stan Sclaroff	Ross Whitaker
Theo Moons	Mubarak Shah	Lior Wolf
Pavel Mrazek	Eitan Sharon	Ying Wu
Joe Mundy	Jianbo Shi	Ming Xie
Vittorio Murino	Kaleem Siddiqi	Ramin Zabih
David Murray	Cristian Sminchisescu	Assaf Zomet
Hans-Helmut Nagel	Nir Sochen	Steven Zucker
Vic Nalwa	Gerald Sommer	
P.J. Narayanan	Gunnar Sparr	

Additional Reviewers

Lourdes Agapito	Claudio Fanti	Jocelyn Marchadier
Manoj Aggarwal	Michela Farenzena	Scott McCloskey
Parvez Ahammad	Doron Feldman	Leonard McMillan
Fernando Alegre	Darya Frolova	Marci Meingast
Jonathan Alon	Andrea Fusillo	Anurag Mittal
Hans Jorgen Andersen	Chunyu Gao	Thomas B. Moeslund
Marco Andreetto	Kshitiz Garg	Jose Montiel
Anelia Angelova	Yoram Gat	Philippos Mordohai
Himanshu Arora	Dan Gelb	Pierre Moreels
Thangali Ashwin	Ya'ara Goldschmidt	Hesam Najafi
Vassilis Athitsos	Michael E. Goss	P.J. Narayanan
Henry Baird	Leo Grady	Ara Nefian
Harlyn Baker	Sertan Grigin	Oscar Nestares
Evgeniy Bart	Michael Grossberg	Michael Nielsen
Moshe Ben-Ezra	J.J. Guerrero	Peter Nillius
Manuele Bicego	Guodong Guo	Fredrik Nyberg
Marten Björkman	Yanlin Guo	Tom O'Donnell
Paul Blaer	Robert Hanek	Eyal Ofek
Ilya Blayvas	Matthew Harrison	Takahiro Okabe
Eran Borenstein	Tal Hassner	Kazunori Okada
Lars Bretzner	Horst Hausscker	D. Ortin
Alexia Briassouli	Yakov Hel-Or	Patrick Perez
Michael Bronstein	Anton van den Hengel	Christian Perwass
Rupert Brooks	Tat Jen Cham	Carlos Phillips
Gabriel Brostow	Peng Chang	Srikumar Ramalingam
Thomas Brox	John Isidoro	Alex Rav-Acha
Stephanie Brubaker	Vishal Jain	Stefan Roth
Andres Bruhn	Marie-Pierre Jolly	Ueli Rutishauser
Darius Burschka	Michael Kaess	C. Sagues
Umberto Castellani	Zia Khan	Garbis Salgian
J.A. Castellanos	Kristian Kirk	Ramin Samadani
James Clark	Dan Kong	Bernard Sarel
Andrea Colombari	B. Kröse	Frederik Schaffalitzky
Marco Cristani	Vivek Kwatra	Adam Seeger
Xiangtian Dai	Michael Langer	Cheng Dong Seon
David Demirdjian	Catherine Laporte	Ying Shan
Maxime Descoteaux	Scott Larsen	Eli Shechtman
Nick Diakopoulos	Barbara Levienaise-	Grant Schindler
Anthony Dicks	Obadia	Nils T. Siebel
Carlotta Domeniconi	Frederic Leymarie	Leonid Sigal
Roman Dovgard	Fei-Fei Li	Greg Slabaugh
R. Dugad	Rui Li	Ben Southall
Ramani Duraiswami	Kok-Lim Low	Eric Spellman
Kerrien Erwan	Le Lu	Narasimhan Srinivasa

Drew Steedly	Zhizhou Wang	Ruigang Yang
Moritz Stoerring	Joost van de Weijer	Yll Haxhimusa
David Suter	Wolfgang Wein	Tianli Yu
Yi Tan	Martin Welk	Lihi Zelnik-Manor
Donald Tanguay	Michael Werman	Tao Zhao
Matthew Toews	Horst Wildenauer	Wenyi Zhao
V. Javier Traver	Christopher R. Wren	Sean Zhou
Yaron Ukrainitz	Ning Xu	Yue Zhou
F.E. Wang	Hulya Yalcin	Ying Zhu
Hongcheng Wang	Jingyu Yan	

Sponsors

BIG - Business Information Group a.s.
Camea spol. s r.o.
Casablanca INT s.r.o.
ECVision – European Research Network for Cognitive Computer Vision Systems
Microsoft Research
Miracle Network s.r.o.
Neovision s.r.o.
Toyota

Table of Contents – Part III

Learning and Recognition

A Constrained Semi-supervised Learning Approach to Data Association	1
<i>Hendrik Kück, Peter Carbonetto, Nando de Freitas</i>	
Learning Mixtures of Weighted Tree-Unions by Minimizing Description Length	13
<i>Andrea Torsello, Edwin R. Hancock</i>	
Decision Theoretic Modeling of Human Facial Displays	26
<i>Jesse Hoey, James J. Little</i>	
Kernel Feature Selection with Side Data Using a Spectral Approach	39
<i>Amnon Shashua, Lior Wolf</i>	

Tracking II

Tracking Articulated Motion Using a Mixture of Autoregressive Models	54
<i>Ankur Agarwal, Bill Triggs</i>	
Novel Skeletal Representation for Articulated Creatures	66
<i>Gabriel J. Brostow, Irfan Essa, Drew Steedly, Vivek Kwatra</i>	
An Accuracy Certified Augmented Reality System for Therapy Guidance	79
<i>Stéphane Nicolau, Xavier Pennec, Luc Soler, Nichlas Ayache</i>	

Posters III

3D Human Body Tracking Using Deterministic Temporal Motion Models	92
<i>Raquel Urtasun, Pascal Fua</i>	
Robust Fitting by Adaptive-Scale Residual Consensus	107
<i>Hanzi Wang, David Suter</i>	
Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures	119
<i>Tal Nir, Alfred M. Bruckstein</i>	

An Adaptive Window Approach for Image Smoothing and Structures Preserving	132
<i>Charles Kervrann</i>	
Extraction of Semantic Dynamic Content from Videos with Probabilistic Motion Models	145
<i>Gwenaëlle Piriou, Patrick Bouthemy, Jian-Feng Yao</i>	
Are Iterations and Curvature Useful for Tensor Voting?	158
<i>Sylvain Fischer, Pierre Bayerl, Heiko Neumann, Gabriel Cristóbal, Rafael Redondo</i>	
A Feature-Based Approach for Determining Dense Long Range Correspondences	170
<i>Josh Wills, Serge Belongie</i>	
Combining Geometric- and View-Based Approaches for Articulated Pose Estimation	183
<i>David Demirdjian</i>	
Shape Matching and Recognition – Using Generative Models and Informative Features	195
<i>Zhuowen Tu, Alan L. Yuille</i>	
Generalized Histogram: Empirical Optimization of Low Dimensional Features for Image Matching	210
<i>Shin'ichi Satoh</i>	
Recognizing Objects in Range Data Using Regional Point Descriptors ...	224
<i>Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, Jitendra Malik</i>	
Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces	238
<i>Ye Duan, Liu Yang, Hong Qin, Dimitris Samaras</i>	
Structure and Motion Problems for Multiple Rigidly Moving Cameras ...	252
<i>Henrik Stewenius, Kalle Åström</i>	
Detection and Tracking Scheme for Line Scratch Removal in an Image Sequence	264
<i>Bernard Besserer, Cedric Thiré</i>	
Color Constancy Using Local Color Shifts	276
<i>Marc Ebner</i>	
Image Anisotropic Diffusion Based on Gradient Vector Flow Fields	288
<i>Hongchuan Yu, Chin-Seng Chua</i>	

Optimal Importance Sampling for Tracking in Image Sequences: Application to Point Tracking	302
<i>Elise Arnaud, Etienne Mémin</i>	
Learning to Segment	315
<i>Eran Borenstein, Shimon Ullman</i>	
MCMC-Based Multiview Reconstruction of Piecewise Smooth Subdivision Curves with a Variable Number of Control Points	329
<i>Michael Kaess, Rafal Zbownik, Frank Dellaert</i>	
Bayesian Correction of Image Intensity with Spatial Consideration	342
<i>Jiaya Jia, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum</i>	
Stretching Bayesian Learning in the Relevance Feedback of Image Retrieval	355
<i>Ruofei Zhang, Zhongfei (Mark) Zhang</i>	
Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera	368
<i>Antonis A. Argyros, Manolis I.A. Lourakis</i>	
Evaluation of Image Fusion Performance with Visible Differences	380
<i>Vladimir Petrović, Costas Xydeas</i>	
An Information-Based Measure for Grouping Quality	392
<i>Erik A. Engbers, Michael Lindenbaum, Arnold W.M. Smeulders</i>	
Bias in Shape Estimation	405
<i>Hui Ji, Cornelia Fermüller</i>	
Contrast Marginalised Gradient Template Matching	417
<i>Saleh Basalamah, Anil Bharath, Donald McRobbie</i>	
The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition	430
<i>Nuno Vasconcelos, Purdy Ho, Pedro Moreno</i>	
Partial Object Matching with Shapeme Histograms	442
<i>Y. Shan, H.S. Sawhney, B. Matei, R. Kumar</i>	
Modeling and Synthesis of Facial Motion Driven by Speech	456
<i>Payam Saisan, Alessandro Bissacco, Alessandro Chiuso, Stefano Soatto</i>	
Recovering Local Shape of a Mirror Surface from Reflection of a Regular Grid	468
<i>Silvio Savarese, Min Chen, Pietro Perona</i>	

Structure of Applicable Surfaces from Single Views	482
<i>Nail Gumerov, Ali Zandifar, Ramani Duraiswami, Larry S. Davis</i>	
Joint Bayes Filter: A Hybrid Tracker for Non-rigid Hand Motion Recognition	497
<i>Huang Fei, Ian Reid</i>	
Iso-disparity Surfaces for General Stereo Configurations	509
<i>Marc Pollefeys, Sudipta Sinha</i>	
Camera Calibration with Two Arbitrary Coplanar Circles	521
<i>Qian Chen, Haiyuan Wu, Toshikazu Wada</i>	
Reconstruction of 3-D Symmetric Curves from Perspective Images without Discrete Features	533
<i>Wei Hong, Yi Ma, Yizhou Yu</i>	
A Topology Preserving Non-rigid Registration Method Using a Symmetric Similarity Function-Application to 3-D Brain Images	546
<i>Vincent Noblet, Christian Heinrich, Fabrice Heitz, Jean-Paul Armspach</i>	
A Correlation-Based Approach to Robust Point Set Registration	558
<i>Yanghai Tsin, Takeo Kanade</i>	
Hierarchical Organization of Shapes for Efficient Retrieval	570
<i>Shantanu Joshi, Anuj Srivastava, Washington Mio, Xiuwen Liu</i>	
Information-Based Image Processing	
Intrinsic Images by Entropy Minimization	582
<i>Graham D. Finlayson, Mark S. Drew, Cheng Lu</i>	
Image Similarity Using Mutual Information of Regions	596
<i>Daniel B. Russakoff, Carlo Tomasi, Torsten Rohlfing, Calvin R. Maurer, Jr.</i>	
Author Index	609

Table of Contents – Part I

Tracking I

A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation	1
<i>René Vidal, Yi Ma</i>	
Enhancing Particle Filters Using Local Likelihood Sampling	16
<i>Péter Torma, Csaba Szepesvári</i>	
A Boosted Particle Filter: Multitarget Detection and Tracking	28
<i>Kenji Okuma, Ali Taleghani, Nando de Freitas, James J. Little, David G. Lowe</i>	

Feature-Based Object Detection and Recognition I

Simultaneous Object Recognition and Segmentation by Image Exploration	40
<i>Vittorio Ferrari, Tinne Tuytelaars, Luc Van Gool</i>	
Recognition by Probabilistic Hypothesis Construction	55
<i>Pierre Moreels, Michael Maire, Pietro Perona</i>	
Human Detection Based on a Probabilistic Assembly of Robust Part Detectors	69
<i>Krystian Mikolajczyk, Cordelia Schmid, Andrew Zisserman</i>	

Posters I

Model Selection for Range Segmentation of Curved Objects	83
<i>Alireza Bab-Hadiashar, Niloofar Gheissari</i>	
High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging	95
<i>Changsoo Je, Sang Wook Lee, Rae-Hong Park</i>	
Using Inter-feature-Line Consistencies for Sequence-Based Object Recognition	108
<i>Jiun-Hung Chen, Chu-Song Chen</i>	
Discriminant Analysis on Embedded Manifold	121
<i>Shuicheng Yan, Hongjiang Zhang, Yuxiao Hu, Benyu Zhang, Qiansheng Cheng</i>	

Multiscale Inverse Compositional Alignment for Subdivision Surface Maps	133
<i>Igor Guskov</i>	
A Fourier Theory for Cast Shadows	146
<i>Ravi Ramamoorthi, Melissa Koudelka, Peter Belhumeur</i>	
Surface Reconstruction by Propagating 3D Stereo Data in Multiple 2D Images	163
<i>Gang Zeng, Sylvain Paris, Long Quan, Maxime Lhuillier</i>	
Visibility Analysis and Sensor Planning in Dynamic Environments	175
<i>Anurag Mittal, Larry S. Davis</i>	
Camera Calibration from the Quasi-affine Invariance of Two Parallel Circles	190
<i>Yihong Wu, Haijiang Zhu, Zhanyi Hu, Fuchao Wu</i>	
Texton Correlation for Recognition	203
<i>Thomas Leung</i>	
Multiple View Feature Descriptors from Image Sequences via Kernel Principal Component Analysis	215
<i>Jason Meltzer, Ming-Hsuan Yang, Rakesh Gupta, Stefano Soatto</i>	
An Affine Invariant Salient Region Detector	228
<i>Timor Kadir, Andrew Zisserman, Michael Brady</i>	
A Visual Category Filter for Google Images	242
<i>Robert Fergus, Pietro Perona, Andrew Zisserman</i>	
Scene and Motion Reconstruction from Defocused and Motion-Blurred Images via Anisotropic Diffusion	257
<i>Paolo Favaro, Martin Burger, Stefano Soatto</i>	
Semantics Discovery for Image Indexing	270
<i>Joo-Hwee Lim, Jesse S. Jin</i>	
Hand Gesture Recognition within a Linguistics-Based Framework	282
<i>Konstantinos G. Derpanis, Richard P. Wildes, John K. Tsotsos</i>	
Line Geometry for 3D Shape Understanding and Reconstruction	297
<i>Helmut Pottmann, Michael Hofer, Boris Odehnal, Johannes Wallner</i>	
Extending Interrupted Feature Point Tracking for 3-D Affine Reconstruction	310
<i>Yasuyuki Sugaya, Kenichi Kanatani</i>	

Many-to-Many Feature Matching Using Spherical Coding of Directed Graphs	322
<i>M. Fatih Demirci, Ali Shokoufandeh, Sven Dickinson, Yakov Keselman, Lars Bretzner</i>	
Coupled-Contour Tracking through Non-orthogonal Projections and Fusion for Echocardiography	336
<i>Xiang Sean Zhou, Dorin Comaniciu, Sriram Krishnan</i>	
A Statistical Model for General Contextual Object Recognition	350
<i>Peter Carbonetto, Nando de Freitas, Kobus Barnard</i>	
Reconstruction from Projections Using Grassmann Tensors	363
<i>Richard I. Hartley, Fred Schaffalitzky</i>	
Co-operative Multi-target Tracking and Classification	376
<i>Pankaj Kumar, Surendra Ranganath, Kuntal Sengupta, Huang Weimin</i>	
A Linguistic Feature Vector for the Visual Interpretation of Sign Language	390
<i>Richard Bowden, David Windridge, Timor Kadir, Andrew Zisserman, Michael Brady</i>	
Fast Object Detection with Occlusions	402
<i>Yen-Yu Lin, Tyng-Luh Liu, Chiou-Shann Fuh</i>	
Pose Estimation of Free-Form Objects	414
<i>Bodo Rosenhahn, Gerald Sommer</i>	
Interactive Image Segmentation Using an Adaptive GMMRF Model	428
<i>Andrew Blake, Carsten Rother, M. Brown, Patrick Perez, Philip Torr</i>	
Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model	442
<i>Xianghua Ying, Zhanyi Hu</i>	
Image Clustering with Metric, Local Linear Structure, and Affine Symmetry	456
<i>Jongwoo Lim, Jeffrey Ho, Ming-Hsuan Yang, Kuang-chih Lee, David Kriegman</i>	
Face Recognition with Local Binary Patterns	469
<i>Timo Ahonen, Abdenour Hadid, Matti Pietikäinen</i>	
Steering in Scale Space to Optimally Detect Image Structures	482
<i>Jeffrey Ng, Anil A. Bharath</i>	
Hand Motion from 3D Point Trajectories and a Smooth Surface Model	495
<i>Guillaume Dewaele, Frédéric Devernay, Radu Horaud</i>	

A Robust Probabilistic Estimation Framework for Parametric Image Models	508
<i>Maneesh Singh, Himanshu Arora, Narendra Ahuja</i>	
Keyframe Selection for Camera Motion and Structure Estimation from Multiple Views.....	523
<i>Thorsten Thormählen, Hellward Broszio, Axel Weissenfeld</i>	
Omnidirectional Vision: Unified Model Using Conformal Geometry	536
<i>Eduardo Bayro-Corrochano, Carlos López-Franco</i>	
A Robust Algorithm for Characterizing Anisotropic Local Structures	549
<i>Kazunori Okada, Dorin Comaniciu, Navneet Dalal, Arun Krishnan</i>	
Dimensionality Reduction by Canonical Contextual Correlation Projections	562
<i>Marco Loog, Bram van Ginneken, Robert P.W. Duin</i>	
Illumination, Reflectance, and Reflection	
Accuracy of Spherical Harmonic Approximations for Images of Lambertian Objects under Far and Near Lighting	574
<i>Darya Frolova, Denis Simakov, Ronen Basri</i>	
Characterization of Human Faces under Illumination Variations Using Rank, Integrability, and Symmetry Constraints.....	588
<i>S. Kevin Zhou, Rama Chellappa, David W. Jacobs</i>	
User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior.....	602
<i>Anat Levin, Yair Weiss</i>	
The Quality of Catadioptric Imaging – Application to Omnidirectional Stereo	614
<i>Wolfgang Stürzl, Hansjürgen Dahmen, Hanspeter A. Mallot</i>	
Author Index	629

Table of Contents – Part II

Geometry

A Generic Concept for Camera Calibration	1
<i>Peter Sturm, Srikumar Ramalingam</i>	
General Linear Cameras	14
<i>Jingyi Yu, Leonard McMillan</i>	
A Framework for Pencil-of-Points Structure-from-Motion	28
<i>Adrien Bartoli, Mathieu Coquerelle, Peter Sturm</i>	
What Do Four Points in Two Calibrated Images Tell Us about the Epipoles?	41
<i>David Nistér, Frederik Schaffalitzky</i>	

Feature-Based Object Detection and Recognition II

Dynamic Visual Search Using Inner-Scene Similarity: Algorithms and Inherent Limitations	58
<i>Tamar Avraham, Michael Lindenbaum</i>	
Weak Hypotheses and Boosting for Generic Object Detection and Recognition	71
<i>A. Opelt, M. Fussenegger, A. Pinz, P. Auer</i>	
Object Level Grouping for Video Shots	85
<i>Josef Sivic, Frederik Schaffalitzky, Andrew Zisserman</i>	

Posters II

Statistical Symmetric Shape from Shading for 3D Structure Recovery of Faces	99
<i>Roman Dovgard, Ronen Basri</i>	
Region-Based Segmentation on Evolving Surfaces with Application to 3D Reconstruction of Shape and Piecewise Constant Radiance	114
<i>Hailin Jin, Anthony J. Yezzi, Stefano Soatto</i>	
Human Upper Body Pose Estimation in Static Images	126
<i>Mun Wai Lee, Isaac Cohen</i>	
Automated Optic Disc Localization and Contour Detection Using Ellipse Fitting and Wavelet Transform	139
<i>P.M.D.S. Pallawala, Wynne Hsu, Mong Li Lee, Kah-Guan Au Eong</i>	

View-Invariant Recognition Using Corresponding Object Fragments	152
<i>Evgeniy Bart, Evgeny Byvatov, Shimon Ullman</i>	
Variational Pairing of Image Segmentation and Blind Restoration	166
<i>Leah Bar, Nir Sochen, Nahum Kiryati</i>	
Towards Intelligent Mission Profiles of Micro Air Vehicles:	
Multiscale Viterbi Classification	178
<i>Sinisa Todorovic, Michael C. Nechyba</i>	
Stitching and Reconstruction of Linear-Pushbroom Panoramic Images	
for Planar Scenes	190
<i>Chu-Song Chen, Yu-Ting Chen, Fay Huang</i>	
Audio-Video Integration for Background Modelling	202
<i>Marco Cristani, Manuele Bicego, Vittorio Murino</i>	
A Combined PDE and Texture Synthesis Approach to Inpainting	214
<i>Harald Grossauer</i>	
Face Recognition from Facial Surface Metric	225
<i>Alexander M. Bronstein, Michael M. Bronstein, Alon Spira, Ron Kimmel</i>	
Image and Video Segmentation by Anisotropic Kernel Mean Shift	238
<i>Jue Wang, Bo Thiesson, Yingqing Xu, Michael Cohen</i>	
Colour Texture Segmentation by Region-Boundary Cooperation	250
<i>Jordi Freixenet, Xavier Muñoz, Joan Martí, Xavier Lladó</i>	
Spectral Solution of Large-Scale Extrinsic Camera Calibration as	
a Graph Embedding Problem	262
<i>Matthew Brand, Matthew Antone, Seth Teller</i>	
Estimating Intrinsic Images from Image Sequences with	
Biased Illumination	274
<i>Yasuyuki Matsushita, Stephen Lin, Sing Bing Kang, Heung-Yeung Shum</i>	
Structure and Motion from Images of Smooth Textureless Objects	287
<i>Yasutaka Furukawa, Amit Sethi, Jean Ponce, David Kriegman</i>	
Automatic Non-rigid 3D Modeling from Video	299
<i>Lorenzo Torresani, Aaron Hertzmann</i>	
From a 2D Shape to a String Structure Using the Symmetry Set	313
<i>Arjan Kuijper, Ole Fogh Olsen, Peter Giblin, Philip Bille, Mads Nielsen</i>	

Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-directional Multi-camera System	326
<i>Tomokazu Sato, Sei Ikeda, Naokazu Yokoya</i>	
Evaluation of Robust Fitting Based Detection	341
<i>Sio-Song Ieng, Jean-Philippe Tarel, Pierre Charbonnier</i>	
Local Orientation Smoothness Prior for Vascular Segmentation of Angiography	353
<i>Wilbur C.K. Wong, Albert C.S. Chung, Simon C.H. Yu</i>	
Weighted Minimal Hypersurfaces and Their Applications in Computer Vision	366
<i>Bastian Goldlücke, Marcus Magnor</i>	
Interpolating Novel Views from Image Sequences by Probabilistic Depth Carving	379
<i>Annie Yao, Andrew Calway</i>	
Sparse Finite Elements for Geodesic Contours with Level-Sets	391
<i>Martin Weber, Andrew Blake, Roberto Cipolla</i>	
Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture	405
<i>Lorna Herda, Raquel Urtasun, Pascal Fua</i>	
Separating Specular, Diffuse, and Subsurface Scattering Reflectances from Photometric Images	419
<i>Tai-Pang Wu, Chi-Keung Tang</i>	
Temporal Factorization vs. Spatial Factorization	434
<i>Lihi Zelnik-Manor, Michal Irani</i>	
Tracking Aspects of the Foreground against the Background	446
<i>Hieu T. Nguyen, Arnold Smeulders</i>	
Example-Based Stereo with General BRDFs	457
<i>Adrien Treuille, Aaron Hertzmann, Steven M. Seitz</i>	
Adaptive Probabilistic Visual Tracking with Incremental Subspace Update	470
<i>David Ross, Jongwoo Lim, Ming-Hsuan Yang</i>	
On Refractive Optical Flow	483
<i>Sameer Agarwal, Satya P. Mallick, David Kriegman, Serge Belongie</i>	
Matching Tensors for Automatic Correspondence and Registration	495
<i>Ajmal S. Mian, Mohammed Bennamoun, Robyn Owens</i>	

A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks	506
<i>Iasonas Kokkinos, Rachid Deriche, Petros Maragos, Olivier Faugeras</i>	
Appearance Based Qualitative Image Description for Object Class Recognition	518
<i>Johan Thoreson, Stefan Carlsson</i>	
Consistency Conditions on the Medial Axis	530
<i>Anthony Pollitt, Peter Giblin, Benjamin Kimia</i>	
Normalized Cross-Correlation for Spherical Images	542
<i>Lorenzo Sorgi, Kostas Daniilidis</i>	
Bias in the Localization of Curved Edges	554
<i>Paulo R.S. Mendonça, Dirk Padfield, James Miller, Matt Turek</i>	
Texture	
Texture Boundary Detection for Real-Time Tracking	566
<i>Ali Shahrokni, Tom Drummond, Pascal Fua</i>	
A TV Flow Based Local Scale Measure for Texture Discrimination	578
<i>Thomas Brox, Joachim Weickert</i>	
Spatially Homogeneous Dynamic Textures	591
<i>Gianfranco Doretto, Eagle Jones, Stefano Soatto</i>	
Synthesizing Dynamic Texture with Closed-Loop Linear Dynamic System	603
<i>Lu Yuan, Fang Wen, Ce Liu, Heung-Yeung Shum</i>	
Author Index	617

Table of Contents – Part IV

Scale Space, Flow, Restoration

A l^1 -Unified Variational Framework for Image Restoration	1
<i>Julien Bect, Laure Blanc-Féraud, Gilles Aubert, Antonin Chambolle</i>	
Support Blob Machines. The Sparsification of Linear Scale Space	14
<i>Marco Loog</i>	
High Accuracy Optical Flow Estimation Based on a Theory for Warping	25
<i>Thomas Brox, Andrés Bruhn, Nils Papenberg, Joachim Weickert</i>	

Model-Based Approach to Tomographic Reconstruction Including Projection Deblurring. Sensitivity of Parameter Model to Noise on Data	37
<i>Jean Michel Lagrange, Isabelle Abraham</i>	

2D Shape Detection and Recognition

Unlevel-Sets: Geometry and Prior-Based Segmentation.....	50
<i>Tammy Riklin-Raviv, Nahum Kiryati, Nir Sochen</i>	

Learning and Bayesian Shape Extraction for Object Recognition	62
<i>Washington Mio, Anuj Srivastava, Xiuwen Liu</i>	

Multiphase Dynamic Labeling for Variational Recognition-Driven Image Segmentation	74
<i>Daniel Cremers, Nir Sochen, Christoph Schnörr</i>	

Posters IV

Integral Invariant Signatures	87
<i>Siddharth Manay, Byung-Woo Hong, Anthony J. Yezzi, Stefano Soatto</i>	

Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes	100
<i>Bill Triggs</i>	

Spectral Simplification of Graphs	114
<i>Huaijun Qiu, Edwin R. Hancock</i>	

Inferring White Matter Geometry from Diffusion Tensor MRI: Application to Connectivity Mapping	127
<i>Christophe Lenglet, Rachid Deriche, Olivier Faugeras</i>	

Unifying Approaches and Removing Unrealistic Assumptions in Shape from Shading: Mathematics Can Help	141
<i>Emmanuel Prados, Olivier Faugeras</i>	
Morphological Operations on Matrix-Valued Images	155
<i>Bernhard Burgeth, Martin Welk, Christian Feddern, Joachim Weickert</i>	
Constraints on Coplanar Moving Points	168
<i>Sujit Kuthirummal, C.V. Jawahar, P.J. Narayanan</i>	
A PDE Solution of Brownian Warping	180
<i>Mads Nielsen, P. Johansen</i>	
Stereovision-Based Head Tracking Using Color and Ellipse Fitting in a Particle Filter	192
<i>Bogdan Kwolek</i>	
Parallel Variational Motion Estimation by Domain Decomposition and Cluster Computing	205
<i>Timo Kohlberger, Christoph Schnörr, Andrés Bruhn, Joachim Weickert</i>	
Whitening for Photometric Comparison of Smooth Surfaces under Varying Illumination	217
<i>Margarita Osadchy, Michael Lindenbaum, David Jacobs</i>	
Structure from Motion of Parallel Lines	229
<i>Patrick Baker, Yiannis Aloimonos</i>	
A Bayesian Framework for Multi-cue 3D Object Tracking	241
<i>Jan Giebel, Darin M. Gavrila, Christoph Schnörr</i>	
On the Significance of Real-World Conditions for Material Classification	253
<i>Eric Hayman, Barbara Caputo, Mario Fritz, Jan-Olof Eklundh</i>	
Toward Accurate Segmentation of the LV Myocardium and Chamber for Volumes Estimation in Gated SPECT Sequences	267
<i>Diane Lingrand, Arnaud Charnoz, Pierre Malick Koulibaly, Jacques Darcourt, Johan Montagnat</i>	
An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets	279
<i>Zia Khan, Tucker Balch, Frank Dellaert</i>	
Human Pose Estimation Using Learnt Probabilistic Region Similarities and Partial Configurations	291
<i>Timothy J. Roberts, Stephen J. McKenna, Ian W. Ricketts</i>	

Tensor Field Segmentation Using Region Based Active Contour Model	304
<i>Zhizhou Wang, Baba C. Vemuri</i>	
Groupwise Diffeomorphic Non-rigid Registration for Automatic Model Building	316
<i>T.F. Cootes, S. Marsland, C.J. Twining, K. Smith, C.J. Taylor</i>	
Separating Transparent Layers through Layer Information Exchange	328
<i>Bernard Sarel, Michal Irani</i>	
Multiple Classifier System Approach to Model Pruning in Object Recognition	342
<i>Josef Kittler, Ali R. Ahmadifard</i>	
Coaxial Omnidirectional Stereopsis	354
<i>Libor Spacek</i>	
Classifying Materials from Their Reflectance Properties	366
<i>Peter Nillius, Jan-Olof Eklundh</i>	
Seamless Image Stitching in the Gradient Domain	377
<i>Anat Levin, Assaf Zomet, Shmuel Peleg, Yair Weiss</i>	
Spectral Clustering for Robust Motion Segmentation	390
<i>JinHyeong Park, Hongyuan Zha, Rangachar Kasturi</i>	
Learning Outdoor Color Classification from Just One Training Image	402
<i>Roberto Manduchi</i>	
A Polynomial-Time Metric for Attributed Trees	414
<i>Andrea Torsello, Džena Hidović, Marcello Pelillo</i>	
Probabilistic Multi-view Correspondence in a Distributed Setting with No Central Server	428
<i>Shai Avidan, Yael Moses, Yoram Moses</i>	
Monocular 3D Reconstruction of Human Motion in Long Action Sequences	442
<i>Gareth Loy, Martin Eriksson, Josephine Sullivan, Stefan Carlsson</i>	
Fusion of Infrared and Visible Images for Face Recognition	456
<i>Aglika Gyaourova, George Bebis, Ioannis Pavlidis</i>	
Reliable Fiducial Detection in Natural Scenes	469
<i>David Claus, Andrew W. Fitzgibbon</i>	
Light Field Appearance Manifolds	481
<i>Chris Mario Christoudias, Louis-Philippe Morency, Trevor Darrell</i>	

Galilean Differential Geometry of Moving Images	494
<i>Daniel Fagerström</i>	
Tracking People with a Sparse Network of Bearing Sensors	507
<i>A. Rahimi, B. Dunagan, T. Darrell</i>	
Transformation-Invariant Embedding for Image Analysis	519
<i>Ali Ghodsi, Jiayuan Huang, Dale Schuurmans</i>	
The Least-Squares Error for Structure from Infinitesimal Motion	531
<i>John Oliensis</i>	
Stereo Based 3D Tracking and Scene Learning, Employing Particle Filtering within EM	546
<i>Trausti Kristjansson, Hagai Attias, John Hershey</i>	
3D Shape Representation and Reconstruction	
The Isophotic Metric and Its Application to Feature Sensitive Morphology on Surfaces	560
<i>Helmut Pottmann, Tibor Steiner, Michael Hofer, Christoph Haider, Allan Hanbury</i>	
A Closed-Form Solution to Non-rigid Shape and Motion Recovery	573
<i>Jing Xiao, Jin-xiang Chai, Takeo Kanade</i>	
Stereo Using Monocular Cues within the Tensor Voting Framework	588
<i>Philippos Mordohai, Gérard Medioni</i>	
Shape and View Independent Reflectance Map from Multiple Views	602
<i>Tianli Yu, Ning Xu, Narendra Ahuja</i>	
Author Index	
	617

A Constrained Semi-supervised Learning Approach to Data Association

Hendrik Kück, Peter Carbonetto, and Nando de Freitas

Dept. of Computer Science
University of British Columbia
Vancouver, Canada
`{kueck, pcarbo, nando}@cs.ubc.ca`

Abstract. Data association (obtaining correspondences) is a ubiquitous problem in computer vision. It appears when matching image features across multiple images, matching image features to object recognition models and matching image features to semantic concepts. In this paper, we show how a wide class of data association tasks arising in computer vision can be interpreted as a constrained semi-supervised learning problem. This interpretation opens up room for the development of new, more efficient data association methods. In particular, it leads to the formulation of a new principled probabilistic model for constrained semi-supervised learning that accounts for uncertainty in the parameters and missing data. By adopting an ingenious data augmentation strategy, it becomes possible to develop an efficient MCMC algorithm where the high-dimensional variables in the model can be sampled efficiently and directly from their posterior distributions. We demonstrate the new model and algorithm on synthetic data and the complex problem of matching image features to words in the image captions.

1 Introduction

Data association is an ubiquitous problem in computer vision. It manifests itself when matching images (*eg* stereo and motion data [1]), matching image features to object recognition models [2] and matching image features to language descriptions [3]. The data association task is commonly mapped to an unsupervised probabilistic mixture model [4,1,5]. The parameters of this model are typically learned with the EM algorithm or approximate variants. This approach is fraught with difficulties. EM often gets stuck in local minima and is highly dependent on the initial values of the parameters. Markov chain Monte Carlo (MCMC) methods also perform poorly in this mixture model scenario [6]. The reason for this failure is that the number of modes in the posterior distribution of the parameters is factorial in the number of mixture components [7]. Maximisation in such a highly peaked space is a formidable task and likely to fail in high dimensions. This is unfortunate as it is becoming clear that effective learning techniques for computer vision have to manage many mixture components and high dimensions. Here, we take a new route to solve this vision problem. We cast the data association problem as one of constrained semi-supervised learning. We argue that

it is possible to construct efficient MCMC algorithms in this new setting. Efficiency here is a result of using a data augmentation method, first introduced in econometrics by economics Nobel laureate Daniel McFadden [8], which enables us to compute the distribution of the high-dimensional variables analytically. That is, instead of sampling in high-dimensions with a Markov chain, we sample directly from the posterior distribution of the high-dimensional variables. This, so called *Rao-Blackwellised*, sampler achieves an important decrease in variance as predicted by well known theorems from Markov chain theory [9].

Our approach is similar in spirit to the multiple instance learning paradigm of Dietterich *et al* [10]. This approach is expanded in [11] where the authors adopt support vector machines to deal with the supervised part of the model and integer programming constraints to handle the missing labels. This optimisation approach suffers from two problems. First, it is NP-hard so one has to introduce heuristics. Second, it is an optimisation technique and as such it only gives us a point estimate of the decision boundary. That is, it lacks a probabilistic interpretation. The approach we propose here allows us to compute all probabilities of interest and consequently we are able to obtain not only point estimates, but also confidence measures. These measures are essential when the data association mechanism is embedded in a meta decision problem, as is often the case.

The problem of semi-supervised learning has received great attention in the recent machine learning literature. In particular, very efficient kernel methods have been proposed to attack this problem [12,13]. Our approach, still based on kernel expansions, favours sparse solutions. Moreover, it does not require supervised samples from each category and, in addition, it is probabilistic. The most important point is that our approach allows for the introduction of constraints. Adding constraints to existing algorithms for semi-supervised learning leads to NP-hard problems, typically of the integer programming type as in [11].

We introduce a coherent, fully probabilistic Bayesian model for constrained semi-supervised learning. This enables us to account for uncertainty in both the parameters and unknown labels in a principled manner. The model applies to both regression and classification, but we focus on the problem of binary classification so as to demonstrate the method in the difficult task of matching image regions to words in the image caption [3].

Our contribution is therefore threefold: a new approach to a known complex data association (correspondence) problem, a general principled probabilistic model for constrained semi-supervised learning and a sophisticated blocked MCMC algorithm to carry out the necessary computations.

2 Data Association as Constrained Semi-supervised Learning

There are many large collections of annotated images on the web, galleries and news agencies. Figure 1 shows a few annotated images from the Corel image database. By, for example, segmenting the images, we can view object recogni-

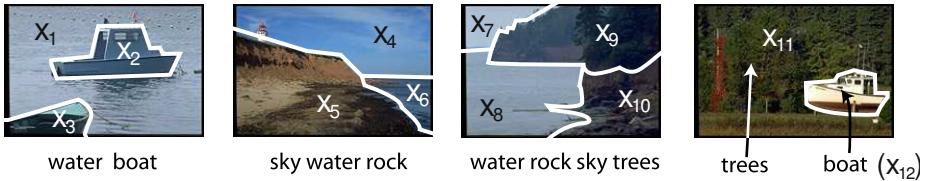


Fig. 1. Annotated images from the Corel database. We would like to automatically match image regions to words in the caption. That is we don't know the right associations (correspondences) between image features and text features.

tion as the process of finding the correct associations between the labels in the caption and the image segments. Knowing the associations allows us to build a translation model that takes as input image features and outputs the appropriate words; see [3] for a detailed description. A properly trained translation model takes images (without any captions) as input and outputs images with labelled regions.

What makes this approach feasible is that the training set of images like the leftmost three images in Figure 1 is vast and ever increasing. On the other hand, a supervised approach using training data like the right-most image, where segments have been annotated, is very problematic in practice, as labelling individual segments (or other local image features) is hard and time-consuming.

This data association problem can be formulated as a mixture model similar to the ones used in statistical machine translation. This is the approach originally proposed in [3] and extended in [14] to handle continuous image features. The parameters in both cases were learned with EM. The problem with this approach is that the posterior over parameters of the mixture model has a factorial number of modes and so EM tends to get stuck in local minima. The situation is no better for MCMC algorithms for mixture models because of this factorial explosion of modes [6]. This calls for a new approach.

We can convert the data association problem to a constrained semi-supervised learning problem. We demonstrate this with the toy example of Figure 1. Suppose we are interested in being able to detect boats in images. We could assume that if the word *boat* does not appear in the caption, then there are no boats in the image¹. In this case, we assign the label 0 to each segment in the image. If however the word *boat* appears in the caption, then we know that at least one of the segments corresponds to a boat. The problem is that we do not know which. So we assign question marks to the labels in this image. Sometimes, we might be fortunate and have a few segment labels as in the rightmost image of Figure 1.

By letting x_i denote the feature vector corresponding to the i -th segment and y_i denote the existence of a boat, our data association problem is mapped to the following semi-supervised binary classification task

¹ Of course, this depends on how good the labels are, but as mentioned earlier, there are many databases with very good captions; see for example www.corbis.com. So for now we work under this assumption.

	image 1	image 2	image 3	image 4
Input \mathbf{x}	$x_1 \ x_2 \ x_3$	$x_4 \ x_5 \ x_6$	$x_7 \ x_8 \ x_9 \ x_{10}$	$x_{11} \ x_{12}$
Labels \mathbf{y}	?	?	0 0 0	0 0 0 0

Note that for the question marks, we have the constraint that at least one of them has to be a 1 (this is what leads to the integer programming problem in optimisation approaches). To be able to annotate all the image segments, we need to build one classifier for each word of interest. This is sound from an information retrieval point of view [11]. From an object recognition perspective, we would like to adopt mult categorial classifiers. Here, we opt for a simple solution by combining the responses of the various binary classifiers [15].

In more precise terms, given the training data \mathcal{D} (a collection of images with captions) the goal is then to learn the predictive distribution $p(y = 1|x)$, where y is a binary indicator variable that is 1 iff the new test-set image segment represented by x is part of the concept. If we use a model with parameters θ , the Bayesian solution is given by

$$p(y = 1|x) = \int p(y = 1|x, \theta)p(\theta|\mathcal{D})d\theta.$$

That is, we integrate out the uncertainty of the parameters. The problem with this theoretical solution is that the integral is intractable. To overcome this problem, we sample θ according to $p(\theta|\mathcal{D})$ to obtain the following approximation

$$p(y = 1|x) \approx \frac{1}{N} \sum_i p(y = 1|x, \theta_i)$$

where θ_i is one of the samples. This approximation converges to the true solution by the Strong Law of Large Numbers. This approach not only allows us to compute point estimates, but also confidence intervals. In the next section, we outline the probabilistic model.

3 Parametrization and Probabilistic Model

Our training data \mathcal{D} consists of two parts, the set of blob description vectors $\{x_{1:N}\}$ with $x_i \in \mathbb{R}^d$ for $i = 1, \dots, N$ and a set of binary labels y^k . The full set of labels includes the known and unknown labels, $y \triangleq \{y^k, y^u\}$. Our classification model is as follows

$$\Pr(y_i = 1|x_i, \beta, \gamma) = \Phi(f(x_i, \beta, \gamma)), \quad (1)$$

where $\Phi(u) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^u \exp(-a^2/2) da$ is the cumulative function of the standard Normal distribution. This is the so-called probit link. By convention, researchers tend to adopt the logistic link function $\varphi(u) = (1 + \exp(-u))^{-1}$. However, from a Bayesian computational point of view, the probit link has many

advantages and is equally valid. Following Tam, Doucet and Kotagiri[16], the unknown function is represented with a sparse kernel machine with kernels centered at the data points $x_{1:N}$:

$$f(x, \beta, \gamma) = \beta_0 + \sum_{i=1}^N \gamma_i \beta_i K(x, x_i). \quad (2)$$

Here β is a N -dimensional parameter vector and K is a kernel function. Typical choices for the kernel function K are:

- Linear: $K(x_i, x) = \|x_i - x\|$
- Cubic: $K(x_i, x) = \|x_i - x\|^3$
- Gaussian: $K(x_i, x) = \exp(-\lambda \|x_i - x\|^2)$
- Sigmoidal: $K(x_i, x) = \tanh(\lambda \|x_i - x\|^2)$

The last two kernels require a scale parameter λ to be chosen. The vector of unknown binary indicator variables $\gamma \in \{0, 1\}^N$ is used to control the complexity of the model. It leads to sparser solutions and updates, where the subset of active kernels adapts to the data. This is a well studied statistical model [16].

When all the kernels are active, we can express equation (2) in matrix notation

$$f(x_i, \beta) = \Psi_i^T \beta,$$

where Ψ_i denotes the i -th row of the kernel matrix

$$\Psi = \begin{bmatrix} 1 & K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_N) \\ 1 & K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_N, x_1) & K(x_N, x_2) & \cdots & K(x_N, x_N) \end{bmatrix} \quad (3)$$

When only a subset of kernels is active, we obtain a sparse model:

$$f(x_i, \beta_\gamma) = \Psi_{\gamma i}^T \beta_\gamma,$$

where Ψ_γ is the matrix consisting of the columns j of Ψ where $\gamma_j = 1$. $\Psi_{\gamma i}$ then is the i -th row of this matrix. β_γ is the reduced version of β , only containing the coefficients for the activated kernels. In [16], this model is applied to supervised learning and shown to produce more accurate results than support vector machines and other kernel machines. Here, we need to extend the model to the more general scenario of semi-supervised learning with constraints in the labels.

We adopt a hierarchical Bayesian model [17]. We assume that each kernel is active with probability τ , i.e. $p(\gamma | \tau)$ is a Bernoulli distribution. Instead of having the user choose a fixed τ a priori, we deal with this parameter in the Bayesian way and assign a prior $p(\tau)$ to it. This way, the value of τ is allowed to adapt to the data. At the same time we can bias it by specifying the prior $p(\tau)$ according to our prior belief as to what the value of τ should be. While

Tam, Doucet and Kotagiri [16] use the completely uninformative uniform prior, we instead choose to put a conjugate Beta-prior on τ which allows the user to exert as much control as desired over the percentage of active kernels

$$p(\tau) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\tau^{a-1}(1-\tau)^{b-1}. \quad (4)$$

For the choice $a = b = 1.0$, we get the uninformative uniform distribution. We obtain the prior on the binary vector γ by integrating over τ

$$p(\gamma) = \int p(\gamma|\tau)p(\tau)d\tau = \frac{\Gamma(\Sigma\gamma+a)\Gamma(N-\Sigma\gamma+b)}{\Gamma(N+a+b)}, \quad (5)$$

where $\Sigma\gamma$ is the number of active kernels, i.e. the number of non zero elements in γ .

A (maximum entropy) g-prior is placed on the coefficients β :

$$p(\beta) = \mathcal{N}(0, \delta^2(\Psi_\gamma^T\Psi_\gamma)^{-1}) \quad (6)$$

where the regularisation parameter is assigned an inverse gamma prior:

$$p(\delta^2) = \text{IG}\left(\frac{\mu}{2}, \frac{\nu}{2}\right). \quad (7)$$

This prior has two parameters μ and ν that have to be specified by the user. One could argue that this is worse than the single parameter δ^2 . However, the parameters of this hyper-prior have a much less direct influence than δ^2 itself and are therefore less critical for the performance of the algorithms [17]. Assigning small values to these parameters results in an uninformative prior and allows δ^2 to adapt to the data.

3.1 Augmented Model

We augment the probabilistic model artificially in order to obtain an analytical expression for the posterior of the high-dimensional variables β . In particular, we introduce the set of independent variables $z_i \in \mathbb{R}$, such that

$$z_i = f(x_i, \beta, \gamma) + n_t, \quad (8)$$

where $n_t \stackrel{i.i.d.}{\sim} \mathcal{N}(0, 1)$. The set of augmentation variables consists of two subsets $z \triangleq \{z^k, z^u\}$, one corresponding to the known labels y^k and the other to the unknown labels y^u . For the labelled data, we have

$$p(z_i^k | \beta, \gamma, x_i) = \mathcal{N}(f(x_i, \beta, \gamma), 1) = \mathcal{N}(\Psi_{\gamma_i}^T \beta, 1). \quad (9)$$

We furthermore define

$$y_i^k = \begin{cases} 1 & \text{if } z_i^k > 0, \\ 0 & \text{otherwise.} \end{cases}$$

It is then easy to check that one has the required result:

$$\Pr(y_i^k = 1 | x_i, \beta_\gamma, \gamma) = \Pr(z_i^k \geq 0 | x_i, \beta_\gamma) = \Pr(n_i \geq -\Psi_{\gamma i}^T \beta_\gamma) = \Phi(\Psi_{\gamma i}^T \beta_\gamma).$$

Now, let $y_{k:k+l}^u$ denote the set of missing labels for a particular image (a set of question marks as described in Section 2). The prior distribution for the corresponding augmentation variables $z_{k:k+l}^u$ is then:

$$p(z_{k:k+l}^u | \beta, \gamma, x_i) \propto \left[\prod_{j=k}^{j=k+l} \mathcal{N}(\Psi_{\gamma j}^T \beta_\gamma, 1) \right] \mathbb{I}_{\mathcal{C}}(z_{k:k+l}^u) \quad (10)$$

where $\mathbb{I}_\Omega(\omega)$ is the set indicator function: 1 if $\omega \in \Omega$ and 0 otherwise. Our particular set of constraints is $\mathcal{C} \triangleq \{\text{one or more } z_j^u > 0\}$. That is, one or more of the z_j^u must be positive so that at least one of the y^u are positive. This prior is a truncated Normal distribution with the negative octant missing. The hierarchical Bayesian model is summarised in Figure 2.

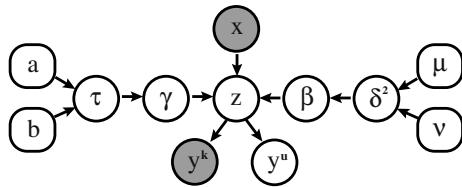


Fig. 2. Our directed acyclic graphical model. Note that by conditioning on z , y is independent of the model parameters.

3.2 Posterior Distribution

The posterior distribution follows from Bayes rule

$$p(\beta, \gamma, \delta^2, z | y^k, x_{1:N}) \propto p(y^k | z^k) p(\gamma) p(\beta | \delta^2) p(\delta^2) p(z^u | \beta, \gamma, x) p(z^k | \beta, \gamma, x)$$

The key thing to note, by looking at our graphical model, is that by conditioning on the 1-dimensional variables z , the model reduces to a standard linear-Gaussian model [17]. We can as a result obtain analytical expressions for the conditional posteriors of the high-dimensional variables β and the regularisation parameter δ

$$p(\beta | z, x, \gamma, \delta^2) = \mathcal{N}\left(\frac{\delta^2}{1 + \delta^2} (\Psi_\gamma^T \Psi_\gamma)^{-1} \Psi_\gamma^T z, \frac{\delta^2}{1 + \delta^2} (\Psi_\gamma^T \Psi_\gamma)^{-1}\right) \quad (11)$$

$$p(\delta^2 | z, \beta, \gamma) = \mathcal{IG}\left(\frac{\mu + \Sigma_\gamma + 1}{2}, \frac{\nu + \beta^T \Psi_\gamma^T \Psi_\gamma \beta}{2}\right) \quad (12)$$

where z is the vector $(z_1, z_2, \dots, z_N)^T$. The posterior distribution of the augmentation variables z^k is given by the following truncated Normal distributions:

$$p(z_i^k | \beta, \gamma, x_i, y_i^k) \propto p(y_i^k | z_i^k) p(z_i^k | x_i, \beta, \gamma) = \begin{cases} \mathcal{N}(\Psi_{\gamma i}^T \beta, 1) \mathbb{I}_{(0, +\infty)}(z_i^k) & \text{if } y_i^k = 1 \\ \mathcal{N}(\Psi_{\gamma i}^T \beta, 1) \mathbb{I}_{(-\infty, 0]}(z_i^k) & \text{if } y_i^k = 0 \end{cases} \quad (13)$$

4 MCMC Computation

We need to sample from the posterior distribution $p(\theta | \mathcal{D})$, where θ represents the full set of parameters. To accomplish this, we introduce a Metropolised blocked Gibbs sampler. In short, we sample the high-dimensional parameters β and the regularisation parameters directly from their posterior distributions (equations (11) and (12)). It is important to note that only the components of β associated with the active kernels need to be updated. This computation is therefore very efficient. The γ are sampled with the efficient MCMC algorithm described in detail in [16]. The z^u are sampled from the truncated multivariate Gaussian in equation (10), while the z^k are sampled from the truncated distributions given by equation (13).

To sample from the truncated Gaussian distributions, we use the specialised routines described in [18]. These routines based on results from large deviation theory are essential in order to achieve good acceptance rates. We found in our experiments that the acceptance rate was satisfactory (70% to 80%).

5 Experiments

5.1 Synthetic Data

In this first experiment we tested the performance of our algorithm on synthetic data. We sampled 300 data points from a mixture model consisting of a Gaussian and a surrounding ring with Gaussian cross section (see Figure 3(a)). Data points generated by the inner Gaussian were taken to be the positive instances, while those on the ring were assumed to be negative. The data points were then randomly grouped into groups (representing documents) of 6 data points each. In the given example, this resulted in 12 groups with exclusively negative data points, and 38 groups with both positives and negative instances. This corresponds to 72 data points with known negative labels and 228 data points with unknown but constrained labels.

We ran our algorithm on this data for 2000 samples (after a burn-in period of 1000 samples) using uninformative priors and a sigmoidal kernel with kernel parameter $\lambda = 1.0$. Although no data points were explicitly known to be positive in this case, the information of the constraints was sufficient to learn a nice distribution $p(y = 1 | x)$ as shown in Figure 3(b). Using an appropriate threshold produces a perfect classification in this example.

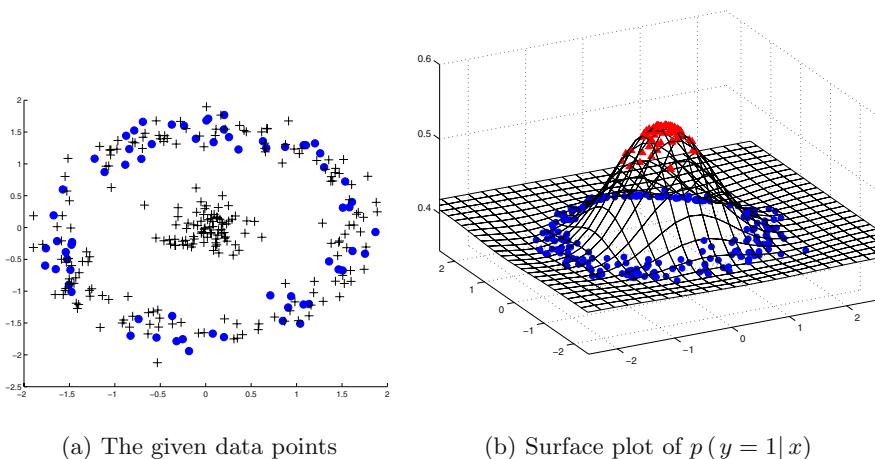


Fig. 3. Experiment with synthetic data. (a) shows the generated data points. Instances with known negative labels are shown as filled circles whereas data points with unknown label are represented by the + symbol. The plot in (b) visualizes the probability distribution computed by our approach. The distribution obviously nicely separates positive and negative examples and thus provides an excellent classifier.

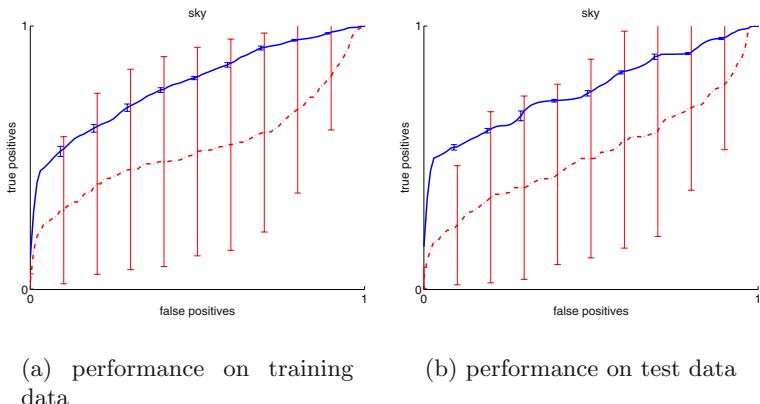


Fig. 4. ROC plots (as in Figure 5) for the annotation 'sky'. The average performance of our proposed approach is visualized by the solid line, that of the EM mixture algorithm by the dashed line. The error bars represent the standard deviation across 20 runs. It is clear from the plots that our proposed algorithm is more reliable and stable.

5.2 Object Recognition Data Set

For this experiment, we used a set of 300 annotated images from the Corel database. The images in this set were annotated with in total 38 different words and each image was segmented into regions using normalised cuts [19]. Each of the regions is described by a 6-dimensional feature vector (CIE-Lab colour, y position in the image, boundary to area ratio and standard deviation of brightness). The data set was split into one training set containing 200 images with 2070 image regions and a test set of 100 images with 998 regions.

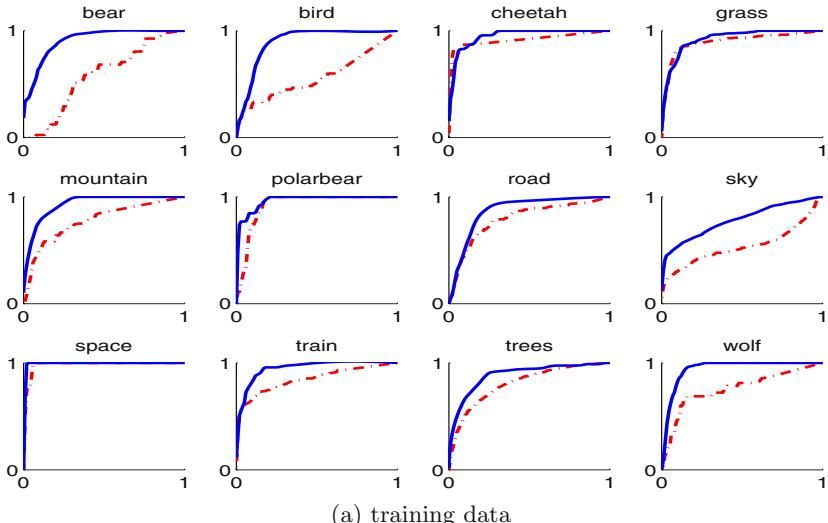
We compared two learning methods in this experiment. The first consisted of a mixture of Gaussians translation model trained with EM [3,14]. The second is the method proposed in this paper. We adopted a vague hyper-prior for δ^2 ($\mu = \nu = 0.01$). Experiments with different types of kernels showed the sigmoidal kernel to work best for this data set. Not only did it produce better classifiers than linear, multi-quadratic, cubic and Gaussian kernels, it also led to numerically more stable and sparser samplers. The average number of activated kernels per sample was between 5 and 20, depending on the learned concept.

We used both EM with the mixture model and our new constrained semi-supervised approach to learn binary classifiers for several of the words in this dataset. The Markov chains were run for 10,000 samples after a burn-in phase of 10,000 samples. On a 2.6 Ghz Pentium 4, run times for this were in the range of 5 to 10 minutes, which is perfectly acceptable in our setting.

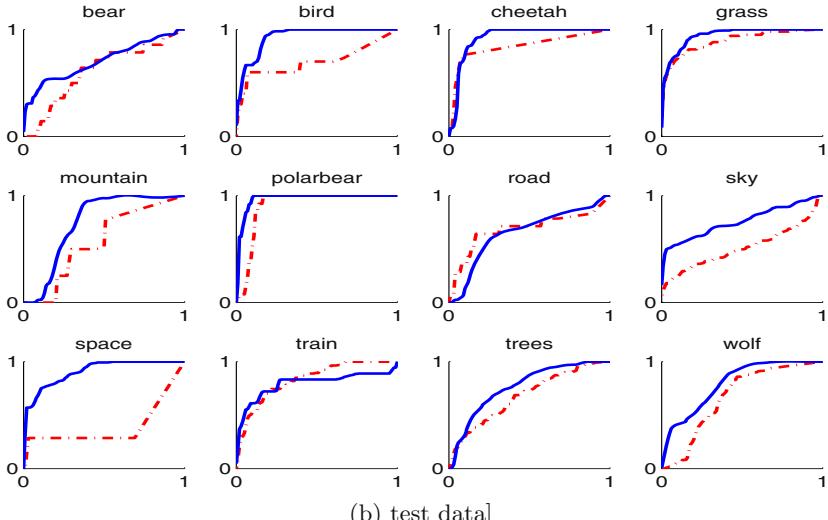
The performance of the learned classifiers was then evaluated by comparing their classification results for varying thresholds against a manual annotation of the individual image regions. The ROC plots in Figure 5 show the results averaged over 20 runs, plotting true positives against false positives. The plots show that the approach proposed in this paper yields significantly better classification performance than the EM mixture method. Given the relative simple features used and the small size of the data set, the performance is remarkably good. Figure 5.2 shows that the classifiers learned using the proposed approach generalize fairly well even where the EM mixture approach fails due to overfitting (look at the results for the concept 'space' for an example).

Figure 4 illustrates the dramatically higher consistency across runs of the algorithm proposed in this paper as compared to the EM algorithm for the mixture model. The error bars indicate the standard deviation of the ROC plots across the 20 runs. The large amount of variation indicates that the EM got stuck in local minima on several runs. While with the Corel data set this problem arose only for some of the categories, in larger and higher dimensional data sets, local minima are known to become a huge problem for EM.

Acknowledgements. We would like to thank Arnaud Doucet, Kerry Mengersen and Herbie Lee. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Institute for Robotics and Intelligent Systems (IRIS) under the project title 'Robot Partners: Visually Guided Multi-agent Systems'.



(a) training data



(b) test data]

Fig. 5. ROC plots measuring the classification performance on image regions from the Corel image dataset of both the proposed algorithm (solid line) and the EM mixture algorithm (dashed line), averaged over 20 runs. The x axis measures $\frac{\text{negatives falsely classified as positives}}{\text{actual negatives}}$ while the y axis corresponds to $\frac{\text{correctly classified positives}}{\text{actual positives}}$. The plots are generated by using the learned probabilistic classifiers with varying thresholds and allow to compare the classifiers independent of a chosen fixed threshold value. The performance on the test set is remarkable considering that the algorithm only has access to simple image features (and no text in any form).

References

1. Dellaert, F., Seitz, S., Thorpe, C., Thrun, S.: EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning* **50** (2003) 45–71
2. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: *IEEE Conference on Computer Vision and Pattern Recognition*. (2003)
3. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: *ECCV*. (2002) 97–112
4. Avitzour, D.: A maximum likelihood approach to data association. *IEEE Transactions on Aerospace and Electronic Systems* **28** (1992) 560–566
5. Blei, D., Jordan, M.: Modeling annotated data. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press (2003) 127–134
6. Celeux, G., Hurn, M., Robert, C.P.: Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association* **95** (2000) 957–970
7. Stephens, M.: Bayesian Methods for Mixtures of Normal Distributions. PhD thesis, Department of Statistics, Oxford University, England (1997)
8. McFadden, D.: A method of simulated moments for estimation of discrete response models without numerical integration. *Econometrica* **57** (1989) 995–1026
9. Liu, J., Wong, W.H., Kong, A.: Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika* **81** (1994) 27–40
10. Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple instance learning with axis-parallel rectangles. *Artificial Intelligence* **89** (1997) 31–71
11. Andrews, S., Tsachanidis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: *Advances in Neural Information Processing Systems 16*, Cambridge, MA, MIT Press (2003)
12. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: *International Conference on Machine Learning*. (2003)
13. Belkin, M., Niyogi, P.: Semi-supervised learning on manifolds. Technical Report TR-2002-12, Computer Science Department, The University of Chicago, MA (1994)
14. Carbonetto, P., de Freitas, N., Gustafson, P., Thompson, N.: Bayesian feature weighting for unsupervised learning, with application to object recognition. In: *AI-STATS*, Florida, USA (2003)
15. Tipping, M.E.: Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* **1** (2001) 211–244
16. Tham, S.S., Doucet, A., Ramamohanarao, K.: Sparse Bayesian learning for regression and classification using Markov chain Monte Carlo. In: *International Conference on Machine Learning*. (2002) 634–641
17. Bernardo, J.M., Smith, A.F.M.: *Bayesian Theory*. Wiley Series in Applied Probability and Statistics (1994)
18. Geweke, J.: Efficient simulation from the multivariate normal and Student t-distributions subject to linear constraints. In: *Proceedings of 23rd Symp. Interface*. (1991) 571–577
19. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. (1997) 731–737

Learning Mixtures of Weighted Tree-Unions by Minimizing Description Length

Andrea Torsello¹ and Edwin R. Hancock²

¹ Dipartimento di Informatica, Universita' Ca' Foscari di Venezia
via Torino 155, 30172 Venezia Mestre, Italy
torsello@dsi.unive.it

² Department of Computer Science, University of York
York YO10 5DD, England
erh@cs.york.ac.uk

Abstract. This paper focuses on how to perform the unsupervised clustering of tree structures in an information theoretic setting. We pose the problem of clustering as that of locating a series of archetypes that can be used to represent the variations in tree structure present in the training sample. The archetypes are tree-unions that are formed by merging sets of sample trees, and are attributed with probabilities that measure the node frequency or weight in the training sample. The approach is designed to operate when the correspondences between nodes are unknown and must be inferred as part of the learning process. We show how the tree merging process can be posed as the minimisation of an information theoretic minimum descriptor length criterion. We illustrate the utility of the resulting algorithm on the problem of classifying 2D shapes using a shock graph representation.

1 Introduction

Graph-based representations have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Concrete examples include the use of shock graphs to represent shape-skeletons [10,15], the use of trees to represent articulated objects [8,19] and the use of aspect graphs for 3D object representation [2]. The attractive feature of structural representations is that they concisely capture the relational arrangement of object primitives, in a manner which can be invariant to changes in object viewpoint. However, despite the many advantages and attractive features of graph representations, the methodology available for learning structural representations from sets of training examples is relatively limited. As a result, the process of constructing shape-spaces which capture the modes of structural variation for sets of graphs has proved to be elusive. Hence, geometric representations of shape such as point distribution models [6], have proved to be more amenable when variable sets of shapes must be analyzed. There are two reasons why pattern spaces are more easily constructed for curves and surfaces than for graphs. First, there is no canonical ordering for the nodes or edges of a graph. Hence,

before a vector-space can be constructed, then correspondences between nodes must be established. Second, structural variations in graphs manifest themselves as differences in the numbers of nodes and edges. As a result, even if a vector mapping can be established then the vectors will be of variable length.

One way of circumventing this problem is to embed the graphs in a low dimensional space using the distances between graphs or by using simple graph features that do not require correspondence analysis. For instance, Cyr and Kimia have used a geometric procedure to embed graphs on a view-sphere [1]. Demerici and Dickinson [9] have shown how the minimum distortion embedding procedure of Linial, London and Rabinovich [11] can be used for the purposes of correspondence matching. A recent review of methods that could be used to perform the embedding process is provided in the paper of Hjaltason and Samet [7]. However, although this work provides a means of capturing the distribution of graphs and can be used for clustering, it does not provide an embedding which allows a generative model of detailed graph structure to be learned. In other words, the distribution does not capture in an explicit manner the variations in the graphs in terms of changes in node and edge structure. Recently, though, there has been considerable interest in learning structural representations from samples of training data, in particular in the context of Bayesian networks [5,3], mixtures of tree-classifiers [12], or general relational models [4]. Unfortunately, these methods require the availability of node correspondences as a prerequisite.

The aim in this paper is to develop an information theoretic framework for the unsupervised learning of generative models of tree-structures from sets of examples. We pose the problem as that of learning a mixture of union trees. Each tree union is an archetype that represents a class of trees. Those trees that belong to a particular class can be obtained from the relevant tree archetype by node removal operations. Hence, the union-tree can be formed using a sequence of tree merge operations. We work under conditions in which the node correspondences required to perform merges are unknown and must be located by minimising tree edit distance. Associated with each node of the union structure is a probability. This is a random variable which represents the frequency of the node in the training sample. Since every tree in the sample can be obtained from one of the union structures in the mixture, the tree archetypes are generative models. There are three quantities that must be estimated to construct this generative model. The first of these are the correspondences between the nodes in the training examples and the estimated union structure. Secondly, there is the union structure itself. Finally, there are the node probabilities. We cast the estimation of these three quantities in an information theoretic setting using the description length for the union structure and its associated node probabilities given correspondences with the set of training examples [13]. With the tree-unions to hand, then we can apply use PCA to project the trees into a low dimensional vector space.

2 Generative Tree Model

Consider the set or sample of trees $\mathcal{D} = \{t_1, t_2, \dots, t_n\}$. Our aim in this paper is to cluster these trees, i.e. to perform unsupervised learning of the class structure of the sample. We pose this problem as that of learning a mixture of generative class archetypes. Each class archetype is constructed by merging sets of sample trees together to form a set of union structures. This merge process requires node correspondence information, and we work under conditions in which these are unknown and must be inferred as part of the learning process. Each tree in the sample can hence be obtained from one of the union-structures using a sequence of node removal operations. Thus the class archetypes are generative models since they capture in an explicit manner the structural variations for the sample trees belonging to a particular class in a probabilistic manner.

Suppose that the set of class archetypes constituting the mixture model is denoted by $\mathcal{H} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_k\}$. For the class c , the tree model \mathcal{T}_c is a structural archetype derived from the tree-union obtained by merging the set of trees $\mathcal{D}_c \subseteq \mathcal{D}$ constituting the class. Associated with the archetype is a probability distribution which captures the variations in tree structure within the class. Hence, the learning process involves estimating the union structure and the parameters of the associated probability distribution for the class model \mathcal{T}_c . As a prerequisite, we require the set of node correspondences \mathcal{C} between sample trees and the union tree for each class.

Our aim is to cast the learning process into an information theoretic setting. The estimation of the required class models is effected using a simple greedy optimization method. The quantity to be optimized is the descriptor length for the sample data-set \mathcal{D} . The parameters to be optimized include the structural archetype of the model \mathcal{T} as well as the node correspondences \mathcal{C} between the samples in the set \mathcal{D} and the archetype. Hence, the inter-sample node correspondences are not assumed to be known *a priori*. Since the correspondences are uncertain, we must solve two interdependent optimization problems. These are the optimization of the union structure given a set of correspondences, and the optimization of the correspondences given the tree structure. These dual optimization steps are approximated by greedily merging similar tree-models.

We characterize uncertainties in the structure obtained by tree merge operations by assigning probabilities to nodes. By adopting an information theoretic approach we demonstrate that the tree-edit distance, and hence the costs for the edit operations used to merge trees, are related to the entropies associated with the node probabilities.

2.1 Probabilistic Framework

More formally, the basis of the proposed structural learning approach is a generative tree model which allows us to assign a probability distribution to a sample of hierarchical trees. Each hierarchical tree t is defined by a set of nodes \mathcal{N}^t , a tree-order relation $\mathcal{O}^t \subset \mathcal{N}^t \times \mathcal{N}^t$ between the nodes, and, in the case of weighted trees, a weight set $W^t = \{w_i^t | i \in \mathcal{N}^t\}$ where w_i^t is the weight associated with

node i of tree t . A tree-order relation \mathcal{O}^t is an order relation with the added constraint that if $(x, y) \in \mathcal{O}^t$ and $(z, y) \in \mathcal{O}^t$, then either $(x, z) \in \mathcal{O}^t$ or $(z, x) \in \mathcal{O}^t$. A node b is said to be a *descendent* of a , or $a \rightsquigarrow b$, if $(a, b) \in \mathcal{O}^t$. Furthermore, if b is a descendent of a then it is also a *child* of a if there is no node x such that $a \rightsquigarrow x$ and $x \rightsquigarrow b$, that is there is no node between a and b in the tree-order.

Our aim is to construct a generative model for a class of trees $\mathcal{D}_c \subset \mathcal{D}$. The structural component of this model \mathcal{T}_c consists of a set of nodes \mathcal{N}_c and an associated tree order relation $\mathcal{O}_c \subset \mathcal{N}_c \times \mathcal{N}_c$. Additionally, there is a set $\Theta_c = \{\theta_i^c, i \in \mathcal{N}_c\}$ of sampling probabilities θ_i^c for each node $i \in \mathcal{N}_c$. Hence the model is the triple $\mathcal{T}_c = (\mathcal{N}_c, \mathcal{O}_c, \Theta_c)$. A sample from this model is a hierarchical tree $t = (\mathcal{N}^t, \mathcal{O}^t)$ with node set $\mathcal{N}^t \subset \mathcal{N}_c$ and a node hierarchy \mathcal{O}^t that is the restriction to \mathcal{N}^t of \mathcal{O}_c . In other words, the sample tree is just a subtree of the class archetype, which can be obtained using a simple set of edit operations that prune the archetype.

To develop our generative model we make a number of simplifying assumptions. First, we drop the class index c to simplify notation. Second, we assume that the set of nodes for the union structure \mathcal{T} spans each of the encountered sample trees \mathcal{D} , i.e. $\mathcal{N} = \bigcup_{t \in \mathcal{D}} \mathcal{N}^t$. Third, we assume that the sampling error acts only on nodes, while the hierarchical relations are always sampled correctly. That is, if nodes i and j satisfy the relation $i \mathcal{O} j$, then node i will be an ancestor of node j in each tree-sample that has both nodes.

Our assumptions imply that two nodes will always satisfy the same hierarchical relation whenever they are both present in a sample tree. A consequence of this assumption is that the structure of a sample tree is completely determined by restricting the order relation of the model \mathcal{O} to the nodes observed in the sample tree. Hence, the links in the sampled tree can be viewed as the minimal representation of the order relation between the nodes. The sampling process is equivalent to the application of a set of node removal operations to the archetypical structure $\mathcal{T} = (\mathcal{N}, \mathcal{O}, \Theta)$, which makes the archetype a union of the set of all possible tree samples.

To define a probability distribution over the union structure \mathcal{T} , we require the correspondences between the nodes in each sample tree t and the nodes in the class-model \mathcal{T} . We hence define a map $C : \mathcal{N}^t \rightarrow \mathcal{N}$ from the set \mathcal{N}^t of the nodes of t , to the nodes of the class model \mathcal{T} . The mapping induces a sample-correspondence for each node $i \in \mathcal{N}$. When the nodes of the sample trees have weights associated with them, then we would expect the sampling likelihood to reflect the distribution of weights. Hence, the simple probability distribution described above, which is based on uniform sample node probability, is not sufficient because it does not take into account the weight distribution. To overcome this shortcoming, in addition to the set of sampling probabilities Θ , we associate with the union model a weight distribution function. Here we assume that the weight distribution is a rectified Gaussian. For the node i of the union tree the weight probability distribution is given by

$$p(w_j | C(j) = i) \begin{cases} \frac{1}{\theta_i \sigma_i \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(w_j - \mu_i)^2}{\sigma_i^2}\right) & \text{if } w_j \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where the weight distribution has mode μ_i and standard deviation σ_i . The sampling probability is the integral of the distribution over positive weights, i.e.

$$\theta_i = \int_0^\infty \frac{\exp\left(-\frac{1}{2}\frac{(w-\mu_i)^2}{\sigma_i^2}\right)}{\sigma_i\sqrt{2\pi}} dw = 1 - \text{erfc}(\tau_i), \quad (1)$$

where $\tau_i = \mu_i/\sigma_i$ and erfc is the complementary error function. Taking into account the correspondences, the probability for node i induced by the mapping is

$$\phi(i|t, \mathcal{T}, \mathcal{C}) = \begin{cases} \theta_i p(w_j | C(j) = i) & \text{if there exists } j \in \mathcal{N}^t \text{ such that } \mathcal{C}(j) = i \\ 1 - \theta_i & \text{otherwise.} \end{cases}$$

2.2 Estimating Node Parameters

We can compute the log-likelihood of the sample data \mathcal{D} given the tree-union model \mathcal{T} and the correspondence mapping function \mathcal{C} . Under the assumption that the sampling process acts independently on the nodes of the structure the log-likelihood is

$$\mathcal{L}(\mathcal{D}|\mathcal{T}, \mathcal{C}) = \sum_{t \in \mathcal{D}} \sum_{i \in \mathcal{N}^t} \ln [\phi(i|t, \mathcal{T}, \mathcal{C})]$$

Our ultimate aim is to optimize the log-likelihood with respect to the correspondence map \mathcal{C} and the tree union model \mathcal{T} . These variables, though, are not independent since they both depend on the node-set \mathcal{N} . A variation in the actual identity and number of the nodes does not change the log-likelihood. Hence the dependency on the node-set can be lifted by simply assuming that the node set is the image of the correspondence map i.e. $\text{Im}(\mathcal{C})$. As we will see later, the reason for this is that those nodes that remain unmapped do not affect the maximization process.

We defer details of how we estimate the correspondence map \mathcal{C} and the order relation \mathcal{O} to later sections of the paper. However, assuming estimates of them are to hand, then we can make maximum likelihood estimates of the selected node model. That is, the set of sampling probabilities Θ in the unweighted case, and the node parameters $\bar{\tau}$ and $\bar{\sigma}$ in the weighted case.

To proceed, let $K_i = \{j \in \mathcal{N}^t | t \in \mathcal{D}, C(j) = i\}$ be the set of nodes in the different trees for which \mathcal{C} maps a node to i and let $p_i = |K_i|$ be the number of trees satisfying this condition. Further, let n_i be the number of trees in \mathcal{D} for which \mathcal{C} results in no mapping to the node i . Using the *weighted* node model, the log-likelihood function can be expressed as the sum of per-node log-likelihood functions

$$\mathcal{L}(\mathcal{D}|\mathcal{T}, \mathcal{C}) = \sum_{i \in \mathcal{N}} \log \left(\text{erfc}(\tau_i)^{n_i} (2\pi\sigma_i)^{-\frac{p_i}{2}} \exp \left[-\frac{1}{2} \sum_{j \in K_i} \left(\frac{w_j^t}{\sigma_i} - \tau_i \right)^2 \right] \right). \quad (2)$$

To estimate the parameters of the weight distribution, we take the derivatives of the log-likelihood function with respect to σ_i and τ_i and set them to zero. When $n_i > 0$, we maximize the log likelihood by setting $\tau_i^{(0)} = \text{erfc}^{-1}\left(\frac{n_i}{n_i + p_i}\right)$, and iterating the recurrence:

$$\sigma_i^{(k)} = -\frac{\tau_i^{(k)}}{2} \bar{W} + \sqrt{\left(\frac{\tau_i^{(k)}}{2} \bar{W}\right)^2 + \bar{W}^2} \quad \tau_i^{(k+1)} = \tau_i^{(k)} - \frac{f(\tau_i^{(k)}, \sigma_i^{(k)})}{\frac{d}{d\tau_i^{(k)}} f(\tau_i^{(k)}, \sigma_i^{(k)})} \quad (3)$$

where $\bar{W} = \sum_{j \in K_i} w_j^t$, $\bar{W}^2 = \sum_{j \in K_i} (w_j^t)^2$, and $f(\tau_i, \sigma_i) = n_i \text{erfc}'(\tau_i) + p_i \text{erfc}(\tau_i) \left(\frac{\bar{W}}{\sigma_i} - \tau_i\right)$.

3 Mixture Model

We now commence our discussion of how to estimate the order relation \mathcal{O} for the tree union \mathcal{T} , and the set of correspondences \mathcal{C} needed to merge the sample trees to form the tree-union. We pose the problem as that of fitting a mixture of tree unions to the set of sample trees. Each tree-union may be used to represent a distribution of trees that belong to a single class \mathcal{D}_c . The defining characteristic of the class is the fact that the nodes present in the sample trees satisfy a single order relation \mathcal{O}_c . However, the sample set \mathcal{D} may have a more complex class structure and it may be necessary to describe it using multiple tree unions. Under these conditions the unsupervised learning process must allow for multiple classes. We represent the distribution of sample trees using a mixture model over separate union structures. Suppose that there are k tree-unions and that the tree union for the class c is denoted by \mathcal{T}_c , and that the mixing proportion for this tree-union is α_c . The mixture model for the distribution of sample trees is

$$P(t|\bar{\mathcal{T}}, \mathcal{C}) = \sum_{c=1}^k \alpha_c \prod_{t \in \mathcal{D}} \prod_{i \in \mathcal{N}^t} \phi(i|t, \mathcal{T}_c, \mathcal{C}).$$

The expected log-likelihood function for the mixture model over the sample-set \mathcal{D} is:

$$\mathcal{L}(\mathcal{D}|\bar{\mathcal{T}}, \mathcal{C}, \bar{z}) = \sum_{t \in \mathcal{D}} \sum_{i \in \mathcal{N}^t} \sum_{c=1}^k z_c^t \alpha_c \ln \phi(i|t, \mathcal{T}_c, \mathcal{C}),$$

where z_c^t is an indicator variable, that takes on the value 1 if tree t belongs to the mixture component c , and is zero otherwise.

We hence require an information criterion that can be used to select the set of tree merge operations over the sample set \mathcal{D} that results in the optimal set of tree-unions. It is well known that the maximum likelihood criterion cannot be directly used to estimate the number of mixture components, since the maximum of the

likelihood function is a monotonic function on the number of components. In order to overcome this problem we use the Minimum Description Length (MDL) principle [13], which asserts that the model that best describes a set of data is that which minimizes the combined cost of encoding the model, and, the error between the model and the data. The MDL principle allows us to select from a family of possibilities the most parsimonious model that best approximates the underlying data.

More formally, the expected descriptor length of a data set \mathcal{D} generated by an estimate \mathcal{H} of the true or underlying model \mathcal{H}^* is

$$\begin{aligned} E[\text{LL}(\mathcal{D}, \mathcal{H})] &= - \int P(\mathcal{D}|\mathcal{H}^*) \log [P(\mathcal{D}|\mathcal{H})P(\mathcal{H})] d\mathcal{D} = \\ &\quad - \frac{1}{P(\mathcal{H}^*)} \int P(\mathcal{D}, \mathcal{H}^*) \log [P(\mathcal{D}, \mathcal{H})] d\mathcal{D} = \\ &= - \frac{1}{P(\mathcal{H}^*)} \left[\int P(\mathcal{D}, \mathcal{H}^*) \log (P(\mathcal{D}, \mathcal{H}^*)) d\mathcal{D} + \int P(\mathcal{D}, \mathcal{H}^*) \log \left(\frac{P(\mathcal{D}, \mathcal{H})}{P(\mathcal{D}, \mathcal{H}^*)} \right) d\mathcal{D} \right] = \\ &\quad \frac{1}{P(\mathcal{H}^*)} [I(P(\mathcal{D}, \mathcal{H}^*)) + KL(P(\mathcal{D}, \mathcal{H}^*), P(\mathcal{D}, \mathcal{H}))], \end{aligned} \quad (4)$$

where

$$I(P(\mathcal{D}, \mathcal{H}^*)) = - \int P(\mathcal{D}, \mathcal{H}^*) \log (P(\mathcal{D}, \mathcal{H}^*)) d\mathcal{D}$$

is the entropy of the joint probability of the data and the underlying model \mathcal{H}^* , and

$$KL(P(\mathcal{D}, \mathcal{H}^*), P(\mathcal{D}, \mathcal{H})) = - \int P(\mathcal{D}, \mathcal{H}^*) \log \left(\frac{P(\mathcal{D}, \mathcal{H})}{P(\mathcal{D}, \mathcal{H}^*)} \right) d\mathcal{D}$$

is the Kullback-Leiber divergence between the joint probabilities using the underlying model \mathcal{H}^* and the estimated model \mathcal{H} . This quantity is minimized when $\mathcal{H} = \mathcal{H}^*$, and hence $P(\mathcal{D}, \mathcal{H}) = P(\mathcal{D}, \mathcal{H}^*)$.

Under these conditions $KL(P(\mathcal{D}, \mathcal{H}^*), P(\mathcal{D}, \mathcal{H})) = 0$ and $E[\text{LL}(\mathcal{D}, \mathcal{H})] = I(P(\mathcal{D}, \mathcal{H}))$. In other words, the description length associated with the maximum likelihood set of parameters is just the expected value of the negative log likelihood, i.e. the Shannon entropy.

As noted above, the cost incurred in describing or encoding the model $\bar{\mathcal{T}}$ is $-\log [P(\bar{\mathcal{T}})]$, while the cost of describing the data \mathcal{D} using that model is $-\log [P(\mathcal{D}|\bar{\mathcal{T}})]$. Making the dependence on the correspondences \mathcal{C} explicit, we have that the description length is $LL(\mathcal{D}|\mathcal{T}) = -\mathcal{L}(\mathcal{D}|\bar{\mathcal{T}}, \mathcal{C})$. Asymptotically the cost of describing the set of mixing components $\bar{\alpha} = \{\alpha_c; c = 1, \dots, k\}$ and the set of indicator variables $\bar{z} = \{z_c^t | t \in \mathcal{D}, c = 1, \dots, k\}$ is bounded by $mI(\bar{\alpha})$, where m is the number of samples in \mathcal{D} and $I(\bar{\alpha}) = -\sum_{c=1}^k \alpha_c \log(\alpha_c)$ is the entropy of the mixture distribution $\bar{\alpha}$. We assume that the weight distribution is encoded as a histogram. Hence, we commence by dividing the weight space of the samples associated with the node i of the union-tree c into buckets of

width $k\sigma_i^c$. As a result, the probability that a weight falls in a bucket centered at x is, for infinitesimally small k $b_c^i(x) = \frac{k}{\theta_i^c \sqrt{2\pi}} \exp[-\frac{1}{2}(\frac{x}{\sigma_i^c} - \tau_i^c)^2]$. Hence, the asymptotic cost of describing the node parameters τ_i^c and σ_i^c and, at the same time, describing within the specified precision the $n\alpha_c$ samples associated to node i in union c , is

$$\text{LL}_c^i(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) = -(m\alpha_c - p_i) \log(1 - \theta_i^b) - \sum_{j=1}^{p_i} \log(b_c^i(w_j^i)).$$

where $\theta_i^c = 1 - \text{erfc}(\tau_i)$ is the sampling probability for node i and p_i is the number of times the correspondence \mathcal{C} maps a sample-node to i . Hence $(m\alpha_c - p_i)$ is the number of times node i has not been sampled according to the correspondence map \mathcal{C} . As a result

$$\text{LL}(\mathcal{D}|\mathcal{H}, \mathcal{C}) = mI(\bar{\alpha}) + \sum_{c=1}^k \sum_{i \in \mathcal{N}_c} [\text{LL}_c^i(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) + l]. \quad (5)$$

where l is the description length per node of the tree-union structure, which we set to 1.

4 Learning the Mixture

With the description length criterion to hand, our aim is to locate tree merges that give rise to the set of tree unions that optimally partition the training data \mathcal{D} into non-overlapping classes. Unfortunately, locating the global minimum of the descriptor length in this way is an intractable combinatorial problem. Moreover, the Expectation-Maximization algorithm may not be used since the complexity of the maximization step grows exponentially due to the fact that the membership indicators admit the possibility that each union can potentially include every sample-tree. Hence, we resort to a local search technique, which allows us to limit the complexity of the maximization step. The approach is as follows.

- Commence with an overly-specific model. We use a structural model per sample-tree, where each model is equiprobable and structurally identical to the respective sample-tree, and each node has unit sample probability.
- Iteratively generalize the model by merging pairs of tree-unions. The candidates for merging are chosen so that they maximally decrease the descriptor length.
- The algorithm stops when there are no merges remaining that can decrease the descriptor length.

The main requirement of our description length minimization algorithm is that we can optimally merge two tree models. Given two tree models \mathcal{T}_1 and \mathcal{T}_2 , we wish to construct a union $\hat{\mathcal{T}}$ whose structure respects the hierarchical

constraints present in both \mathcal{T}_1 and \mathcal{T}_2 , and that also minimizes the quantity $\text{LL}(\hat{\mathcal{T}})$. Since the trees \mathcal{T}_1 and \mathcal{T}_2 already assign node correspondences \mathcal{C}_1 and \mathcal{C}_2 from the data samples to the model, we can simply find a map \mathcal{M} from the nodes in \mathcal{T}_1 and \mathcal{T}_2 to $\hat{\mathcal{T}}$ and transitively extend the correspondences from the samples to the final model $\hat{\mathcal{T}}$ in such a way that, given two nodes $v \in \mathcal{N}_1$ and $v' \in \mathcal{N}_2$, then $\hat{\mathcal{C}}(v) = \hat{\mathcal{C}}(v') \Leftrightarrow v' = \mathcal{M}(v)$.

Posed as the merge of two structures, the correspondence problem is reduced to that of finding the set of nodes in \mathcal{T}_1 and \mathcal{T}_2 that are common to both trees. Starting with the two structures, we merge the sets of nodes that reduces the descriptor length by the largest amount, while still satisfying the hierarchical constraint. That is we merge nodes u and v of \mathcal{T}_1 with node u' and v' of \mathcal{T}_2 respectively if and only if $u \rightsquigarrow v \Leftrightarrow u' \rightsquigarrow v'$, where $a \rightsquigarrow b$ indicates that a is an ancestor of b .

The descriptor length advantage obtained by merging the nodes v and v' is:

$$\mathcal{A}(v, v') = \text{LL}^v(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) + \text{LL}^{v'}(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) - \text{LL}^{(vv')}(\mathcal{D}|\mathcal{T}_c, \mathcal{C}) + l. \quad (6)$$

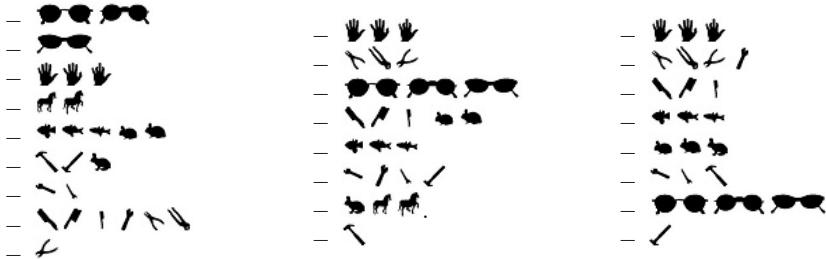
The set of merges \mathcal{M} that minimizes the descriptor length of the combined tree-union also maximizes the advantage function

$$\mathcal{A}(\mathcal{M}) = \sum_{(v, v') \in \mathcal{M}} \mathcal{A}(v, v').$$

For each pair of initial mixture components we calculate the union and the descriptor length of the merged structure. From the set of potential merges, we can identify the one which is both allowable and which reduces the descriptor cost by the greatest amount. The mixing proportion for this optimal merge is equal to the sum of the proportions of the individual unions. At this point we calculate the union and descriptor cost that results from merging the newly obtained model with each of the remaining components. We iterate the algorithm until no more merges can be found that reduce the descriptor length.

5 Pattern Spaces from Union Trees

We can use the union-trees to embed the shapes of the same class in a pattern space using principal components analysis. To do this we place the nodes of the union tree \mathcal{T}_c in an arbitrary order. To each sample tree t we associate a pattern-vector $\mathbf{x}_t = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, where $n = |\mathcal{N}_c|$ is the number of nodes in the tree model \mathcal{T}_c . Here $x_t(i) = w_i^T$ if the tree has a node mapped to the i -th node of the sample and is zero otherwise. For each union-tree \mathcal{T}_c we compute the mean pattern-vector $\hat{\mathbf{x}}_c = \frac{1}{|\mathcal{N}_c|} \sum_{t \in \mathcal{N}_c} \mathbf{x}_t$ and covariance matrix $\Sigma_c = \frac{1}{|\mathcal{N}_c|} \sum_{t \in \mathcal{N}_c} (\mathbf{x}_t - \hat{\mathbf{x}}_c)(\mathbf{x}_t - \hat{\mathbf{x}}_c)^T$ where \mathcal{N}_c is the set of sample trees merged to form the tree union \mathcal{T}_c . Suppose that the eigenvectors (ordered to decreasing eigenvalue) are $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{N_c}$. The leading l_{sig} eigenvectors are used to form the columns of the matrix $E = (\mathbf{e}_1 | \mathbf{e}_2 | \dots | \mathbf{e}_{l_{sig}})$. We perform PCA on the sample-trees by projecting the pattern-vectors onto the leading eigenvectors of the covariance matrix. The



a) Mixture of unattributed b) Weighted edit-distance. c) Union of attributed tree models.

Fig. 1. Clusters extracted with a purely-structural mixture of trees approach versus pairwise clustering of attributed distances obtained with edit distance and tree union.

projection of the pattern-vector for the sample tree indexed t is $\mathbf{y}_t = E^T \mathbf{x}_t$. The distance between the vectors in this space is $D^{PCA}(t, t')(\mathbf{y}_t - \mathbf{y}_{t'})^T(\mathbf{y}_t - \mathbf{y}_{t'})$.

6 Experimental Results

We illustrate the utility of the tree-clustering algorithm on sets of shock trees. The shock tree is a graph-based representation of the differential structure of the boundary of a 2D shape. We augment the skeleton topology with a measure of feature importance based on the rate of change of boundary length with distance along the skeleton.

6.1 Clustering Examples

To illustrate the clustering process, we commence with a study on a small database of 25 shapes. In order to assess the quality of the method, we compare the clusters defined by the components of the mixture with those obtained by applying a graph spectral pairwise clustering method recently developed by Robles-Kelly and Hancock [14] to the distances between graphs. This method locates the clusters by iteratively extracting the eigenvectors from the matrix of edit-distances between the graphs. The edit distances are computed in two alternative ways. First, we compute weighted edit distance using the method outlined in [17]. The second method involves computing the distance matrix using the projected vectors by embedding the trees in a single tree union [18]. These two distance measures are enhanced with geometrical information linked to the nodes of the trees in the form of a node weight. The weight of each node is equal to the proportion of the boundary length that generated the skeletal branch associated to the node.

Figure 1 shows the clusters extracted from the database of 25 shapes. The first column shows the clusters extracted through the mixture of tree unions

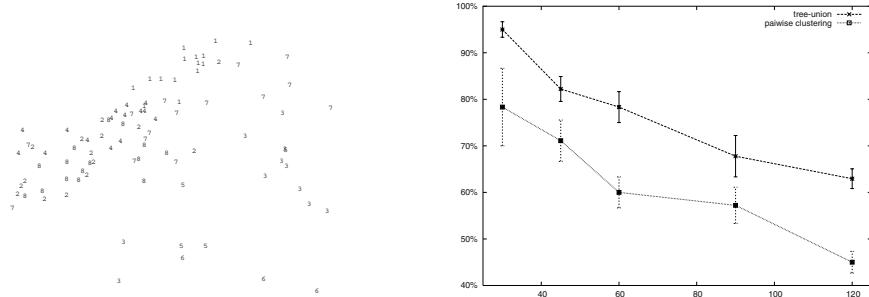


Fig. 2. Left: 2D multi-dimensional scaling of the pairwise distances of the shock graphs. (The numbers correspond to the shape classes.); Right: Proportion of correct classifications obtained with the mixture of tree versus those obtained with pairwise clustering.

approach, and relies on a purely structural representation of shape. The second column displays the clusters extracted from the weighted edit-distances between shock-trees; here the structural information is enhanced with geometrical information. The third column shows the clusters extracted from the distances obtained by embedding the geometrically-enhanced shock-trees in a single tree-union. While there is some merge and leakage, the clusters extracted with the mixture of tree unions compare favorably with those obtained using the alternative clustering algorithms, even though these are based on data enhanced with geometrical information. The second to last cluster extracted using the mixture of tree unions deserves some further explanation. The structure of the shock-trees of the distinct tools in the cluster are identical. Hence, by using only structural information, the method clusters the shock-trees together. To distinguish between the objects, geometrical information must be provided too. Hence, the two alternative clustering methods are able to distinguish between the wrenches, brushes and pliers.

A more challenging experimental vehicle is provided by a larger database of 120 trees, which is divided into 8 shape classes containing 15 shapes each. To perform an initial evaluation of this database, we have applied multidimensional scaling to the weighted edit distances between the shock graphs for the different shapes. By doing this we embed points representing the graphs in a low dimensional space spanned by the eigenvectors of a similarity matrix computed from the pairwise distances. In Figure 2 we show the projection of the graphs onto the 2D space spanned by the leading two eigenvectors of the similarity matrix. Each label in the plot corresponds to a particular shape class. Label 1 identifies hands, label 2 horses, label 3 ducks, 4 men, 5 pliers, 6 screwdrivers, 7 dogs, and, finally, label 8 is associated with leaves. The plot clearly shows the difficulty of this clustering problem. The shape groups are not well separated. Rather, there is a good deal of overlap between them. Furthermore, there are a considerable number of outliers.

To assess the ability of the clustering algorithm to separate the shape classes, we performed experiments on an increasing number of shapes. We commenced

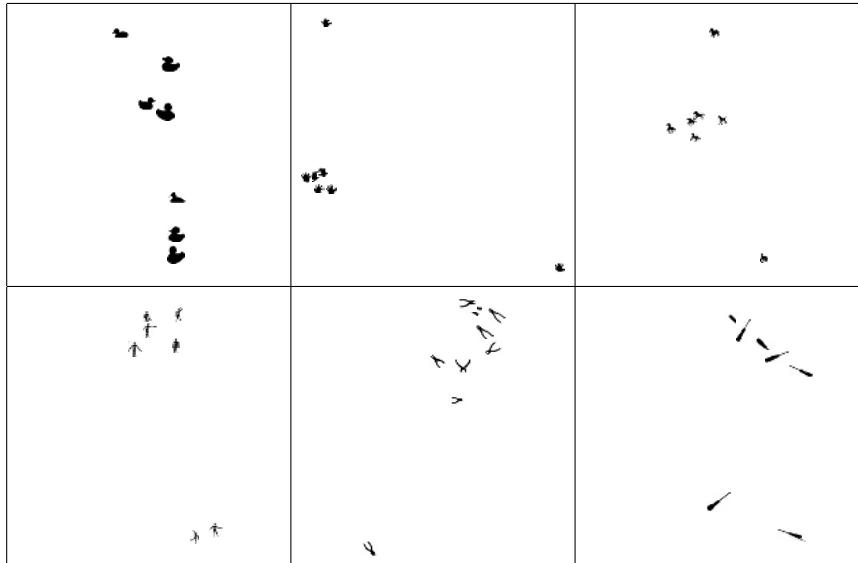


Fig. 3. Principal components analysis of the union embedding of the clusters.

with the 30 shapes from the first two shape classes, and then increased the number of shape classes under consideration until the full set of 120 shapes was included. Figure 2 plots the proportion of shapes correctly classified as the number of shapes is increased. The solid line plots the result obtained using the mixture of weighted tree unions, while the dotted line displays the results obtained with pairwise clustering of the weighted edit distances between the shapes. The mixture of tree unions clearly outperforms the pairwise clustering algorithm.

We now turn our attention to the results of applying PCA to the union trees, as described in Section 5. Figure 3 displays the first two principal components of the sample-tree distribution for the embedding spaces extracted from six shape classes. In most cases there appears to be a tightly packed central cluster with a few shapes scattered further away than the rest. This separation is linked to substantial variations in the structure of the shock trees. For example, in the shape-space formed by the class of pliers the outlier is the only pair-of-pliers with the nose closed. In the case of shape-space for the horse-class, the outliers appear to be the cart-horses while the inliers are the ponies.

7 Conclusions

In this paper we have presented an information theoretic framework for clustering trees and for learning a generative model of the variation in tree structure. The problem is posed as that of learning a mixture of tree unions. We demonstrate how the three sets of operations needed to learn the generative model,

namely node correspondence, tree merging and node probability estimation, can each be couched in terms of minimising a description length criterion. We provide variants of algorithm that can be applied to samples of both weighted and unweighted trees. The method is illustrated on the problem of learning shape-classes from sets of shock trees.

References

1. C. Cyr and B.Kimia, 3D Object Recognition Using Shape Similarity-Based Aspect Graph, *ICCV* 2001.
2. S. J. Dickinson, A. P. Pentland, and A. Rosenfeld, 3-D shape recovery using distributed aspect matching, *PAMI*, Vol. 14(2), pp. 174-198, 1992.
3. N. Friedman and D. Koller, Being Bayesian about Network Structure, *Machine Learning*, to appear, 2002
4. L. Getoor et al., Learning Probabilistic models of relational structure, in *8th Int. Conf. on Machine Learning*, 2001.
5. D. Heckerman, D. Geiger, and D. M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, *Machine Learning*, Vol. 20(3), pp. 197-243, 1995.
6. T. Heap and D. Hogg, Wormholes in shape space: tracking through discontinuous changes in shape, in *ICCV*, pp. 344-349, 1998.
7. G.R. Hjaltason and H. Samet, Properties of embedding methods for similarity searching in metric spaces, *PAMI*(25), pp. 530-549, 2003.
8. S. Ioffe and D. A. Forsyth, Human Tracking with Mixtures of Trees, *ICCV*, Vol. I, pp. 690-695, 2001.
9. Y. Keselman, A. Shokoufandeh, M.F. Demirci, and S.Dickinson, Many-to-many graph matching via metric embedding, *CVPR03*(I: 850-857).
10. B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, Shapes, shocks, and deformations I, *International Journal of Computer Vision*, Vol. 15, pp. 189-224, 1995.
11. N. Linial, E. London and Y. Rabinovich, The geometry of graphs and some of its applications, 35th Anual Symposium on Foundations of Computer Science, pp. 169-175, 1994.
12. M. Meilă. *Learning with Mixtures of Trees*. PhD thesis, MIT, 1999.
13. J. Rissanen, Stochastic complexity and modeling, *Annals of Statistics*, Vol. 14, pp. 1080-1100, 1986.
14. A. Robles-Kelly and E. R. Hancock. A maximum likelihood framework for iterative eigendecomposition. In *ICCV*, Vol. I, pp. 654-661, 2001.
15. A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker, Indexing using a spectral encoding of topological structure, in *CVPR*, 1999.
16. T. Sebastian, P. Klein, and B. Kimia, Recognition of shapes by editing shock graphs, in *ICCV*, Vol. I, pp. 755-762, 2001.
17. A. Torsello and E. R. Hancock. Efficiently computing weighted tree edit distance using relaxation labeling. In *EMMCVPR*, pp. 438-453, 2001.
18. A. Torsello and E. R. Hancock, Matching and embedding through edit-union of trees. In *ECCV*, pp. 822-836, 2002.
19. S. C. Zhu and A. L. Yuille, FORMS: A Flexible Object Recognition and Modelling System, *IJCV*, Vol. 20(3), pp. 187-212, 1996.

Decision Theoretic Modeling of Human Facial Displays

Jesse Hoey and James J. Little

Department of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, BC, CANADA V6T 1Z4
{jhoey,little}@cs.ubc.ca

Abstract. We present a vision based, adaptive, decision theoretic model of human facial displays in interactions. The model is a partially observable Markov decision process, or POMDP. A POMDP is a stochastic planner used by an agent to relate its actions and utility function to its observations and to other context. Video observations are integrated into the POMDP using a dynamic Bayesian network that creates spatial and temporal abstractions of the input sequences. The parameters of the model are learned from training data using an *a-posteriori* constrained optimization technique based on the expectation-maximization algorithm. The training does not require facial display labels on the training data. The learning process *discovers* clusters of facial display sequences and their relationship to the context automatically. This avoids the need for human intervention in training data collection, and allows the models to be used without modification for facial display learning in any context without prior knowledge of the type of behaviors to be used. We present an experimental paradigm in which we record two humans playing a game, and learn the POMDP model of their behaviours. The learned model correctly predicts human actions during a simple cooperative card game based, in part, on their facial displays.

1 Introduction

There has been a growing body of work in the past decade on the communicative function of the face [1]. This psychological research has drawn three major conclusions. First, facial displays are often purposeful communicative signals. Second, the purpose is not defined by the display alone, but is dependent on both the display and the context in which the display was emitted. Third, the signals are not universal, but vary widely between individuals in their physical appearance, their contextual relationships, and their purpose. We believe that these three considerations should be used as critical constraints in the design of communicative agents able to learn, recognise, and use human facial signals. They imply that a rational communicative agent must learn the relationships between facial displays, the context in which they are shown, and its own utility function: it must be able to compute the utility of taking actions in situations involving purposeful facial displays. The agent will then be able to make

value-directed decisions based, in part, upon the “meaning” of facial displays as contained in these learned connections between displays, context, and utility. The agent must also be able to adapt to new interactants and new situations, by learning new relationships between facial displays and other context.

This paper presents a vision-based, adaptive, Bayesian model of human facial displays. The model is, in fact, a partially observable Markov decision process, or POMDP [2], with spatially and temporally abstract, continuous observations over the space of video sequences. The POMDP model integrates the recognition of facial signals with their interpretation and use in a utility-maximization framework. This is in contrast to other approaches, such as hidden Markov models, which consider that the goal is simply to categorize a facial display. POMDPs allow an agent to make decisions based upon facial displays, and, in doing so, define facial displays by their use in decision-making. Thus, the POMDP training is freed from the curse of labeling training data which expresses the bias of the labeler, not necessarily the structure of the task. The model can be acquired from data, such that an agent can learn to act based on the facial signals of a human through observation. To ease the burden on decision-making, the model builds temporal and spatial abstractions of input video data. For example, one such abstraction may correspond with the wink of an eye, whereas another may correspond to a smile. These abstractions are also learned from data, and allow decision making to occur over a small set of states which are accurate temporal and spatial summarizations of the continuous sensory signals.

Our work is distinguished from other work on recognising facial communications primarily because the facial displays are not defined prior to learning the model. We do not train classifiers for different facial motions and then base decisions upon the classifier outputs. Instead, the training process *discovers* categories of facial displays in the data and their relationships with context. The advantage of learning without pre-defined labels is threefold. First, we do not need labeled training data, nor expert knowledge about which facial motions are important. Second, since the system learns categories of motions, it will adapt to novel displays without modification. Third, resources can be focused on useful tasks for the agent. It is wasteful to train complex classifiers for the recognition of fine facial motion if only simple displays are being used in the agent’s context.

The POMDPs we learn have observations which are video sequences, modeled with mixtures of coupled hidden Markov models (CHMMs) [3]. The CHMM is used to couple the images and their derivatives, as described in Section 3.1. While it is usual in a hierarchical model to commit to a most likely value at a certain level [4,5], our models propagate noisy evidence from video at the lowest level to actions at the highest, and the choice of actions can be probabilistically based upon all available evidence.

2 Previous Work

There are many examples of work in computer vision analysing facial displays [6], and human motion in general [7,4]. However, this work is usually supervised, in

that models of particular classes of human motion are learned from labeled training data. There has been some recent research in unsupervised learning of motion models [8,5], but few have attempted to explicitly include the modeling of actions and utility, and none have looked at facial displays. Action-Reaction Learning [9] is a system for analysing and synthesising human behaviours. It is primarily reactive, however, and does not learn models conducive for high level reasoning about the long term effects of actions.

Our previous work on this topic has led to the development of many parts of the system described in this paper. In particular, the low-level computer vision system for instantaneous action recognition was described in [10], while the simultaneous learning of the high-level parameters was explored in [11]. This paper combines this previous work, explicitly incorporates actions and utilities, and demonstrates how the model is a POMDP, from which policies of action can be extracted. Complete details can be found in [12].

POMDPs have become the semantic model of choice for decision theoretic planning in the artificial intelligence (AI) community. While solving POMDPs optimally is intractable for most real-world problems, the use of approximation methods have recently enabled their application to substantial planning problems involving uncertainty, for example, card games [13] and robot control [14]. POMDPs were applied to the problem of active gesture recognition in [15], in which the goal is to model unobservable, non-foveated regions. This work models some of the basic mechanics underlying dialogue, such as turn taking, channel control, and signal detection. Work creating embodied agents has led to much progress in creating agents that interact using verbal and non-verbal communication [16]. These agents typically only use a small subset of manually specified facial expressions or gestures. They focus instead primarily on dialogue management and multi-modal inputs, and have not used POMDPs.

3 POMDPs for Facial Display Understanding

A POMDP is a probabilistic temporal model of an agent interacting with the environment [2], shown as a Bayesian network in Figure 1(a). A POMDP is similar to a hidden Markov model in that it describes observations as arising from hidden states, which are linked through a Markovian chain. However, the POMDP adds actions and rewards, allowing for decision theoretic planning. A POMDP is a tuple $\langle S, A, T, R, O, B \rangle$, where S is a finite set of (possible unobservable) states of the environment, A is a finite set of agent actions, $T : S \times A \rightarrow S$ is a transition function which describes the effects of agent actions upon the world states, O is a set of observations, $B : S \times A \rightarrow O$ is an observation function which gives the probability of observations in each state-action pair, and $R : S \rightarrow \mathcal{R}$ is a real-valued reward function, associating with each state s its immediate utility $R(s)$. A POMDP model allows an agent to predict the long term effects of its actions upon his environment, and to choose actions based on these predictions. Factored POMDPs [18] represent the state, S , using a set of variables, such that the state space is the product of the spaces of each variable. Factored POMDPs

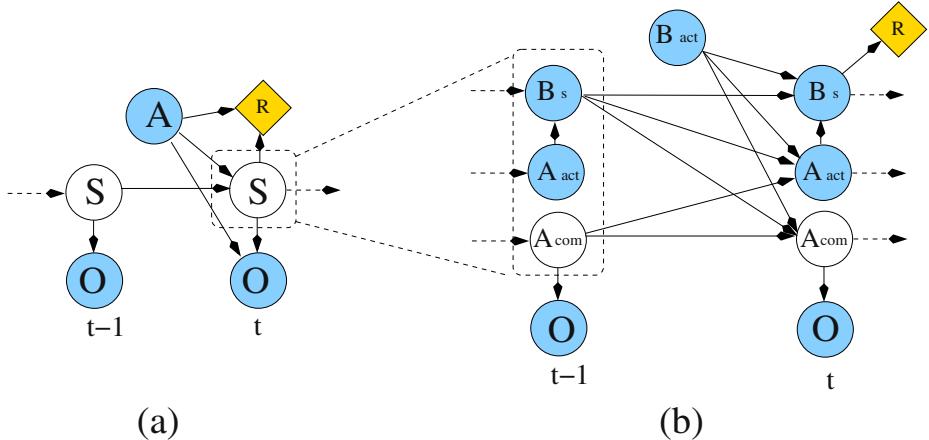


Fig. 1. (a) Two time slices of general POMDP. (b) Two time slices of factored POMDP for facial display understanding. The state, S , has been factored into $\{Bs, Aact, Acom\}$, and conditional independencies have been introduced: Ann’s actions do not depend on her previous actions and Ann’s display is independent of her previous action given the state and her previous display. These independencies are not strictly necessary, but simplify our discussion, and are applicable in the simple game we analyse.

allow conditional independencies in the transition function, T , to be leveraged. Further, T is written as a set of smaller, more intuitive functions.

Purposeful facial display understanding implies a multi-agent setting, such that each agent will need to model all other agent’s decision strategies as part of its internal state¹. In the following, we will refer to the two agents we are modeling as “Bob” and “Ann”, and we will discuss the model from Bob’s perspective. Figure 1(b) shows a factored POMDP model for facial display understanding in simple interactions. The state of Bob’s POMDP is factored into Bob’s private internal state, Bs , Ann’s action, $Aact$, and Ann’s facial display, $Acom$, such that $S_t = \{Bs_t, Aact_t, Acom_t\}$. While Bs and $Aact$ are observable, $Acom$ is not, and must be inferred from video sequence observations, O . We wish to focus on learning models of facial displays, $Acom$, and so we will use games in which $Aact$ and Bs are fully observable, which they are not in general. For example, in a real game of cards, a player must model the suit of any played card as an unobservable variable, which must be inferred from observations of the card. In our case, games will be played through a computer interface, and so these kinds of actions are fully observable.

The transition function is factored into four terms. The first involves only fully observable variables, and is the conditional probability of the state at time t under the effect of both player’s actions: $\Theta_S = P(Bs_t | Aact_t, Bact, Bs_{t-1})$.

¹ This is known as the *decision analytic* approach to games, in which each agent decides upon a strategy based upon his subjective probability distribution over the strategies employed by other players.

The second is over Ann’s actions given Bob’s action, the previous state, and her previous display: $\Theta_A = P(Aact_t | Bact, Acom_{t-1}, Bs_{t-1})$. The third describes Bob’s expectation about Ann’s displays given his action, the previous state and her previous display: $\Theta_D = P(Acom_t | Bact, Bs_{t-1}, Acom_{t-1})$. The fourth describes what Bob expects to see in the video of Ann’s face, \mathbf{O} , given his high-level descriptor, $Acom$: $\Theta_O = P(\mathbf{O}_t | Acom_t)$. For example, for some state of $Acom$, this function may assign high likelihood to sequences in which Ann smiles. This value of $Acom$ is only assigned meaning through its relationship with the context and Bob’s action and utility function. We can, however, look at this observation function, and interpret it as an $Acom$ = ‘smile’ state. Writing $C_t = \{Bact_t, Bs_{t-1}\}$, $A_t = Aact_t$, and $D_t = Acom_t$, the likelihood of a sequence of data, $\{\mathbf{OCA}\}_{1,T} = \{\mathbf{O}_1 \dots \mathbf{O}_T, C_1 \dots C_T, A_1 \dots A_T\}$, is

$$P(\{\mathbf{OCA}\}_{1,T} | \Theta) = \sum_k P(\mathbf{O}_T | D_{T,k}) \sum_l \Theta_A \Theta_D P(D_{T-1,l}, \{\mathbf{OCA}\}_{1,T-1} | \Theta) \quad (1)$$

where $D_{t,k}$ is the k^{th} value of the mixture state, D , at time t . The observations, \mathbf{O} , are temporal sequences of finite extent. We assume that the boundaries of these temporal sequences will be given by the changes in the fully observable context state, C and A . There are many approaches to this problem, ranging from the complete Bayesian solution in which the temporal segmentation is parametrised and integrated out, to specification of a fixed segmentation time [4].

3.1 Observations

We now must compute $P(\mathbf{O} | Acom)$, where \mathbf{O} is a sequence of video frames. We have developed a method for generating temporally and spatially abstract descriptions of sequences of facial displays from video [10,12]. We give a brief outline of the method here. Figure 2 shows the model as a Bayesian network being used to assess a sequence in which a person smiles.

We consider that spatially abstracting a video frame during a human facial display involves modeling both the current configuration and dynamics of the face. Our observations consist of the video images, I , and the temporal derivatives, f_t , between pairs of images. The task is first to spatially summarise both of these quantities, and then to temporally compress the entire sequence to a distribution over high level descriptors, $Acom$. We assume that the face region is tracked through the sequence by a separate tracking process, such that the observations arise from the facial region in the images only. We use a flow-based tracker, described in more detail in [12].

The spatial abstraction of the derivative fields involves a projection of the associated optical flow field, v , over the facial region to a set of pre-determined basis functions. The basis functions are a complete and orthogonal set of 2D polynomials which are effective for describing flow fields [12]. The resulting feature vector, Z_x , is then conditioned on a set of discrete states, X , parametrised by normal distributions. The projection is accomplished by analytically integrating the observation likelihood, $P(f_t | X)$, over the space of optical flow fields and

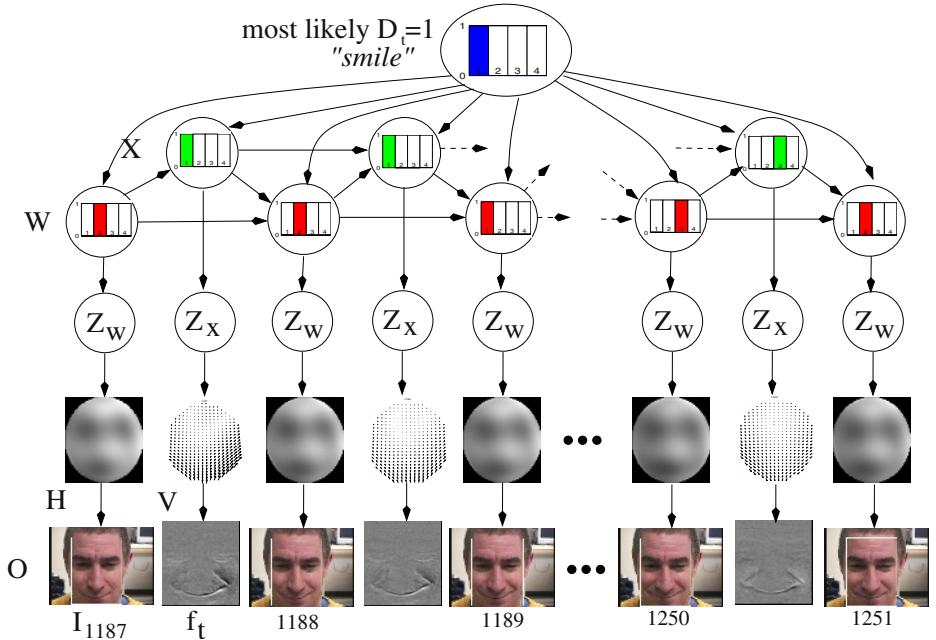


Fig. 2. A person smiling is analysed by the mixture of CHMMs. Observations, O , are sequences of images, I , and image temporal derivatives, f_t , both of which are projected over the facial region to a set of basis functions, yielding feature vectors, Z_x and Z_w . The image regions, H , are projected directly, while it is actually the optical flow fields, V , related to the image derivatives which are projected to the basis functions [10]. Z_x and Z_w are both modeled using mixtures of Gaussians, X and W , respectively. The class distributions, X and W , are temporally modeled as mixture, D , of coupled Markov chains. The probability distribution over D is at the top. The most likely state, $D = 1$, can be associated with the concept “smile”. Probability distributions over X and W are shown for each time step. All other nodes in the network show their expected value given all evidence. Thus, the flow field, v , is actually $\langle v \rangle = \int_v v P(v|O)$.

over the feature vector space. This method ensures that all observation noise is propagated to the high level [10]. The abstraction of the images also uses projections of the raw (grayscale) images to the same set of basis functions, resulting in a feature vector, Z_w , which is also modeled using a mixture of normal distributions with mixture coefficients W .

The basis functions are a complete and orthogonal set, but only a small number may be necessary for modeling any particular motion. We use a feature weighting technique that places priors on the normal means and covariances, so that choosing a set of basis functions is handled automatically by the model [10].

At each time frame, we have a discrete dynamics state, X , and a discrete configuration state, W , which are abstract descriptions of the instantaneous dynamics and configuration of the face, respectively. These are temporally abstracted using a mixture of coupled hidden Markov models (CHMM), in which

the dynamics and configuration states are interacting Markovian processes. The conditional dependencies between the X and W chains are chosen to reflect the relationship between the dynamics and configuration. This mixture model can be used to compute the likelihood of a video sequence given the facial display descriptor, $P(\mathbf{O}|A_{com})$:

$$P(\{\mathbf{O}\}|D_T) = \sum_{1,T} P(f_t|X_{T,i})P(I_t|W_{T,j}) \sum_{kl} \Theta_{Xijk} \Theta_{Wjkl} P(X_{T-1,k}, W_{T-1,l} | D_{T-1}) \quad (2)$$

where Θ_X, Θ_W are the transition matrices in the coupled X and W chains, and $P(f_t|X_{T,i}), P(I_t|W_{T,j})$ are the associated observation functions [12]. The mixture components, D , are a set of discrete abstractions of facial behavior. It is important to remember that there are no labels associated with these states at any time during the training. Labels can be assigned after training, as is done in Figure 2, but these are only to ease exposition.

3.2 Learning POMDPs

We use the expectation-maximization (EM) algorithm [17] to learn the parameters of the POMDP. It is important to stress that the learning takes place over the *entire* model simultaneously: both the output distributions, including the mixtures of coupled HMMs, and the high-level POMDP transition functions are all learned from data during the process. The learning classifies the input video sequences into a spatially and temporally abstract finite set, A_{com} , and learns the relationship between these high-level descriptors, the observable context, and the action. We only present some salient results of the derivation here. We seek the set of parameters, Θ^* , which maximize

$$\Theta^* = \arg \max_{\Theta} \left[\sum_{\mathbf{D}} P(\mathbf{D}|\mathbf{O}, \mathbf{C}, \mathbf{A}, \theta') \log P(\mathbf{D}, \mathbf{O}, \mathbf{C}, \mathbf{A}|\Theta) + \log P(\Theta) \right] \quad (3)$$

subject to constraints on the parameters, Θ^* , that they describe probability distributions (they sum to 1). The “E” step of the EM algorithm is to compute the expectation over the hidden state, $P(\mathbf{D}|\mathbf{O}, \mathbf{C}, \mathbf{A}, \theta')$, given θ' , a current guess of the parameter values. The “M” step is then to perform the maximization which, in this case, can be computed analytically by taking derivatives with respect to each parameter, setting to zero and solving for the parameter.

The update for the D transition parameter, $\Theta_{Dijk} = P(D_{t,i}|D_{t-1,j}C_{t,k})$, is then

$$\Theta_{Dijk} = \frac{\alpha_{Dijk} + \sum_{t \in \{1 \dots N_t\} | C_t = k} P(D_{t,i}D_{t-1,j}|\mathbf{O}, \mathbf{A}, \mathbf{C}\theta')}{\sum_i [\alpha_{Dijk} + \sum_{t \in \{1 \dots N_t\} | C_t = k} P(D_{t,i}D_{t-1,j}|\mathbf{O}, \mathbf{A}, \mathbf{C}\theta')]} \quad (4)$$

where the sum over the temporal sequence is only over time steps in which $C_t = k$, and α_{Dijk} is the parameter of the Dirichlet smoothing prior. The summand can be factored as

$$P(D_{t,i}D_{t-1,j}|\mathbf{O}, \mathbf{A}, \mathbf{C}\theta') = \beta_{t,i} \Theta_{A*i*} P(\mathbf{O}_t|D_{t,i}) \Theta_{Dijk} \alpha_{t-1,j}$$

where $\alpha_{t,j} = P(D_{t,j}\{\mathbf{OAC}\})$ and $\beta_{t,i} = P(\{\mathbf{OAC}\}|D_{t,i})$ are the usual forwards and backwards variables, for which we can derive recursive updates

$$\alpha_{t,j} = \sum_k P(\mathbf{O}_t|D_{t,j})\Theta_{A*j*}\Theta_{Djk*}\alpha_{t-1,k} \quad \beta_{t-1,i} = \sum_k \beta_{t,k}\Theta_{A*k*}P(\mathbf{O}_t|D_{t,k})\Theta_{Dki*}$$

where we write $\Theta_{A*j*} = P(A_t = *|D_{t,j}C_t = *)$ and $P(\mathbf{O}_t|D_{t,i})$ is the likelihood of the data given a state of the mixture of CHMMs (Equation 2). The updates to $\Theta_{Aijk} = P(A_{t,i}|D_{t,j}C_{t,k})$ are $\Theta_{Aijk} = \sum_{t \in \{1\dots N_t\} | A_t = i \vee C_t = k} \xi_j$, where $\xi_j = P(D_{t,j}|\mathbf{OAC}) = \beta_{t,j}\alpha_{t,j}$. The updates to the j^{th} component of the mixture of CHMMs are weighted by ξ_j , but otherwise is the same as for a normal CHMM [3]. The complete derivation, along with the updates to the output distributions of the CHMMs, including to the feature weights, can be found in [12].

3.3 Solving POMDPs

If observations are drawn from a finite set, then an optimal policy of action can be computed for a POMDP [2] using dynamic programming over the space of the agent's belief about the state, $b(s)$. However, if the observation space is continuous, as in our case, the problem becomes much more difficult. In fact, there are no known algorithms for computing optimal policies for such problems. Nevertheless, approximation techniques have been developed, and yield satisfactory results [14]. Since our focus in this paper is to learn POMDP models, we use the simplest possible approximation technique, and simply consider the POMDP as a fully observable MDP: the state, S , is assigned its most likely value in the belief state, $S = \arg \max_s b(s)$. Dynamic programming updates then consist of computing value functions, V^n , where $V^n(s)$ gives the expected value of being in state s with a future of n stages to go (horizon of n), assuming the optimal actions are taken at each step. The actions that maximize V^n are the n stage-to-go policy (the policy looking forward to a horizon 3 stages in the future). We use the SPUDD solver to compute these policies [18].

4 Experiments

In order to study the relationships between display recognition and action we constrain the structure of an interaction between two humans using rules in a computer game. We then observe the humans playing the game and learn models of the relationships between their facial motions and the states and actions in the game. Subsequent analysis of the learned models reveals how the humans were using their faces for achieving value in the game. Our learning method allows such games to be analysed without any prior knowledge about what facial displays will be used during game play. The model automatically "discovers" what display classes are present. We can also compute policies of action from the models. In the following, we describe our experiments with a simple card game. Results on two other simple games, along with further details on the game here described, can be found in [12].

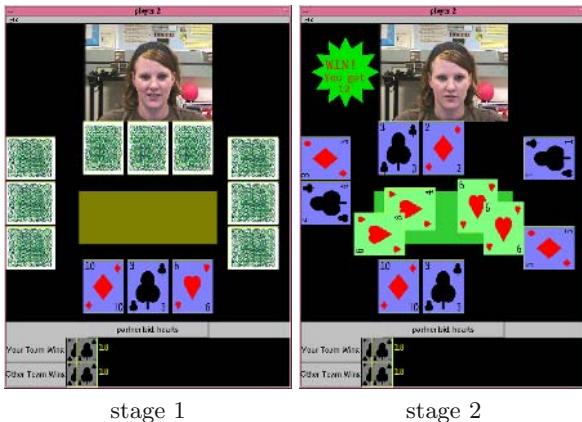


Fig. 3. Bob’s game interfaces during a typical round. His cards are face up below the “table”, while Ann’s cards are above it. The current bid is shown below Bob’s cards, and the winnings are shown along the bottom. The cards along the sides belong to another team, which is introduced only for motivation. A bid of hearts in stage 1 is accepted by Ann, and both players commit their heart in stage 2.

4.1 Card Matching Game

Two players play the card matching game. At the start of a round, each player is dealt three cards from a deck with three suits ($\heartsuit, \diamondsuit, \clubsuit$), with values from 1 to 10. Each player can only see his own set of cards. The players play a single card simultaneously, and both players win the sum of the cards if the suits match. Otherwise, they win nothing. On alternate rounds (*bidding rounds*), a player has an opportunity to send a confidential *bid* to his partner, indicating a card suit. The *bids* are non-binding and do not directly affect the payoffs in the game. During the other rounds (*displaying rounds*), the player can only see her partner’s bids, and then play one of her cards. There is no time limit for playing a card, but the decision to play a card is final once made. Finally, each player can see (but not hear) their teammate through a real-time video link. There are no game rules concerning the video link, so there are no restrictions placed on communication strategies the players can use. The card matching game was played by two students in our laboratory, “Bob” and “Ann” through a computer interface. A picture of Bob’s game interface during a typical interaction is shown in Figure 3. Each player viewed their partner through a direct link from their workstation to a Sony EVI S-video camera mounted about their partner’s screen. The average frame rate at 320×240 resolution was over 28fps. The rules of the game were explained to the subjects, and they played four games of five rounds each. The players had no chance to discuss potential strategies before the game, but were given time to practice.

We will use data from Bob’s bidding rounds in the first three games to train the POMDP model. Observations are three or four variable length video sequences for each round, and the actions and the values of the cards of both

players, as shown in Table 1. The learned model’s performance will then be tested on the data from Bob’s bidding rounds in the last game. It is possible to implement a combined POMDP for both bidding and displaying rounds [12].

There are nine variables which describe the state of the game when a player has the bid. The suit of each the three cards can be one of $\heartsuit, \diamondsuit, \clubsuit$. Bob’s actions, $Bact$, can be *null* (no action), or sending a confidential bid ($bid\heartsuit, bid\diamondsuit, bid\clubsuit$) or committing a card ($cmt\heartsuit, cmt\diamondsuit, cmt\clubsuit$). Ann’s observed actions, $Aact$, can be *null*, or committing a card. The $Acom$ variable describes Ann’s communication through the video link. It is one of N_d high-level states, $D = d_1 \dots d_{N_d}$, of the mixture of CHMMs model described previously. Although these states have no meaning in isolation, they will obtain meaning through their interactions with the other variables in the POMDP. The number of states, (N_d), must be manually specified, but can be chosen as large as possible based on the amount of training data available. The other six, observable, variables in the game are more functional for the POMDP, including the values of the cards, and whether a match occurred or not. The reward function is only based upon fully observable variables, and is simply the sum of the played card values, if the suits match.

4.2 Results

The model was trained with four display states. We inspected the model after training, and found that two of the states (d_1, d_3) corresponded to “nodding” the head, one (d_4) to “shaking” the head, and the last (d_2) to a null display with little motion. Training with only three clusters merges the two nodding clusters together. Figures 4 and 5 show example frames and flows from sequences recognized as d_4 (shake) and as d_1 (nod), respectively. The sequences correspond to the last two rows in Table 1, in which Ann initially refuses a bid of \diamondsuit from Bob, then accepts a bid of \clubsuit .

Table 2(a) shows a part of the learned conditional probability distribution over Ann’s action, $Aact$, given the current bid and Ann’s display, $Acom$. We see that, if the bid is *null*, we expect Ann to do nothing in response. If the bid is \heartsuit , and Ann’s display ($Acom$) is one of the “nodding” displays d_1 or d_3 , then we expect Ann to commit her \heartsuit . On the other hand, if Ann’s display is “shaking”, d_4 , then we expect her to do nothing (and wait for another bid from Bob).

The learned conditional probability distribution of Ann’s display, $Acom$, at time t , given the previous and current bids, bid_{t-1} , and bid_t , carried two important pieces of information for Bob: First, at the beginning of a round, any bid is likely to elicit a non-null display d_1, d_3 or d_4 . Second, a “nodding” display is more likely after a “shaking” display if the bid is changed.

4.3 Computing and Using a Policy

A 3 stage-to-go policy was computed by assuming that the facial display states are observable. There are ten possible values for each card, which expands the state space and makes it more difficult to learn accurate models from limited training data. To reduce this complexity, we approximate these ten values with

Table 1. Log for the first two bidding rounds of one of the training games. A blank means the card values were the same as the previous sequence. Ann’s display, A_{com} , is the most likely as classified by the final model.

round	frames	Bob’s cards		Ann’s cards		<i>bid</i>	Bob’s act	Ann’s act	Ann’s display
		\heartsuit	\diamond	\clubsuit	\heartsuit	\diamond	\clubsuit	\heartsuit	\diamond
1	40-150	3	4	7	2	10	7	-	\clubsuit
1	151-295							\clubsuit	$cmt\clubsuit$
2	725-827	2	5	2	7	3	8	-	\diamond
2	828-976							\diamond	\clubsuit
2	977-1048							\clubsuit	$cmt\clubsuit$

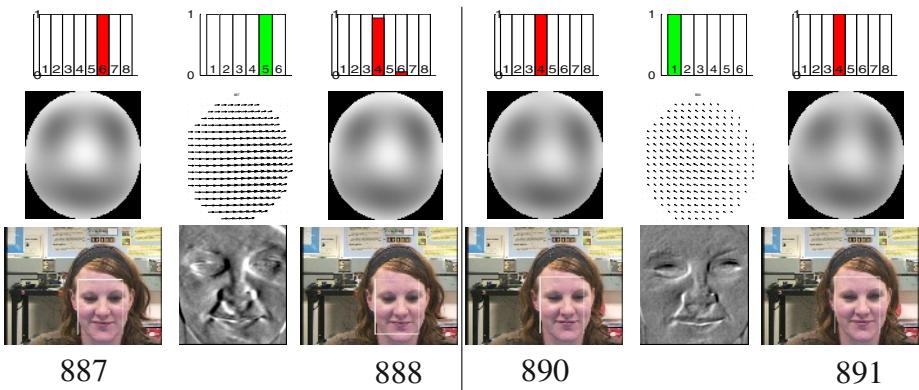


Fig. 4. Frames from the second-to-last row in Table 1. This sequence occurred after Bob had bid \diamond , and was recognized as $A_{com} = d_4$: a head shake. The bottom row shows the original images, I , with tracked face region, and the temporal derivative fields, f_t . The middle row shows the expected configuration, H , and flow field, V (scaled by a factor of 4.0 for visibility). The top row shows distributions over W and X .

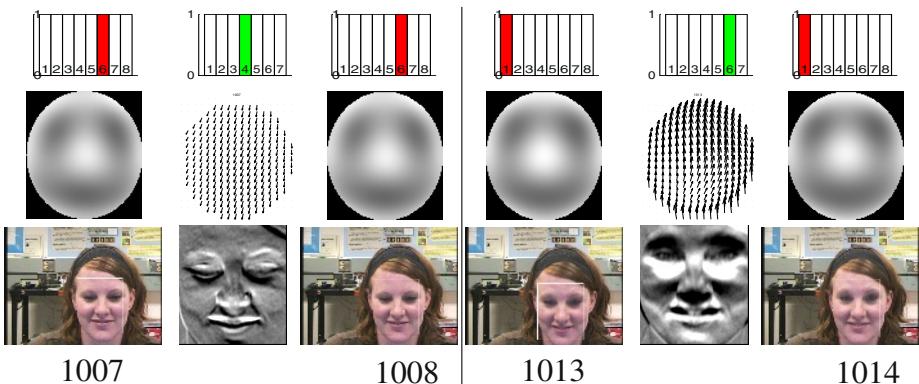


Fig. 5. Frames from the last row in Table 1. This sequence occurred after Bob had made his second bid of \clubsuit after Ann’s negative response to his first bid, and was recognized as $A_{com} = d_1$: a nod. See Figure 4 for more details.

Table 2. (a) Selected parts of the learned conditional probability distribution over Ann’s action, $Aact$, given the current bid and Ann’s display, $Acom$. Even distributions are because of lack of training data. (b) Selected parts of policy of action in the card matching game for the situation in which $B\heartsuit v = v3$, $B\diamondsuit v = v3$ and $B\clubsuit v = v1$.

		(a)						(b)	
bid	$Acom$	$Aact$				bid	$Acom$	policy $Bact$	
		null	$cmt\heartsuit$	$cmt\diamondsuit$	$cmt\clubsuit$			$d1$	$d2, d3$
$null$	-	0.40	0.20	0.20	0.20	“	$d2, d3$	$bid\heartsuit$	$bid\diamondsuit$
\heartsuit	$d1, d3$	0.20	0.40	0.20	0.20	“	$d4$	$cmt\heartsuit$	$cmt\diamondsuit$
\heartsuit	$d2$	0.25	0.25	0.25	0.25	\heartsuit	$d1, d2, d3$	$cmt\heartsuit$	$bid\heartsuit$
\heartsuit	$d4$	0.40	0.20	0.20	0.20	“	$d4$	$bid\diamondsuit$	

three values, $v1, v2, v3$, where cards valued 1-4 are labeled $v1$, 5-7 are $v2$ and 8-10 are labeled $v3$. More training data would obviate the need for this approximation. We then classified the test data with the Viterbi algorithm given the trained model to obtain a fully observable state vector for each time step in the game. The computed policy was consulted, and the recommended actions were compared to Bob’s actual actions taken in the game. The model correctly predicted 6/7 actions in the testing data, and 19/20 in the training data. The error in the testing data was due to the subject glancing at something to the side of the screen, leading to a classification as d_4 . This error demonstrates the need for dialogue management, such as monitoring of the subject’s attention [14].

Table 2(b) shows a part of the policy of action if the player’s cards have values $B\heartsuit v = v3$, $B\diamondsuit v = v3$ and $B\clubsuit v = v1$. For example, if there is no bid on the table, then Bob should bid one of the high cards: hearts or diamonds. If the bid is hearts and Ann nodded or did nothing ($d1, d2$ or $d3$), then Bob should commit his heart. If Ann shook her head, though, Bob should bid the diamond.

Notice that, in Table 2(b), the policy is the same for $Acom = d2, d3$. These states hold similar value for the agent, and could be combined since their distinction is not important for decision making. It is believed that this type of learning, in which the state space is reduced for optimal decision making, will lead to solution techniques for very large POMDPs in the near future [12].

More complex games typically necessitate longer term memory than the Markov assumption we have used. However, POMDPs can accommodate longer dependencies by explicitly representing them in the state space. Further, current research in logical reasoning in first-order POMDPs will extend these models to be able to deal with more complex high-level situations.

5 Conclusion

We have presented an adaptive dynamic Bayesian model of human facial displays in interactions. The model is a partially observable Markov decision process, or POMDP. The model is trained directly on a set of video sequences, and does not need any prior knowledge about the expected types of displays. Without

any behavior labels, the model discovers classes of video sequences and their relationship with actions, utilities and context. It is these relationships which define, or give meaning to, the discovered classes of displays. We demonstrate the method on videos of humans playing a computer game, and show how the model is conducive for intelligent decision making or for prediction.

Acknowledgments. Supported by the Institute for Robotics and Intelligent Systems (IRIS), and a Precarn scholarship. We thank our anonymous reviewers, Pascal Poupart, Nicole Arksey and Don Murray.

References

1. Russell, J.A., Fernández-Dols, J.M., eds.: *The Psychology of Facial Expression*. Cambridge University Press, Cambridge, UK (1997)
2. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. *Artificial Intelligence* **101** (1998) 99–134
3. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: Proc. CVPR (1997), Puerto Rico
4. Oliver, N., Horvitz, E., Garg, A.: Layered representations for human activity recognition. In: Proc. Intl. Conf. on Multimodal Interfaces, Pittsburgh, PA (2002)
5. Galata, A., Cohn, A.G., Magee, D., Hogg, D.: Modeling interaction using learnt qualitative spatio-temporal relations. In: Proc. ECAI. (2002)
6. Tian, Y., Kanade, T., Cohn, J.F.: Recognizing action units for facial expression analysis. *IEEE Trans. PAMI* **23** (2001)
7. Bregler, C.: Learning and recognising human dynamics in video sequences. In: Proc CVPR (1997), Puerto Rico, 568–574
8. Brand, M.: Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation* **11** (1999) 1155–1182
9. Jebara, A., Pentland, A.: Action reaction learning: Analysis and synthesis of human behaviour. In: IEEE Workshop on The Interpretation of Visual Motion. (1998)
10. Hoey, J., Little, J.J.: Bayesian clustering of optical flow fields. In: Proc. ICCV 2003, Nice, France 1086–1093
11. Hoey, J.: Clustering contextual facial display sequences. In: Proceedings of IEEE Intl Conf. on Face and Gesture, Washington, DC (2002)
12. Hoey, J.: Decision Theoretic Learning of Human Facial Displays and Gestures. PhD thesis, University of British Columbia (2004)
13. Fujita, H., Matsuno, Y., Ishii, S.: A reinforcement learning scheme for a multi-agent card game. *IEEE Trans. Syst., Man. & Cybern* (2003) 4071–4078
14. Montemerlo, M., Pineau, J., Roy, N., Thrun, S., Verma, V.: Experiences with a mobile robotic guide for the elderly. In: Proc. AAAI 2002, Edmonton, Canada.
15. Darrell, T., Pentland, A.: Active gesture recognition using partially observable Markov decision processes. In: 13th IEEE ICPR, Austria (1996)
16. Cassell, J., Sullivan, J., Prevost, S., Churchill, E., eds.: *Embodied Conversational Agents*. MIT Press (2000)
17. Dempster, A., Laird, N.M., Rubin, D.: Maximum likelihood from incomplete data using the EM algorithm. *Journal of the Royal Statistical Society* **39** (1977) 1–38
18. Hoey, J., St-Aubin, R., Hu, A., Boutilier, C.: SPUDD: Stochastic planning using decision diagrams. In: Proc. UAI 1999, Stockholm, Sweden

Kernel Feature Selection with Side Data Using a Spectral Approach

Amnon Shashua¹ and Lior Wolf²

¹ School of Engineering and Computer Science, The Hebrew University, Jerusalem
shashua@cs.huji.ac.il

² Center for Biological and Computational Learning, MIT
liorwolf@mit.edu

Abstract. We address the problem of selecting a subset of the most relevant features from a set of sample data in cases where there are multiple (equally reasonable) solutions. In particular, this topic includes on one hand the introduction of hand-crafted kernels which emphasize certain desirable aspects of the data and, on the other hand, the suppression of one of the solutions given “side” data, i.e., when one is given information about undesired aspects of the data. Such situations often arise when there are several, even conflicting, dimensions to the data. For example, documents can be clustered based on topic, authorship or writing style; images of human faces can be clustered based on illumination conditions, facial expressions or by person identity, and so forth.

Starting from a spectral method for feature selection, known as $Q - \alpha$, we introduce first a kernel version of the approach thereby adding the power of non-linearity to the underlying representations and the choice to emphasize certain kernel-dependent aspects of the data. As an alternative to the use of a kernel we introduce a principled manner for making use of auxiliary data within a spectral approach for handling situations where multiple subsets of relevant features exist in the data. The algorithm we will introduce allows for inhibition of relevant features of the auxiliary dataset and allows for creating a topological model of all relevant feature subsets in the dataset.

To evaluate the effectiveness of our approach we have conducted experiments both on real-images of human faces under varying illumination, facial expressions and person identity and on general machine learning tasks taken from the UC Irvine repository. The performance of our algorithm for selecting features with side information is generally superior to current methods we tested (PCA,OPCA,CPCA and SDR-SI).

1 Introduction

The problem of focusing on the most relevant measurements in a potentially overwhelming quantity of data is fundamental in machine vision and learning. Seeking out the relevant coordinates of a measurement vector is essential for making useful predictions as prediction accuracy drops significantly and training set size might grow exponentially with the growth of irrelevant features. To

add complexity to what already is non-trivial, natural data sets may contain multiple solutions, i.e., valid alternatives for relevant coordinate sets, depending on the task at hand. For example, documents can be analyzed based on topic, authorship or writing style; face images can be classified based on illumination conditions, facial expressions or by person identity; gene expressions levels can be clustered by pathologies or by correlations that also exist in other conditions.

The main running example that we will use in this paper is that of selecting features from an unlabeled (unsupervised) dataset consisting of human frontal faces where the desired features are relevant for inter-person variability. The face images we will use vary along four dimensions; (i) people identity, (ii) facial expressions, (iii) illumination conditions, and (iv) occlusions (see Fig. 1). One could possibly select relevant features for each of the three dimensions of relevance — *the challenge is how to perform the feature selection process on unlabeled data given that there are multiple solutions (in this case four different ones)?*

There are two principal ways to handle this problem. First is by embedding the feature selection algorithm into a higher dimensional space using a hand-crafted kernel function (the so called “kernel design” effort [11]). By selecting the right kernel function it may be possible to emphasize certain aspects of the data and de-emphasize others. Alternatively, the second approach is to introduce the notion of side information which is to provide auxiliary data in the form of an additional dataset which contains only the undesired dimensions of relevance. The feature selection process would then proceed by selecting features that enhance general dimensions of relevancy in the main dataset while inhibiting the dimensions of relevance in the auxiliary dataset.

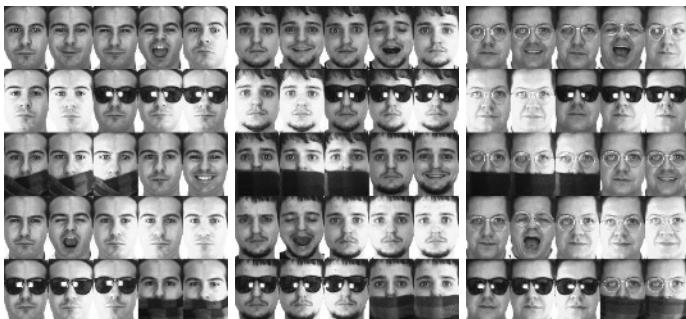


Fig. 1. 25 out of the 26 images in the AR dataset for three different persons. Images vary not only in person identity but also in illumination, facial expression, and amount and type of occlusion.

In this work we address both approaches. We start with the principle of spectral-based feature selection (introduced by [19]) and modify it to serve two new purposes: (i) endowing the approach with the power of kernel functions, satisfying the first approach for enriching the vector representation, and (ii) making use of auxiliary data for situations in which multiple subsets of relevant features exist in the data. The algorithm we will introduce allows for inhibition of relevant features of the auxiliary dataset and allows for creating a topological model of all relevant feature subsets in the dataset. The auxiliary dataset we consider could come in two different forms: the first being additional data points which represent undesired variability of the data, while the second form of side

data consists of pairs of points which belong to different classes of variability, i.e., are considered far away from each other in the space of selected coordinates.

Side information (a.k.a “irrelevant statistics” or “background information”) appears in various contexts in the literature — clustering [20,1,14] and continuous dimensionality reduction [15,5]. In this paper we address the use of side information in the context of a hard selection of a feature subset. Feature selection from unlabeled data differs from dimensionality reduction in that it only selects a handful of features which are “relevant” with respect to some inference task. Dimensionality reduction algorithms, for example PCA, generate a small number of features each of which is a combination of all of the original features. In many situations of interest, in visual analysis in particular but also in other application domains such as Genomics for instance, it is assumed that each process being studied involves only a limited number of features taken from a pool of a very large set of measurements. For this reason feature combination methods are not as desirable as methods that extract a small subset of features. The challenge in the selection process is to overcome the computational burden of pruning an exponential amount of feature subsets. The $Q - \alpha$ algorithm [19] which we propose using as a basis for our approach handles the exponential search space by harnessing the spectral information in such a manner where a computationally straightforward optimization guarantees a sparse solution, i.e., a selection of features rather than a combination of the original features.

In the subsection below we will describe the $Q - \alpha$ algorithm which forms the background for the work presented in this paper. In Section 2 we derive a kernel method version of the $Q - \alpha$ algorithm which enables the representation of high order cumulants among the entries of the feature vectors thereby considerably strengthening the feature selection methodology. In Section 3 we introduce the auxiliary data matrix as a side data and derive the optimization for selecting relevant features using the main dataset while inhibiting relevant features from the auxiliary dataset. In Section 4 we take the notion of auxiliary dataset a step further and form a complete topographical model of the relevant feature subsets. The general idea is based on rounds where the relevant features selected in previous rounds form “side” information for subsequent rounds. In this manner a hierarchical modeling the feature subsets becomes feasible and can be used for visualization and data modeling. In Section 5 we make use of another form of side information where the auxiliary data consists of pairs of points which belong to different classes of variability, i.e., are considered far away from each other in the space of selected coordinates. In Section 6 we evaluate the effectiveness of our algorithms by experiments on various datasets including real-image experiments on our main running example, and also running examples on general machine learning tasks taken from the UC Irvine repository.

1.1 Selecting Relevant Features with the $Q - \alpha$ Algorithm

The $Q - \alpha$ algorithm for unsupervised feature selection is based on the assumption that the selection of the relevant features (coordinates) will result in a coherent set of clusters formed by the input data points restricted to the selected

coordinates. The clustering score in this approach is measured indirectly. Rather than explicitly performing a clustering phase per feature selection candidates, one employs spectral information in order to measure the cluster arrangement coherency. Spectral algorithms have been proven to be successful in clustering [16], manifold learning or dimensionality reduction [12], approximation methods for NP-hard graph theoretical questions. In a nutshell, given a selection of features, the strength (magnitude) of the leading k eigenvalues of the affinity matrix constructed from the corresponding feature values across the sample data are directly related to the coherence of the cluster arrangement induced by the subset of selected features. The scheme is described as follows:

Let the data matrix be denoted by M . The feature values form the rows of M denoted by $\mathbf{m}_1^\top, \dots, \mathbf{m}_n^\top$ and normalized to unit norm $\|\mathbf{m}_i\| = 1$. Each row vector represents a feature (coordinate) sampled over the q trials. The column vectors of M represent the q samples (each sample is a vector in R^n). For example, a column can represent an image represented by its pixel values and a row can represent a specific pixel location whose value runs over the q images. As mentioned in the previous section, our goal is to select rows (features) from M such that the corresponding candidate data matrix (containing only the selected rows) consists of columns that are coherently clustered in k groups. The value of k is user dependent and is specific to the task at hand. The challenge in this approach is to avoid the exponential number of row selections and preferably avoid explicitly clustering the columns of the data matrix per each selection.

Mathematically, to obtain a clustering coherency score we compute the "affinity" matrix of the candidate data matrix defined as follows. Let $\alpha_i \in \{0, 1\}$ be the indicator value associated with the i 'th feature, i.e., $\alpha_i = 1$ if the i 'th feature is selected and zero otherwise. Let A_α be the corresponding affinity matrix whose (i, j) entries are the inner-product (correlation) between the i 'th and j 'th columns of the resulting candidate data matrix: $A_\alpha = \sum_{i=1}^n \alpha_i \mathbf{m}_i \mathbf{m}_i^\top$ (sum of rank-1 matrices). From algebraic graph theory, if the columns of the candidate data matrix are coherently grouped into k clusters, we should expect the leading k eigenvalues of A_α to be of high magnitude [8,10,2,16]. The resulting scheme should therefore be to maximize the sum of eigenvalues of the candidate data matrix over all possible settings of the indicator variables α_i .

What is done in practice, in order to avoid the exponential growth of assigning binary values to n indicator variables, is to allow α_i to receive real values in an unconstrained manner. A least-squares energy function over the variables α_i is formed and its optimal value is sought after. What makes this approach different from the "garden variety" soft-decision-type algorithms is that this particular setup of optimizing over spectral properties guarantees that the α_i *always come out positive and sparse* over all local maxima of the energy function. This property is intrinsic rather than being the result of explicit constraints in the form of regularizers, priors or inequality constraints. We optimize the following:

$$\max_{Q, \alpha_i} \text{trace}(Q^\top A_\alpha^\top A_\alpha Q) \quad \text{subject to} \quad \sum_{i=1}^n \alpha_i^2 = 1, \quad Q^\top Q = I \quad (1)$$

Note that the matrix Q holds the first k eigenvectors of A_α and that $\text{trace}(Q^\top A_\alpha^\top A_\alpha Q)$ is equal to the sum of squares of the leading k eigenvalues: $\sum_{j=1}^k \lambda_j^2$. A local maximum of the energy function is achieved by interleaving the “orthogonal iteration” scheme [6] within the computation of α as follows:

Definition 1 ($Q - \alpha$ Method). Let M be an $n \times q$ input matrix with rows $\mathbf{m}_1^\top, \dots, \mathbf{m}_n^\top$, and some orthonormal $q \times k$ matrix $Q^{(0)}$, i.e., $Q^{(0)^\top} Q^{(0)} = I$. Perform the following steps through a cycle of iterations with index $r = 1, 2, \dots$

1. Let $G^{(r)}$ be a matrix whose (i, j) components are

$$(\mathbf{m}_i^\top \mathbf{m}_j) \mathbf{m}_i^\top Q^{(r-1)^\top} Q^{(r-1)} \mathbf{m}_j.$$

2. Let $\alpha^{(r)}$ be the leading eigenvector of $G^{(r)}$.
3. Let $A^{(r)} = \sum_{i=1}^n \alpha_i^{(r)} \mathbf{m}_i \mathbf{m}_i^\top$.
4. Let $Z^{(r)} = A^{(r)} Q^{(r-1)}$.
5. $Z^{(r)} \xrightarrow{QR} Q^{(r)} R^{(r)}$, that is, $Q^{(r)}$ is determined by the “QR” factorization of $Z^{(r)}$.
6. Increment index r and go to step 1.

Note that steps 4,5 of the algorithm consist of the “orthogonal iteration” module, i.e., if we were to repeat steps 4,5 *only* we would converge onto the eigenvectors of $A^{(r)}$. However, the algorithm does not repeat steps 4,5 in isolation and instead recomputes the weight vector α (steps 1,2,3) before applying another cycle of steps 4,5.

The algorithm would be meaningful provided that three conditions are met:

1. the algorithm converges to a local maximum,
2. at the local maximum $\alpha_i \geq 0$ (because negative weights are not admissible), and
3. the weight vector α is *sparse* (because without it the soft decision does not easily translate into a hard gene selection).

Conditions (2) and (3) are not readily apparent in the formulation of the algorithm (the energy function lacks the explicit inequality constraint $\alpha_i \geq 0$ and an explicit term to “encourage” sparse solutions) but are nevertheless satisfied. The key for having sparse and non-negative (same sign) weights is buried in the matrix G (step 1). Generally, the entries of G are not necessarily positive (otherwise α would have been non-negative due to the Perron-Frobenius theorem) — nevertheless due its makeup it can be shown that in a probabilistic manner the leading eigenvector of G is positive with probability $1 - o(1)$. In other words, as the number of features n grows larger the chances that the leading eigenvector of G is positive increases rapidly to unity. The details of why the makeup of G induces such a property, the convergence proof and the proof of the “Probabilistic Perron-Frobenius” claim can be found in [19].

Finally, it is worth noting that the scheme can be extended to handle the supervised situation (when class labels are provided); that the scheme can be applied also to the Laplacian affinity matrix; and that the scheme readily applies when the spectral gap $\sum_{i=1}^k \lambda_i^2 - \sum_{j=k+1}^q \lambda_j^2$ is maximized rather than $\sum_{i=1}^k \lambda_i^2$ alone. Details can be found in [19].

2 Representing Higher-Order Cumulants Using Kernel Methods

The information on which the $Q - \alpha$ method relies on to select features is contained in the matrix G . Recall that the criterion function underlying the $Q - \alpha$ algorithm is a sum over all pairwise feature vector relations:

$$\text{trace}(Q^\top A_\alpha^\top A_\alpha Q) = \alpha^\top G \alpha,$$

where G is defined such that $G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j) \mathbf{m}_i^\top Q Q^\top \mathbf{m}_j$. It is apparent that feature vectors interact in pairs and the interaction is *bilinear*. Consequently, cumulants of the original data matrix M which are of higher order than two are not being considered by the feature selection scheme. For example, if M were to be decorrelated (i.e., $M M^\top$ is diagonal) the matrix G would be diagonal and the feature selection scheme would select only a single feature.

In this section we employ the "kernel trick" to include cumulants of higher orders among the feature vectors in the feature selection process. This serves two purposes: On one hand the representation is enriched with non-linearities induced by the kernel, and on the other hand, given a successful choice of a kernel (so called Kernel Design effort [11]) one could possibly emphasize certain desirable aspects of the data while inhibiting others.

Kernel methods in general have been attracting much attention in the machine learning literature — initially with the support vector machines [13] and later took a life of their own (see [11]). Mathematically, the kernel approach is defined as follows: let $\mathbf{x}_1, \dots, \mathbf{x}_l$ be vectors in the input space, say R^q , and consider a mapping $\phi(\mathbf{x}) : R^q \rightarrow \mathcal{F}$ where \mathcal{F} is an inner-product space. The kernel-trick is to calculate the inner-product in \mathcal{F} using a kernel function $k : R^q \times R^q \rightarrow R$, $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$, while avoiding explicit mappings (evaluation of) $\phi()$. Common choices of kernel selection include the d 'th order polynomial kernels $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + c)^d$ and the Gaussian RBF kernels $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}_j\|^2)$. If an algorithm can be restated such that the input vectors appear in terms of inner-products only, one can substitute the inner-products by such a kernel function. The resulting kernel algorithm can be interpreted as running the original algorithm on the space \mathcal{F} of mapped objects $\phi(\mathbf{x})$. Kernel methods have been applied to the support vector machine (SVM), principal component analysis (PCA), ridge regression, canonical correlation analysis (CCA), QR factorization and the list goes on. We will focus below on deriving a kernel method for the $Q - \alpha$ algorithm.

2.1 Kernel $Q - \alpha$

We will consider mapping the rows \mathbf{m}_i^\top of the data matrix M such that the rows of the mapped data matrix become $\phi(\mathbf{m}_1)^\top, \dots, \phi(\mathbf{m}_n)^\top$. Since the entries of G consist of inner-products between pairs of mapped feature vectors, the interaction will be no longer bilinear and will contain higher-order cumulants whose nature depends on the choice of the kernel function.

Replacing the rows of M with their mapped version introduces some challenges before we could apply the kernel trick. The affinity matrix $A_\alpha = \sum_i \alpha_i \phi(\mathbf{m}_i)\phi(\mathbf{m}_i)^\top$ cannot be explicitly evaluated because A_α is defined by *outer-products* rather than inner-products of the mapped feature vectors $\phi(\mathbf{m}_i)$. The matrix Q holding the eigenvectors of A_α cannot be explicitly evaluated as well and likewise the matrix $Z = A_\alpha Q$ (in step 4). As a result, kernelizing the $Q - \alpha$ algorithm requires one to represent α without explicitly representing A_α and Q both of which were instrumental in the original algorithm. Moreover, the introduction of the kernel should be done in such a manner to preserve the key property of the original $Q - \alpha$ algorithm of producing a sparse solution.

Let $V = MM^\top$ be the $n \times n$ matrix whose entries are evaluated using the kernel $v_{ij} = k(\mathbf{m}_i, \mathbf{m}_j)$. Let $Q = M^\top E$ for some $n \times k$ (recall k being the number of clusters in the data) matrix E . Let $D_\alpha = \text{diag}(\alpha_1, \dots, \alpha_n)$ and thus $A_\alpha = M^\top D_\alpha M$ and $Z = A_\alpha Q = M^\top D_\alpha V E$. The matrix Z cannot be explicitly evaluated but $Z^\top Z = E^\top V D_\alpha V D_\alpha V E$ can be evaluated. The matrix G can be expressed with regard to E instead of Q :

$$\begin{aligned} G_{ij} &= (\phi(\mathbf{m}_i)^\top \phi(\mathbf{m}_j)) \phi(\mathbf{m}_i)^\top Q Q^\top \phi(\mathbf{m}_j) \\ &= k(\mathbf{m}_i, \mathbf{m}_j) \phi(\mathbf{m}_i)^\top (M^\top E) (M^\top E)^\top \phi(\mathbf{m}_j) \\ &= k(\mathbf{m}_i, \mathbf{m}_j) \mathbf{v}_i^\top E E^\top \mathbf{v}_j \end{aligned}$$

where $\mathbf{v}_1, \dots, \mathbf{v}_n$ are the columns of V . Step 5 of the $Q - \alpha$ algorithm consists of a QR factorization of Z . Although Z is uncomputable it is possible to compute R and R^{-1} directly from the entries of $Z^\top Z$ without computing Q using the Kernel Gram-Schmidt described in [18]. Since $Q = ZR^{-1} = M^\top D_\alpha V E R^{-1}$ the update step is simply to replace E with ER^{-1} and start the cycle again. In other words, rather than updating Q we update E and from E we obtain G and from there the newly updated α . The kernel $Q - \alpha$ is summarized below:

Definition 2 (Kernel $Q - \alpha$). *Let M be an uncomputable matrix with rows $\phi(\mathbf{m}_1)^\top, \dots, \phi(\mathbf{m}_n)^\top$. The kernel function is given by $\phi(\mathbf{m}_i)^\top \phi(\mathbf{m}_j) = k(\mathbf{m}_i, \mathbf{m}_j)$. The matrix $V = MM^\top$ is a computable $n \times n$ matrix. Let $E^{(0)}$ be an $n \times k$ matrix selected such that $M^\top E^{(0)}$ has orthonormal columns. Iterate over the steps below, with the index $r = 1, 2, \dots$*

1. Let $G^{(r)}$ be a $n \times n$ matrix whose (i, j) components are

$$k(\mathbf{m}_i, \mathbf{m}_j) \mathbf{v}_i^\top E^{(r-1)} E^{(r-1)^\top} \mathbf{v}_j.$$

2. Let $\alpha^{(r)}$ be the largest eigenvector of $G^{(r)}$, and let $D^{(r)} = \text{diag}(\alpha_1^{(r)}, \dots, \alpha_n^{(r)})$.
3. Let $Z^{(r)}$ be an uncomputable matrix

$$Z^{(r)} = (M^\top D^{(r)} M) (M^\top E^{(r-1)}) = M^\top D^{(r)} V E^{(r-1)}.$$

4. $Z^{(r)} \xrightarrow{QR} QR$. It is possible to compute directly R, R^{-1} from the entries of the computable matrix $Z^{(r)^\top} Z^{(r)}$ without explicitly computing the matrix Q (see [18]).
5. Let $E^{(r)} = E^{(r-1)} R^{-1}$.
6. Increment index r and go to step 1.

The result of the algorithm is the weight vector α and the design matrix G which contains all the data about the features. The drawback of the kernel approach for handling multiple structures of the data is that the successful choice of a kernel depends on the user and is largely an open problem. For example, with regard to our main running example it is unclear which kernel to choose that will strengthen the clusters induced by inter-personal variation and inhibit the clusters induced by lighting facial expressions. We therefore move our attention to the alternative approach using the notion of side data.

3 $Q - \alpha$ with Side Information

Consider the $n \times q$ data matrix M defined above as the “main” data. We are given an auxiliary $n \times p$ data matrix W with rows $\mathbf{w}_1^\top, \dots, \mathbf{w}_n^\top$ representing p data points comprising the “side” information. Our goal is to select a subset of coordinates, namely, determine the weight vector α such the affinity matrix $\sum_i \alpha_i \mathbf{m}_i \mathbf{m}_i^\top$ has coherent k clusters (measured by the sum of squares of the first k eigenvalues) whereas $\sum_i \alpha_i \mathbf{w}_i \mathbf{w}_i^\top$ has low cluster coherence. The desire for low cluster coherence for the side information can be represented by small variance of each coordinate value along the p samples. Namely, if \mathbf{m}_i is selected as a relevant feature of the main data, we should expect that the corresponding side feature vector \mathbf{w}_i will have a small variance. Small variance of the selected rows of W means that the corresponding affinity matrix $\sum_i \alpha_i \mathbf{w}_i \mathbf{w}_i^\top$ represents a single cluster (whether coherent or not is immaterial).

To clarify the logic behind our approach, consider the scenario presented in [5]. Assume we are given face images of 5 individuals covering variability of illumination and facial expressions — a total of 26 images per individual. The main data matrix M will contain therefore 130 columns. We wish to select relevant features (rows of M), however, there are three dimensions of relevancy: (i) person identity, (ii) illumination direction, and (iii) facial expressions. One could possibly select relevant features for each dimension of relevance and obtain a coherent clustering in that dimension. Say we are interested in the person identity dimension of relevance. In that case the auxiliary matrix W will contain 26 images of a 6th individual (covering facial expressions and illumination conditions). Features selected along the dimensions of facial expression or illumination will induce coherent clusters in the side data, whereas features selected along the person identity dimension will induce a single cluster (or no structure at all) in the side data — and low variance of the feature vectors is indicative to single cluster or no structure at all. In formal notations we have the following:

Let $D = \text{diag}(\text{var}(\mathbf{w}_1^\top), \dots, \text{var}(\mathbf{w}_n^\top))$ be a diagonal matrix with the variance of the rows of W . The low coherence desire over the side data translates to minimization of $\alpha^\top D \alpha$. Taken together, we have a Rayleigh quotient type of energy function to maximize:

$$\max_{Q, \alpha_i} \frac{\text{trace}(Q^\top A_\alpha^\top A_\alpha Q)}{\alpha^\top (D + \lambda I) \alpha} = \frac{\alpha^\top G \alpha}{\alpha^\top (D + \lambda I) \alpha} \quad (2)$$

$$\text{subject to} \quad \sum_{i=1}^n \alpha_i^2 = 1, \quad Q^\top Q = I$$

where G is the matrix defined above whose entries are: $G_{ij} = (\mathbf{m}_i^\top \mathbf{m}_j) \mathbf{m}_i^\top Q Q^\top \mathbf{m}_j$. The scalar $\lambda \geq 0$ is user-setable with the purpose of providing the tradeoff between the main data and the side data. Large values of λ translates to low weight for the side information in the feature selection scheme. A vanishing value $\lambda = 0$ is admissible provided that none of the variances vanishes (D has no vanishing entries along its diagonal) — in that case equal weight is given to the two sources of data. The $Q - \alpha$ with side information algorithm becomes:

Definition 3 ($Q - \alpha$ with Side Information). Let M be an $n \times q$ input matrix with rows $\mathbf{m}_1^\top, \dots, \mathbf{m}_n^\top$, W be an $n \times p$ “side” matrix where the variance of its rows form a diagonal matrix D , and $Q^{(0)}$ is some orthonormal $q \times k$ matrix, i.e., $Q^{(0)\top} Q^{(0)} = I$. Perform the following steps through a cycle of iterations with index $r = 1, 2, \dots$

1. Let $G^{(r)}$ be a matrix whose (i, j) components are $(\mathbf{m}_i^\top \mathbf{m}_j) \mathbf{m}_i^\top Q^{(r-1)\top} Q^{(r-1)} \mathbf{m}_j$.
2. Let $\alpha^{(r)}$ be the largest eigenvector of $(D + \lambda I)^{-1} G^{(r)}$.
3. Let $A^{(r)} = \sum_{i=1}^n \alpha_i^{(r)} \mathbf{m}_i \mathbf{m}_i^\top$.
4. Let $Z^{(r)} = A^{(r)} Q^{(r-1)}$.
5. $Z^{(r)} \xrightarrow{QR} Q^{(r)} R^{(r)}$.
6. Increment index r and go to step 1.

Note the change in step 2 compared to the $Q - \alpha$ algorithm. Since $D + \lambda I$ is a diagonal positive matrix, its inverse is also positive therefore the positivity of G is not affected. In other words, the properties of G which induce a positive (and sparse) solution for the weight vector α (see [19]) are not negatively affected when G is multiplied with a positive diagonal matrix. If D were not diagonal, then D^{-1} would not have been positive and the optimized α values would not come out positive and sparse.

4 Topographical Model of All Relevant Feature Subsets

We can further extend the notion of “negative variability” embedded in the side information to a wider perspective of representing a hierarchy of feature subsets extracted iteratively. The general idea is to treat the weight vector α (which determines the feature selection as it is a sparse positive vector) as representing axes of negative variability for subsequent rounds. Let α be the feature selection solution given by running the $Q - \alpha$ algorithm. We wish to run $Q - \alpha$ again while looking for an alternative solution along a different dimension of variability. We construct a “side information” matrix D whose diagonal is $D = \text{diag}(\alpha_1^2, \dots, \alpha_n^2)$ and run the $Q - \alpha$ -with-SI algorithm. The new weight vector α' will be encouraged to have high values in coordinates where α has low values. This is applied

iteratively where in each round $Q - \alpha$ -with-SI is executed with the matrix D containing the sum of square α values summed over all previous rounds.

Furthermore, the G matrix resulting from each round of the above scheme can be used for generating a coordinization of the features as a function of the implicit clustering of the (projected) data. The weight vector α is the largest eigenvector of G , but as in Multi-Dimensional-Scaling (MDS), the first largest eigenvectors of G form a coordinate frame. Assume we wish to represent the selected features by a 1D coordinate. This can be achieved by taking the first two largest eigenvectors of G thereby each feature is represented by two coordinates. A 1D representation is made by normalizing the coordinate-pair (i.e., each feature is represented by a direction in the 2D MDS frame). Given r rounds, each feature is represented by r coordinates which can be used for visualization and data modeling.

An example of such a topographical map is shown in figure 2. The data matrix consists of 150 data points each described by 20 features out of which 9 are relevant. The relevant features form two possible solution sets where each solution induces three clusters of data points. The first set consists of three features marked by “1,2,3”, while the second set consists of three different features marked by “A,B,C”. Three additional features marked by “1A,2B,3C” were constructed by summing the corresponding feature vectors 1,2,3 and A,B,C, respectively. The remaining 11 (irrelevant) features were constructed by randomly permuting the entries of the first feature vector. We ran $Q - \alpha$ twice creating for each feature two coordinates (one per each run) as described above. In addition to the coordinization of each feature we have associated the corresponding α value as a measure of “relevancy” of the feature per solution. Taken together, each feature is represented by a position in the 2D topographical map and a 2D magnitude represented by an ellipse whose major axes capture the respective α values. The horizontal axis in Fig. 2(b) is associated with the solution set of features “1,2,3” and the vertical axis with the solution set “A,B,C”. We see that the hybrid features 1A,2B,3C, which are relevant to both cluster configurations, have equal (high) relevancy in both sets (large circles in the topographical map).

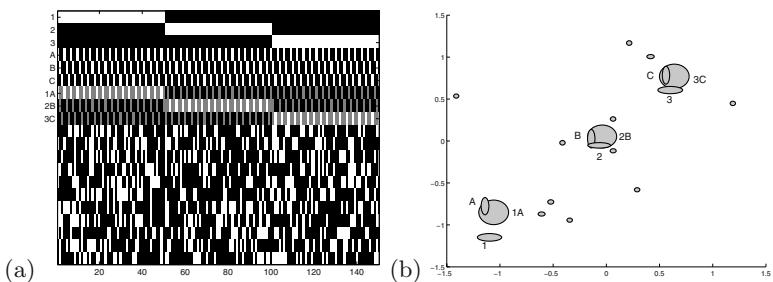


Fig. 2. (a) A synthetic dataset used to demonstrate the creation of a topographical model of the features (b) The resulting topographical model (see text).

5 Pairwise Side Information

Another possible variant of $Q - \alpha$ -SI is when the side information is given over pairs of “negative” data points. Consider the (adapted) problem raised by [20] in the context of distance metric learning for clustering: we are given a set of data points forming a data matrix M (the “main” data). As side information we are given pairs of points $\mathbf{x}_i, \mathbf{x}_j$ which are *known* to be part of different clusters. We wish to select features (coordinates) such that the main data contains maximally coherent clusters while obeying the side information (i.e., features are selected such that for each of the “side” pairs $(\mathbf{x}_i^\top \mathbf{x}_j)^2$ is small).

We can incorporate the side information by constructing a side matrix B which functions similarly to the diagonal matrix D we constructed in the previous sections. The difference here would be that B is not diagonal and therefore needs to be handled differently. Consider a pair of side points \mathbf{x}, \mathbf{y} . We wish to find the weight vector α such that: $(\mathbf{x}^\top \mathbf{y})^2 = (\sum_i \alpha_i x_i y_i)^2 = \alpha^\top F \alpha$ is small, where $F_{rs} = x_r y_r x_s y_s$. Denote by F^{ij} the matrix corresponding to the pair of side points $\mathbf{x}_i, \mathbf{x}_j$ and let $B = \sum_i \sum_j F^{ij}$.

Our goal is to maximize the spectral information coming from the main data (as before) while minimizing $\alpha^\top B \alpha$. We are back to the same framework as in Sections 3 and 4 with the difference that B is not diagonal therefore the product $B^{-1}G$ is not guaranteed to obey the properties necessary for the weight vector α to come out positive and sparse. Instead, we define an additive energy function:

$$\begin{aligned} & \max_{Q, \alpha_i} \text{trace}(Q^\top A_\alpha^\top A_\alpha Q) - \lambda \alpha^\top B \alpha \\ & \text{subject to} \quad \sum_{i=1}^n \alpha_i^2 = 1, \quad Q^\top Q = I \end{aligned} \tag{3}$$

This energy function is equivalent to $\alpha^\top (G - \lambda B) \alpha$ where λ tradeoffs the weight given to the side data. The algorithm follows the steps of the $Q - \alpha$ algorithm with the difference in step 2: “ $\alpha^{(r)}$ is the largest eigenvector of $G^{(r)} - \lambda B$.”

There is an open issue of showing that α comes out positive and sparse. The matrix G is “dominantly positive”, i.e., when treated as a random matrix each entry has a positive mean and thus it can be shown that the probability of a positive α asymptotes at unity very fast with n [19]. The question what happens to the mean when one subtracts λB from G . Our working assumption is that the entries of B are significantly smaller than the corresponding entries of G because the inner-products of the side points should be small — otherwise they wouldn’t have been supplied as side points. Empirical studies on this algorithm validate this assumption and indeed α maintains the positivity and sparsity properties in our experiments.

6 Experiments

We present below three types of experiments (i) simulations on synthetic data for the purpose of studying the effects of different weightings of the side data,

(ii) our main running example on the AR face dataset, and (iii) various examples taken from the UC Irvine repository of data sets. Due to space constraints the synthetic simulations are given only in the technical report [17].

Face images. Our main running example is the selection of features from an unlabeled data set of face images taken from the AR dataset [7]. The dataset consists of 100 people with 26 images per person varying according to lighting direction and facial expressions. Our task is to select those features which are relevant for distinguishing between people identities only. The dataset contains three dimensions of relevancy, and the use of side data is crucial for inhibiting the unwanted dimensions of facial expressions and lighting variations. Following [5] we adopted the setting where the main data set contained the images of 5 randomly chosen men (out of the 50 men) totaling 130 images. The side dataset consisted of the 26 images of a random sixth man. The feature selection process $Q - \alpha - SI$ looks for coordinates which maximize the cluster coherence of the main dataset while minimizing the variance of the coordinate vectors of the side data. As a result, the selected coordinates are relevant for separating among person identities while being invariant to the other dimensions of variability. The task of clustering those images into the five *correct* clusters is hard since the nuisance structures (such as those generated by variation of lighting and facial expressions) are far more dominant than the structure of person variability.

The feature values we use as a representation of the image is designed to capture the relationship between average intensities of neighboring regions. This suggests the use of a family of basis functions, like the Haar wavelets, which encode such relationships along different orientations (see [9,4]). In our implementation the Haar wavelet transform is run over an image and results in a set of 5227 coefficients at several scales that indicate the response of the wavelets over the entire image. Many of the coefficients are irrelevant for the task of separating between facial identities and it is therefore the goal of the $Q - \alpha - SI$ to find those coefficients that represent the relevant regions.

To quantify the performance of our algorithm in a comparative manner we used the normalized precision score introduced in [5,15] which measures the average purity of the k-Nearest Neighbors for varying values of k . We compared the performance to four methods: PCA which is the most popular technique for dimensionality reduction, Constrained PCA (CPCA) and Oriented PCA (OPCA) [3], and Sufficient Dimensionality Reduction with Side Information (SDR-SI) [5]. All but the first method (PCA) utilize the same side data as the $Q - \alpha - SI$. Also worth noting that all the methods we compared to extract features by combinations of the original features rather than just select features.

Optimal parameters (dimensionality and λ) for all methods were chosen to maximize the precision index for a training set. The wavelet decomposition was not optimal for the other methods and therefore the raw image intensities were used instead. Reported results were obtained on a separate test set. The entire procedure was repeated 20 times on randomly chosen subsets of the AR database.

Fig. 3a shows the results averaged over 20 runs. The precision index is normalized between 0 to 1 where 0 is obtained with random neighboring and 1

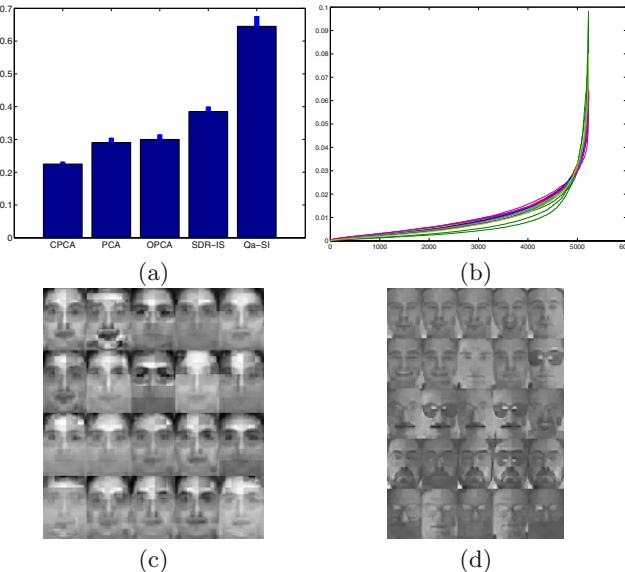


Fig. 3. (a) Comparison of the normalized precision index between CPCCA, PCA, OPCA, SDR-IS, and $Q\alpha - SI$ on the AR dataset. (b) Sorted feature weights (α values) for each of the 20 runs showing the sparsity of the feature selection (c) The average image of all men in the AR dataset projected to the selected features for each one of the 20 runs. (d) For a specific run: each row contains the images of one person projected onto the selected feature space.

when all nearest neighbors are of the same class. Note that the precision index of $Q - \alpha - SI$ is 0.64 which is significantly higher than 0.39 obtained by the next best method (SDR-SI). Fig. 3(b) shows the resulting α values sorted separately at each one of the 20 runs. As can be seen those values are extremely sparse - having only few of the feature weights above a very clear threshold at each run.

Fig. 3(c) illustrates the selected features by the $Q - \alpha - SI$ at each run. This is done by synthesizing (reconstructing) the images from their wavelet coefficients weighted by the α values. What is shown per run is the average male image. Fig. 3(d) shows the projection of random faces from a specific run to the weighted features space. Each row contains images of one person. In both figures (c,d) some characteristic features of each individual (beard, dark glasses frame, distinctive hair line) are highlighted, while the illumination differences are reduced.

Finally, it is worth noting that our attempts to find an appropriate kernel which will perform as well as the side data approach were unsuccessful. Our experiments show that the kernel $Q - \alpha$ has significant advantages over $Q - \alpha$ in general, but selecting an appropriate kernel for the multiple structure paradigm is a hard problem and is left open (see [11] for work on kernel design).

UC Irvine Repository Tests. We also applied our method to several datasets from the UC Irvine repository. On each dataset we applied k-means clustering on the raw data and on features provided by PCA, OPCA and CPCCA.

An accuracy score was computed for each clustering result similarly to what was done in [20]. The results are shown for the dermatology, segmentation, wine and ecoli datasets. We also tested the algorithm on the glass, Boston-housing and arrhythmia datasets where none of the algorithms were significantly better than chance. The results are summarized in the table below. Each report result is an average of several experiments where, at turns, each class served as side information and the other classes were taken to be the main dataset. The features were weighted, combined or selected according to the algorithm in question, and then the data points were clustered by k-means. Each result shown in the table was averaged over 20 runs. The number of features used for each PCA variants was the one which gave the best average accuracy. The parameter λ used in the $Q - \alpha$ with side information was fixed at $\lambda = 0.1$.

Dataset	raw data	$Q - \alpha$ SI	PCA	CPCA	OPCA
dermatology	0.5197	0.8816	0.5197	0.6074	0.8050
ecoli	0.6889	0.7059	0.6953	0.6973	0.5620
segmentation	0.7157	0.7817	0.7208	0.7089	0.7110
wine	0.7280	0.9635	0.7280	0.7280	0.9493

The $Q - \alpha$ -SI performed the best over all the experiments we conducted. In some of the datasets constrained PCA or oriented PCA performed only slightly worse, but none of these methods gave good results consistently in all four datasets. Unlike *PCA* and its variants, the $Q - \alpha$ algorithm tends to produce a sparse selection of features, showing a large preference toward a small number of features. For example, in the wine dataset the α values corresponding to the features Alcohol and Proline were three times larger than the rest.

References

1. G. Chechik and N. Tishby. Extracting relevant structures with side information. In *NIPS 2002*.
2. F.R.K. Chung. *Spectral Graph Theory*. AMS, 1998.
3. K.I. Diamantaras and S.Y. Kung *Principal Component Neural Networks: Theory and Applications* NY: Wiley, 1996.
4. C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. In *SIGGRAPH 1995*
5. A. Globerson, G. Chechik, and N. Tishby. Sufficient dimensionality reduction with irrelevance statistics. In *UAI-2003*.
6. G.H. Golub and C.F. Van Loan. *Matrix computations*. , 1989.
7. A.M. Martinez and R. Benavente. The AR face database. Tech. Rep. 24, CVC, 1998.
8. T.S. Motzkin and E.G. Straus. Maxima for graphs and a new proof of a theorem by turan. *Canadian Journal of Math.*, 17:533–540, 1965.
9. M. Oren, C. Papageorgiou, P. Sinha, E. Osuna and T. Poggio. Pedestrian Detection Using Wavelet Templates. In *CVPR 1997*.
10. M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *CVPR*, 2003.
11. B. Scholkopf and A.J. Smola. *Learning with Kernels* The MIT press, 2002.

12. Joshua B. Tenenbaum A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, (2000).
13. V. N. Vapnik. *The nature of statistical learning*. Springer, 2nd edition, 1998.
14. K. Wagstaff, A. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In *ICML-2001*.
15. D. Weinshall, N. Shental, T. Hertz, and M. Pavel. Adjustment learning and relevant component analysis. In *ECCV*, 2002.
16. A.Y. Ng, M.I. Jordan and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. *NIPS*, 2001.
17. A. Shashua amd L. Wolf Sprse Spectral-based Feature Selection with Side Information. TR 2003-57, Leibniz Center for Research, HUJI, 2003.
18. L. Wolf and A. Shashua. Kernel principal angles for classification machines with applications to image sequence interpretation. In *CVPR*, 2003.
19. L. Wolf and A. Shashua. Direct feature selection with implicit inference. *ICCV'03*.
20. E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russel. Distance metric learning, with applications to clustering with side information. In *NIPS 2002*.

Tracking Articulated Motion Using a Mixture of Autoregressive Models

Ankur Agarwal and Bill Triggs

GRAVIR-INRIA-CNRS, 655 Avenue de l'Europe, Montbonnot 38330, France

{Ankur.Agarwal,Bill.Triggs}@inrialpes.fr

<http://www.inrialpes.fr/lear/people/{agarwal,triggs}>

Abstract. We present a novel approach to modelling the non-linear and time-varying dynamics of human motion, using statistical methods to capture the characteristic motion patterns that exist in typical human activities. Our method is based on automatically clustering the body pose space into connected regions exhibiting similar dynamical characteristics, modelling the dynamics in each region as a Gaussian autoregressive process. Activities that would require large numbers of exemplars in example based methods are covered by comparatively few motion models. Different regions correspond roughly to different action-fragments and our class inference scheme allows for smooth transitions between these, thus making it useful for activity recognition tasks. The method is used to track activities including walking, running, *etc.*, using a planar 2D body model. Its effectiveness is demonstrated by its success in tracking complicated motions like turns, without any key frames or 3D information.

1 Introduction

Tracking and analyzing human motion in video sequences is a key requirement in several applications. There are two main levels of analysis: (*i*) detecting people and tracking their image locations; and (*ii*) estimating their detailed body pose, *e.g.* for motion capture, action recognition or human-machine-interaction. The two levels interact, as accurate detection and tracking requires prior knowledge of pose and appearance, and pose estimation requires reliable tracking. Using an explicit body model allows the state of the tracker to be represented as a vector of interpretable pose parameters, but the problem is non-trivial owing to the great flexibility of the human body, which requires the modelling of many degrees of freedom, and the frequent non-observability of many of these degrees of freedom in monocular sequences owing to self-occlusions and depth ambiguities. In fact, if full 3D pose is required from monocular images, there are potentially thousands of local minima owing to kinematic flipping ambiguities [18]. Even without this, pervasive image ambiguities, shadows and loose clothing add to the difficulties.

Previous work: Human body motion work divides roughly into *tracking based approaches*, which involve propagating the pose estimate from one time step to another, and *detection based approaches*, which estimate pose from the current image(s) alone. The latter have become popular recently in the form of ‘exemplars’ [21] and ‘key frames’ [19]. These methods allow the direct use of image data, which eliminates the need for

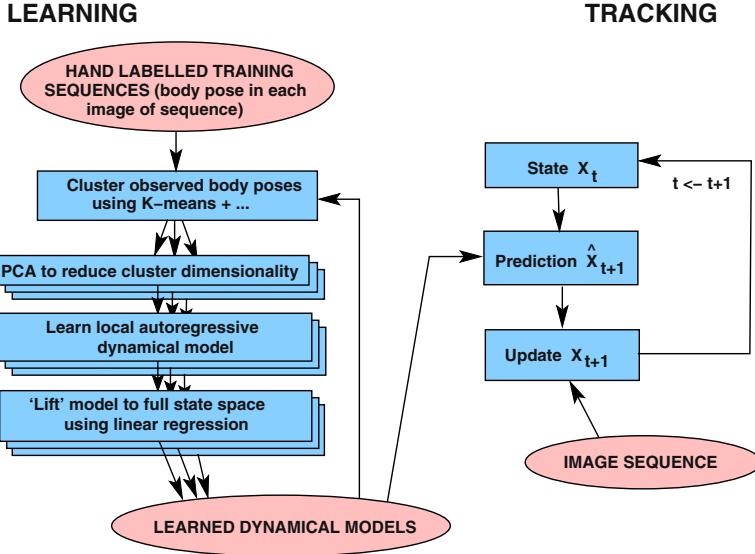


Fig. 1. Overview of the learning and tracking components of our algorithm (see text).

predefined parametric models. But the interpretability of parametric models is lost, and large numbers of exemplars are needed to cover high dimensional spaces such as those of human poses. (Tree-based structures have recently been explored for organizing these datasets [20], but they rely on the existence of accurate distance metrics in the appearance space).

Within the tracking framework, many methods are based on computing optical flow [9,3,2], while others optimize over static images (*e.g.* [18]). On the representation side, a variety of 2D and 3D parametric models have been used [9,3,16,18], as well as non-parametric representations based on motion [4] or appearance [15,11,21]. A few learning based methods have modelled dynamics [8,17,14], motion patterns from motion capture data (*e.g.* [1]), and image features [16,7,6]. To track body pose, Howe *et al* [8] and Sidenbladh *et al* [17] propose plausible next states by recovering similar training examples, while Pavlovic *et al* [14] learn a weak dynamical model over a simplified 8-parameter body for fronto-parallel motions. We extend the learning based approach by modelling complex high dimensional motions within reduced manifolds in an unsupervised setting. In the past, nonlinear motion models have been created by combining Hidden Markov Models and Linear Dynamical Systems in the multi-class dynamics framework, *e.g.* in [13,14]. However, this approach artificially decouples the switching dynamics from the continuous dynamics. We propose a simpler alternative that avoids this decoupling, discussing our philosophy in section 3.4.

Problem formulation: We use a tracking based approach, representing human motions in terms of a fixed parametric body model controlled by pose-related parameters, and focusing on flexible methods for learning the human dynamics. We specialize to monocular sequences using a 2D (image based) body model, but our methods extend immediately to the 3D and multicamera cases. Our main aim is to study how relationships and con-

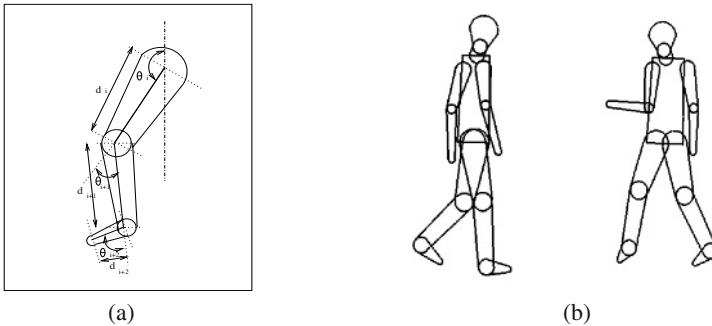


Fig. 2. (a) Human pose parametrization in the Scaled Prismatic Model. (b) Examples of different poses of the complete SPM. Each limb segment is overlayed with its corresponding template shape.

straints in parameter space can be learned automatically from sample trajectories, and how this information can be exploited for tracking. Issues to be handled include the ‘curse of dimensionality’, complex nonlinear motions, and transitions between different parts of the space.

Overview of approach: Our approach is based on learning dynamical models from sample trajectories. We learn a collection of local motion models (Gaussian autoregressive processes) by automatically partitioning the parameter space into regions with similar dynamical characteristics. The mixture of dynamical models is built from a set of hand-labelled training sequences as follows: (i) the state vectors are clustered using K-means and projected to a lower dimensional space using PCA to stabilize the subsequent estimation process; (ii) a local linear autoregression for the state given the p previous reduced states is learned for each cluster ($p = 1, 2$ in practice); (iii) the data is reclustered using a criterion that takes into account the accuracy of the local model for the given point, as well as the spatial contiguity of points in each model; (iv) the models are refitted to the new clusters, and the process is iterated to convergence.

We sidestep the difficult depth estimation problem by using a purely 2D approach, so our dynamical models are view dependent. Our tracking framework is similar to Covariance Scaled Sampling [18]: well-shaped random sampling followed by local optimization of image likelihood. Figure 1 illustrates the basic scheme of dividing the problem into learning and tracking stages.

2 Body Representation

We choose a simple representation for the human body: a modified Scaled Prismatic Model [12] that encodes the body as a set of 2D chains of articulated limb segments. This avoids 3D ambiguities while still capturing the natural degrees of freedom. Body parts are represented by rounded trapezoidal image templates defined by their end widths, and body poses are parametrized by their joint angles and apparent (projected) limb lengths. Including limb lengths, joint angles and hip and shoulder positions, our model

contains 33 parameters, giving 33-D state vectors $\mathbf{x} = (\theta_1, d_1, \theta_2, d_2, \dots, \theta_n, d_n)$. Figure 2 illustrates the parametrization and shows some sample poses.

Three additional parameters are used during tracking, two for the image location of the body centre and one for overall scale. We learn scale and translation independently of limb movements, so these parameters are not part of the learned body model. The template for each body part contains texture information used for model-image matching. Its width parameters depend on the subject's clothing and physique. They are defined during initialization and afterwards remain fixed relative to the overall body scale, which is actively tracked.

3 Dynamical Model Formulation

Human motion is both complex and time-varying. It is not tractable to build an exact analytical model for it, but approximate models based on statistical methods are a potential substitute. Such models involve learning characteristic motions from example trajectories in parameter space. Our model learns the nonlinear dynamics by partitioning the parameter space into distinct regions or motion classes, and learning a linear autoregressive process covering each region.

3.1 Partitioning of State Space

In cases where the dynamics of a time series changes with time, a single model is often inadequate to describe the evolution in state space. To get around this, we partition the state space into regions containing separate models that describe distinct motion patterns. The partitions must satisfy two main criteria: (i) different motion patterns must belong to different regions; and (ii) regions should be contiguous in state space. *I.e.*, we need to break the state space into *contiguous regions* with *coherent dynamics*. Coherency means that the chosen dynamical model is locally accurate, contiguity that it can be reliably deduced from the current state space position. Different walking or running styles, viewpoints, *etc.*, tend to use separate regions of state space and hence separate sets of partitions, allowing us to infer pose or action from class information.

We perform an initial partitioning on unstructured input points in state space by using K-means on Mahalanobis distances (see fig. 3). The clusters are found to cut the state trajectories into short sections, all sections in a given partition having similar dynamics. The partitions are then refined to improve the accuracies of the nearby dynamical models. The local model estimation and dynamics based partition refinement are iterated in an EM-like loop, details of which are given in section 3.3.

3.2 Modelling the Local Dynamics

Despite the complexity of human dynamics and the use of unphysical image-based models, we find that the local dynamics within each region is usually well described by a linear Auto-Regressive Process (ARP):

$$\mathbf{x}_t = \sum_{i=1}^p \mathbf{A}_i \mathbf{x}_{t-i} + \mathbf{w}_t + \mathbf{v}_t \quad (1)$$

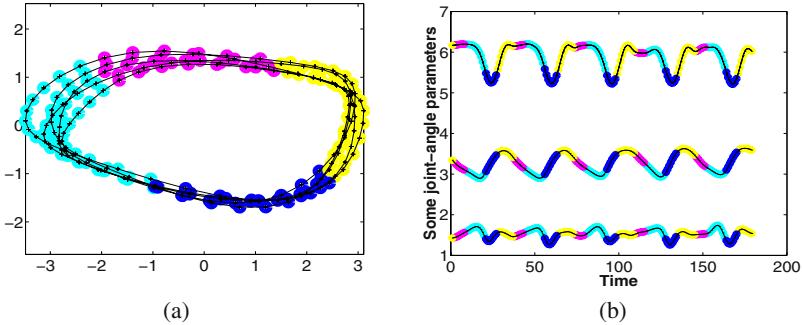


Fig. 3. (a) The initial partition of the state space of a walking motion (5 cycles), projected to 2-D using PCA (see text). (b) The clusters correspond to different *phases* of the walking cycle, here illustrated using the variations of individual joint angles with time. (The cluster labels are coded by colour). These figures illustrate the optimal clustering obtained for a $p=1$ ARP. For $p=2$, a single class suffices for modelling unidirectional walking dynamics.

Here, $\mathbf{x}_t \in \mathbb{R}^m$ is the pose at time t (joint angles and link lengths), p is the model order (number of previous states used), \mathbf{A}_i are $m \times m$ matrices giving the influence of \mathbf{x}_{t-i} on \mathbf{x}_t , $\mathbf{w}_t \in \mathbb{R}^m$ is a drift/offset term, and \mathbf{v}_t is a random noise vector (here assumed white and Gaussian, $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$).

The choice of ARP order is strongly dependent on the nature of the motions exhibited by the system. In practice, experiments on different kinds of motion showed that a second order ARP usually suffices for human tracking:

$$\mathbf{x}_t = \mathbf{A}_1 \mathbf{x}_{t-1} + \mathbf{A}_2 \mathbf{x}_{t-2} + \mathbf{v}_t \quad (2)$$

This models the local motion as a mass-spring system (set of coupled damped harmonic oscillators). It can also be written in differential form: $\ddot{\mathbf{x}}_t = \mathbf{B}_1 \dot{\mathbf{x}}_t + \mathbf{B}_2 \mathbf{x}_t + \mathbf{v}_t$.

3.3 Model Parameter Estimation

The parameters to be estimated are the state-space partitioning, here encoded by the class centers \mathbf{c}^k , and the ARP parameters $\{\mathbf{A}_1^k, \mathbf{A}_2^k, \dots, \mathbf{A}_p^k, \mathbf{Q}^k\}$ within each class ($k = 1 \dots K$). There are standard ways of learning ARP models from training data [10]. We compute maximum likelihood parameter estimates. We also want to take advantage of the well-structured nature of human motion. People rarely move their limbs completely independently of one another, although the actual degree of correlation depends on the activity being performed. This can be exploited by learning the dynamics with respect to a *reduced set of degrees of freedom* within each class, *i.e.* locally projecting the system trajectories into a lower dimensional subspace. Thus, within each partition, we:

1. reduce the dimensionality using linear PCA (in practice to about 5);
2. learn an ARP model in the reduced space;
3. “lift” this model to the full state space using the PCA injection;
4. cross-validate the resulting model to choose the PCA dimension.

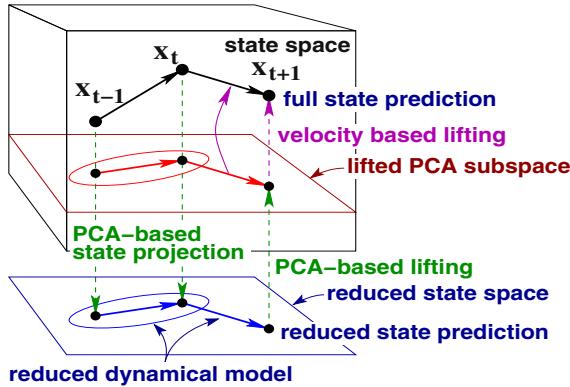


Fig. 4. Using a reduced dynamical model to predict states in a high-dimensional space. A given state is projected onto a low-dimensional space using PCA, within which a linear autoregressive process is used to predict a current (reduced) state. This is then lifted back into full state space to estimate a noise model in the high-dimensional space. To prevent the state from being continually squashed into the PCA subspace, we lift the velocity prediction and not the state prediction.

The basic scheme is illustrated in figure 4, and the complete algorithm is given below. Before applying PCA, the state-space dimensions need to be statistically normalized. This is done by dividing each dimension by its observed variance over the complete set of training data.

Algorithm for estimation of maximum-likelihood parameters

1. Initialize the state-space partitions by K-means clustering based on scaled (diagonal Mahalanobis) distance.
2. Learn an autoregressive model within each partition.
3. Re-partition the input points to minimize the dynamical model prediction error. If the class assignments have converged, stop. Otherwise go to step 2.

Step 2 above is performed as follows

1. Reduce the vectors in the class to a lower dimensional space by:
 - a) Centering them and assembling them into a matrix (by columns):

$$\mathbf{X} = [(\mathbf{x}_{p_1} - \mathbf{c}) (\mathbf{x}_{p_2} - \mathbf{c}) \cdots (\mathbf{x}_{p_m} - \mathbf{c})],$$
 where $p_1 \dots p_m$ are the indices of the points in the class and \mathbf{c} is the class mean.
 - b) Performing a Singular Value Decomposition of the matrix to project out the dominant directions: $\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T$.
 - c) Projecting each vector into the dominant subspace: each $\mathbf{x}_i \in \mathbb{R}^m$ is represented as a reduced vector $\mathbf{q}_i = \tilde{\mathbf{U}}^T(\mathbf{x}_i - \mathbf{c})$ in $\mathbb{R}^{m'} (m' < m)$, where $\tilde{\mathbf{U}}$ is the matrix consisting of the first m' columns of \mathbf{U} .
2. Build an autoregressive model, $\hat{\mathbf{q}} = \sum_{i=1}^p \mathbf{A}_i \mathbf{q}_{t-i}$, and estimate \mathbf{A}_i by writing this in the form of a linear regression:

$$\mathbf{q}_t = \tilde{\mathbf{A}} \tilde{\mathbf{q}}_{t-1}, \quad t = t_{p_1}, t_{p_2}, \dots t_{p_n} \quad (3)$$

where

$$\tilde{\mathbf{A}} = (\mathbf{A}_1 \ \mathbf{A}_2 \ \cdots \ \mathbf{A}_p), \quad \tilde{\mathbf{q}}_{t-1} = \begin{pmatrix} \mathbf{q}_{t-1} \\ \mathbf{q}_{t-2} \\ \vdots \\ \mathbf{q}_{t-p} \end{pmatrix}$$

3. Estimate the error covariance \mathbf{Q} from the residual between $\{\hat{\mathbf{x}}_i\}$ and $\{\mathbf{x}_i\}$ by “lifting” $\hat{\mathbf{q}}_t$ back into m dimensions:

$$\hat{\mathbf{x}}_t = \mathbf{c} + \tilde{\mathbf{U}}\hat{\mathbf{q}}_t \quad (4)$$

Step 3 above is performed as follows: The K-means based partitions are revised by assigning training points to the dynamical model that predicts their true motion best, and the dynamical models are then re-learned over their new training points. This EM / relaxation procedure is iterated to convergence. In practice, using dynamical prediction error as the sole fitting criterion gives erratic results, as models sometimes “capture” quite distant points. So we include a spatial smoothing term by minimizing:

$$\sum_{\text{training points}} (\text{prediction error}) + \lambda \cdot (\text{number of inter-class neighbors})$$

where λ is a relative weighting term, and the number of inter-class neighbors is the number of edges in a neighborhood graph that have their two vertices in different classes (*i.e.*, a measure of the lack of contiguity of a partition).

3.4 Inter-class Transitions

Many example-based trackers use discrete state HMMs (transition probability matrices) to model inter-cluster transitions [21,20]. This is unavoidable when there is no state space model at all (*e.g.* in exemplars [21]), and it is also effective when modelling time series that are known to be well approximated by a set of piecewise linear regimes [5]. Its use has been extended to multi-class linear dynamical systems exhibiting continuous behavior [14], but we believe that this is unwise, as the discrete transitions ignore the location-within-partition information encoded by the continuous state, which strongly influences inter-class transition probabilities. To work around this, quite small regions have to be used, which breaks up the natural structure of the dynamics and greatly inflates the number of parameters to be learned. In fact, in modelling human motion, the current continuous state already contains a great deal of information about the likely future evolution, and we have found that this alone is rich enough to characterize human motion classes, without the need for the separate hidden discrete state labels of HMM based models.

We thus prefer the simpler approach of using a mixture of linear dynamical models over an explicit spatial partition, where the ‘class’ label is just the current partition cell. More precisely, we use soft partition assignments obtained from a Gaussian mixture model based at the class centres, so the dynamics for each point is a weighted random

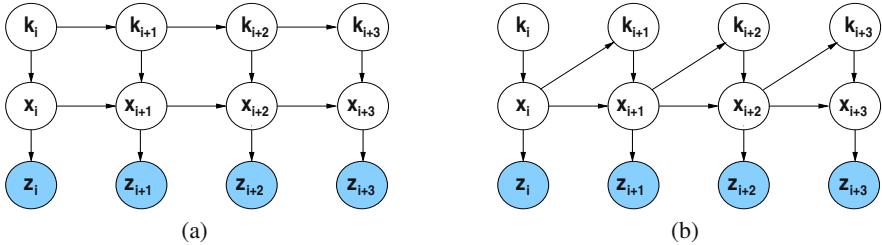


Fig. 5. Graphical models for inter-class transitions of a system. (a) An HMM-like mixed-state model, and (b) our inter-class transition model (\mathbf{z}_i : observation, \mathbf{x}_i : continuous state, k_i : discrete class). Transitions in an HMM are learned as a fixed transition probability matrix, while our model allows location-sensitive estimation of the class label by exploiting continuous state information.

mixture over the models of nearby partitions. Our classes cover relatively large regions of state space, but transitions typically only occur at certain (boundary) areas within them. Constant transition probabilities given the current class label would thus be inappropriate in our case.

Figure 5 compares the two schemes in graphical form. By modelling the class-label to be conditional on continuous state, we ensure a smooth flow from one model to the next, avoiding erratic jumps between classes, and we obviate the need for complex inference over a hidden class-label variable.

4 Image Matching Likelihood

At present, for the model-image matching likelihood we simply use the weighted sum-of-squares error of the backwards-warped image against body-part reference templates fixed during initialization. Occlusions are handled using support maps. Each body part P has an associated support map whose j^{th} entry gives the probability that image pixel j currently ‘sees’ this part. Currently, we use hard assignments, $p(j \text{ sees } P) \in \{0, 1\}$. To resolve the visibility ambiguity when two limbs overlap spatially, each pose has an associated *limb-ordering*, which is known a priori for different regions in the pose space from the training data. This information is used to identify occluded pixels that do not contribute to the image matching likelihood for the pose. We charge a fixed penalty for each such pixel, equal to the mean per-pixel error of the visible points in that segment. Some sample support maps are shown in figure 8(b).

5 Tracking Framework

Our tracking framework is similar to Covariance Scaled Sampling [18]. For each mode of \mathbf{x}_{t-1} , the distribution $\mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{Q})$ estimated by the dynamical model (1,5) is sampled, and the image likelihood is locally optimized at each mode. State probabilities are propagated over time using Bayes’ rule. The probability of the tracker being in state (pose) \mathbf{x}_t at time t given the sequence of observations $\mathcal{Z}_t = \{\mathbf{z}_t, \mathbf{z}_{t-1} \dots \mathbf{z}_0\}$ is:

$$p(\mathbf{x}_t | \mathcal{Z}_t) = p(\mathbf{x}_t | \mathbf{z}_t, \mathcal{Z}_{t-1}) \propto p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathcal{Z}_{t-1})$$

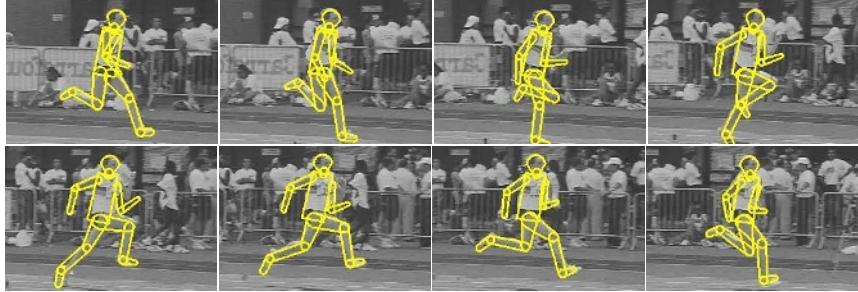


Fig. 6. Results from tracking athletic motion (frames 0,4,8,12,16,20,24). The tracker was trained on a different athlete performing a similar motion. Strong priors from the dynamical model allow individual limbs to be tracked in the presence of a confusing background. Note that the left arm is not tracked accurately. This is due to the fact that it was occluded in the initial image and hence no information about its appearance was captured in the template. However, the dynamics continue to give a good estimate of its position.

where \mathcal{X}_t is the sequence of poses $\{\mathbf{x}_i\}$ up to time t and

$$p(\mathbf{x}_t | \mathcal{Z}_{t-1}) = \int p(\mathbf{x}_t | \mathcal{X}_{t-1}) p(\mathcal{X}_{t-1} | \mathcal{Z}_{t-1}) d\mathcal{X}_{t-1} \quad (5)$$

The likelihood $p(\mathbf{z}_t | \mathbf{x}_t)$ of observing image \mathbf{z}_t given model pose \mathbf{x}_t is computed based on the image-model matching error. The temporal prior $P(\mathbf{x}_t | \mathcal{X}_{t-1})$ is computed from the learned dynamics. In our model, the choice of discrete class label k_t is determined by the current region in state space, which in our current implementation depends only on the previous pose \mathbf{x}_{t-1} , enabling us to express the probability as

$$p(\mathbf{x}_t | \mathcal{X}_{t-1}) = p(\mathbf{x}_t | \mathcal{X}_{t-1}, k_t) p(k_t | \mathbf{x}_{t-1}) \quad (6)$$

The size and contiguity of our dynamical regions implies that $p(k_t | \mathbf{x}_{t-1})$ is usually highly unimodal. The number of modes increases when the state lies close to the boundary between two or more regions, but in this case, the spatial coherence inherited from the training dynamics usually ensures that any of the corresponding models can be used successfully, so the number of distinct modes being tracked does not tend to increase exponentially with time. For each model $k = 1 \dots K$, we use a Gaussian posterior for $p(k | \mathbf{x}_t)$: $p(k | \mathbf{x}_t) \propto e^{-((\mathbf{x}_t - \mathbf{c}_k)\Sigma^{-1}(\mathbf{x}_t - \mathbf{c}_k))/2}$ where \mathbf{c}_k is the center of the k^{th} class. Note that with a second order ARP model, $p(\mathbf{x}_t | \mathcal{X}_{t-1}) = p(\mathbf{x}_t | \mathcal{X}_{t-1}, \mathbf{x}_{t-2})$.

6 Results

We demonstrate our technique by learning models for different classes of human motion and using them to track complete body movements in unseen video sequences. Here, we present results from two challenging sequences.

1. Fast athletic motion: This is a case where traditional methods typically fail due to high motion blur. A hand-labelled sequence covering a few running cycles is used to train a model and this is used to track a different person performing a similar motion.

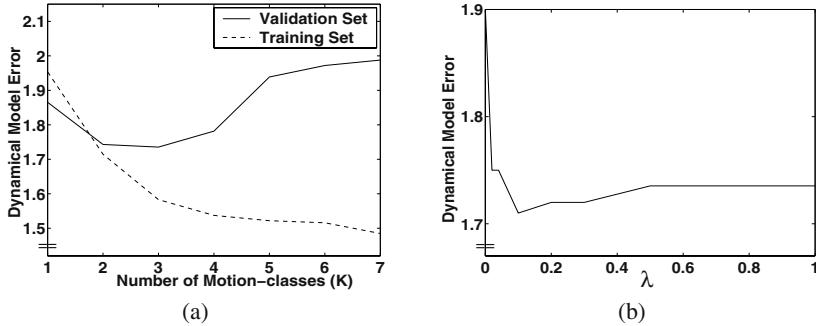


Fig. 7. (a) Dynamical model prediction error w.r.t. number of motion-classes in the turning experiment. Minimizing the validation error selected 3 classes, corresponding to the two walking directions and turning between them. (b) The influence of spatial regularization when re-partitioning the state space. A weak regularization $\lambda \sim 0.1$ gives the optimal dynamical estimates. A larger λ causes the partition to remain too close to the suboptimal initial K-means estimate.

For a given viewing direction, we find that a single 2nd order autoregressive process in 5 dimensions suffices to capture the dynamics of such running motions. A tracking example is shown in figure 6.

2. Switching between turning and walking: This experiment illustrates the effectiveness of our inter-class transition model. A 300-frame sequence consisting of walking in different directions and turning motion is used as training data. Our learning algorithm correctly identifies 3 motion patterns (see figure 7(a)), corresponding to two different walking directions and turning between them. The frames corresponding to the centers of these 3 classes are shown in figure 8(a). While tracking a new sequence, the model correctly shifts between different classes enabling smooth switching between activities. Figure 8(c) shows complete tracking results on an unseen test sequence.

In both cases, the models were initialized manually (we are currently working on automatic initialization), after which only the learned dynamics and appearance information were used for tracking. Position and scale changes were modelled respectively as first and zeroth order random walks and learned online during tracking. This allows us to track sequences without assuming either static or fixating cameras, as is done in several other works. The dynamical model alone gives fairly accurate pose predictions for at least a few frames, but the absence of clear observations for any longer than this may cause mistracking.

Figure 7(b) shows how repartitioning (step 3 of our parameter estimation algorithm) improves on the initial K-means based model, provided that a weak smoothing term is included.

7 Conclusion

We have discussed a novel approach to modelling dynamics of high degree-of-freedom systems such as the human body. Our approach is a step towards describing dynamical behavior of high-dimensional parametric model spaces without having to store extremely

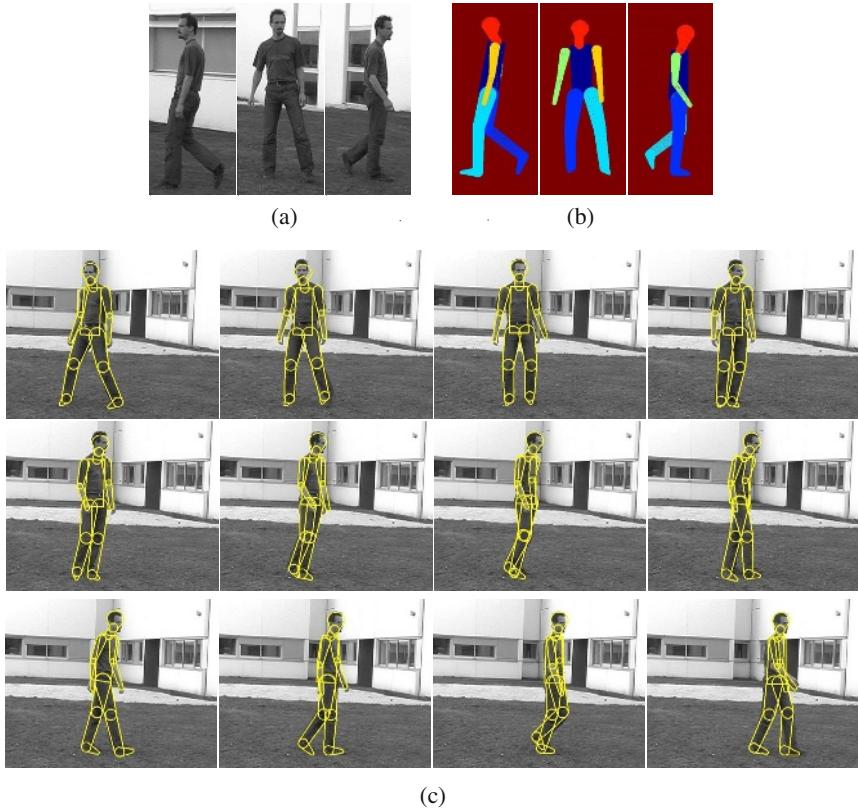


Fig. 8. Examples from our turning experiment. (a) Poses characterizing the 3 motion classes learned. (b) Support maps illustrating occlusion information for the 3 classes (color coded by body part). (c) Tracking results (every 6th frame from 0–66). The corresponding state vectors show a smooth transition between the turning and walking models.

large amounts of training data. It takes advantage of local correlations between motion parameters by partitioning the space into contiguous regions and learning individual local dynamical behavior within reduced dimensional manifolds. The approach was tested on several different human motion sequences with good results, and allows the tracking of complex unseen motions in the presence of image ambiguities. The mixture-based learning scheme developed here is practically effective, and scalable in the sense that it allows models for different actions to be built independently and then stitched together to cover the complete ‘activity space’. The learning process can also be made interactive to allow annotation of different classes for activity recognition purposes.

In terms of future work, the appearance model needs to be improved. Adding detectors for characteristic human features and allowing the appearance to evolve with time would help to make the tracker more robust and more general. Including a wider range of training data would allow the tracker to cover more types of human motions.

An open question is whether non-parametric models could usefully be incorporated to aid tracking. Joint angles are a useful output, and are probably also the most appropriate representation for dynamical modelling. But it might be more robust to use comparison

with real images, rather than comparison with an idealized model, to compute likelihoods for joint-based pose tracking.

Acknowledgements. This work was supported by the European Union FET-Open Project VIBES.

References

1. Matthew Brand and Aaron Hertzmann. Style Machines. In *Siggraph 2000, Computer Graphics Proceedings*, pages 183–192, 2000.
2. C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *International Conference on Computer Vision and Pattern Recognition*, pages 8–15, 1998.
3. T. Cham and J. Rehg. A Multiple Hypothesis Approach to Figure Tracking. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 239–245, 1999.
4. A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing Action at a Distance. In *International Conference on Computer Vision*, 2003. To appear.
5. Z. Ghahramani and G. Hinton. Switching State-Space Models. Technical report, Department of Computer Science, University of Toronto, Canada, 1998.
6. Tony Heap and David Hogg. Nonlinear Manifold Learning for Visual Speech Recognition. In *International Conference on Computer Vision*, pages 494–499, 1995.
7. Tony Heap and David Hogg. Wormholes in Shape Space: Tracking Through Discontinuous Changes in Shape. In *International Conference on Computer Vision*, pages 344–349, 1998.
8. N. Howe, M. Leventon, and W. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. In *Neural Information Processing Systems*, 1999.
9. S. Ju, M. Black, and Y. Yacoob. Cardboard People: A Parameterized Model of Articulated Motion. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 38–44, 1996.
10. H. Lütkepohl. *Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin, Germany, second edition, 1993.
11. G. Mori and J. Malik. Estimating Human Body Configurations Using Shape Context Matching. In *European Conference on Computer Vision*, volume 3, pages 666–680, 2002.
12. D. Morris and J. Rehg. Singularity Analysis for Articulated Object Tracking. In *International Conference on Computer Vision and Pattern Recognition*, pages 289–296, 1998.
13. B. North, A. Blake, M. Isard, and J. Rittscher. Learning and Classification of Complex Dynamics. *Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.
14. V. Pavlovic, J. Rehg, and J. MacCormick. Learning Switching Linear Models of Human Motion. In *Neural Information Processing Systems*, pages 981–987, 2000.
15. D. Ramanan and D. Forsyth. Finding and Tracking People from the Bottom Up. In *International Conference on Computer Vision and Pattern Recognition*, 2003.
16. H. Sidenbladh and M. Black. Learning Image Statistics for Bayesian Tracking. In *International Conference on Computer Vision*, volume 2, pages 709–716, 2001.
17. H. Sidenbladh, M. Black, and L. Sigal. Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In *European Conference on Computer Vision*, volume 1, 2002.
18. C. Sminchisescu and B. Triggs. Covariance Scaled Sampling for Monocular 3D Body Tracking. In *International Conference on Computer Vision and Pattern Recognition*, 2001.
19. J. Sullivan and S. Carlsson. Recognizing and Tracking Human Action. In *European Conference on Computer Vision*, 2002.
20. A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Learning a kinematic prior for tree-based filtering. In *Proc. British Machine Vision Conference*, volume 2, pages 589–598, 2003.
21. K. Toyama and A. Blake. Probabilistic Tracking in a Metric Space. In *International Conference on Computer Vision*, pages 50–59, 2001.

Novel Skeletal Representation for Articulated Creatures

Gabriel J. Brostow, Irfan Essa, Drew Steedly, and Vivek Kwatra

Georgia Institute of Technology, Atlanta GA 30332, USA

<http://www.cc.gatech.edu/cpl/projects/spines>

Abstract. Volumetric structures are frequently used as shape descriptors for 3D data. The capture of such data is being facilitated by developments in multi-view video and range scanning, extending to subjects that are alive and moving. In this paper, we examine vision-based modeling and the related representation of moving articulated creatures using *spines*. We define a spine as a branching axial structure representing the shape and topology of a 3D object's limbs, and capturing the limbs' correspondence and motion over time.

Our spine concept builds on *skeletal* representations often used to describe the internal structure of an articulated object and the significant protrusions. The algorithms for determining both 2D and 3D skeletons generally use an objective function tuned to balance stability against the responsiveness to detail. Our representation of a spine provides for enhancements over a 3D skeleton, afforded by temporal robustness and correspondence. We also introduce a probabilistic framework that is needed to compute the spine from a sequence of surface data.

We present a practical implementation that approximates the spine's joint probability function to reconstruct spines for synthetic and real subjects that move.

1 Introduction

We are interested in the detection and tracking of features in volumetric images. Volume images capture shape as a temporal sequence of boundary voxels or other forms of 3D surfaces. Specifically, we wish to address situations where the subject is known to have and is exercising an articulated structure. This assumption grants us use of a specific class of geometric modeling solutions. The various methods for skeletonizing 2D and 3D images share the objectives of identifying extrema, features with some geometric significance, and capturing the spatial relationships between them [9]. Skeletons, much like generalized cylinders [4,21], serve the purpose of abstracting from raw volume or surface data to get higher level structural information.

We propose that evaluating volumetric data of a subject over *time* can disambiguate real limbs from noisy protrusions. In a single image, knowledge of the specific application alone would dictate the noise threshold to keep or cull small branches of the skeleton. Many such algorithms exist. In the case of articulated

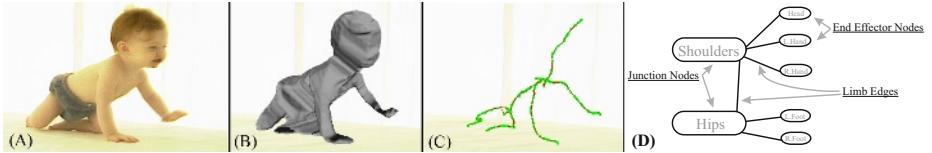


Fig. 1. (A) Articulated subject, (B) reconstructed surface, (C) extracted skeleton, (D) spine graph limbs encoding motion over time; nodes labeled for illustration only.

moving subjects, the volumetric images change but the underlying structure stays the same. We hypothesize that the parts of the skeleton within each image that are consistent over time more reliably capture the subject’s structure. To this end, we introduce our notion of spines.

As defined in [4], a generalized cylinder is a surface obtained by sweeping a planar cross section along an axis, or space curve. To represent a body made of multiple generalized cylinders, we need to merge axes of the different limbs into one branching axial structure. The branching structure can be represented by a graph, $G(LimbBoundaries, Limbs)$, where edges are limbs, leaf nodes are end effectors, and the remaining nodes (all of degree > 2) are limb junctions (see Figure 1D). So far, we have described the general formulation of a skeleton [5]. To parameterize the motion of a skeleton, we express the new spine graph as a function over time:

$$Spine_t = F(G, t). \quad (1)$$

For a given time t , the limbs of G will be in a specific pose, captured by F ’s mapping of G ’s topology to axial curves in 3D – a single *skeleton*. When estimating a data set’s *spine* in the subsequent sections, we will constrain F to manipulate the limbs of a G that represents a series of topologically consistent skeletons. These skeletons are determined as probable given the input data.

The implementation of our algorithm is a modular pipeline. It first reduces the complexity of multi-view video data to voxels, further to polygons, and finally to spines. The resulting model captures the original degrees of freedom needed to play back the subject’s motions (see Figure 1).

2 Related and Motivating Work

The 2D analogue to our problem is the tracking of correspondence in medial axes, which were first introduced by Blum [5]. Given any of the numerous 2D skeletonizing techniques, including the classic grassfire models based on distance and the more robust area-based techniques [3], the work of Sebastian et al. [23] can determine correspondence by minimizing edit-distances of skeleton graphs in 2D.

The medial axes of 3D surfaces are not directly applicable because they generate 2D manifold “sheets” through a surface. While medial scaffolds can be calculated fairly robustly [24,19], they require further processing [28] to estimate good 1D axes.

Several 3D skeletonization algorithms have been developed using 3D Voronoi cells to partition the space within a mesh [2,13,25,12,16]. The cell-walls of these convex polyhedra land at equal distances from their designated surface start-points – some at or near the medial axis. This approach, with various extensions of projection and pruning, can generally serve to synthesize axes. In contrast to these, our approach and implementation are based on two sub-domains of solutions: measuring of geodesic distance from geometric modeling, and principal curves from statistics.

Geodesic Distance: In Section 4.1 we will discuss in greater detail how a surface can be treated as a piecewise continuous distance field that separates features from each other. Verroust and Lazarus [27] used such a technique to determine axes of symmetry within limbs, and how to connect them to critical points (special topological features) on the mesh surface. In an application not requiring branching axes, Nain et al. [22] used geodesic distances on colon models to determine center-lines for virtual colonoscopy navigation. Recently, a geodesic distance based metric was used by Katz and Tal [17] to help assign patches as members of explicit limbs, resulting in course animation control-skeletons. All these approaches benefit from works such as [15] which identify extrema, or features that protrude from or into a surface mesh. Our approach uses such extrema-finding and a geodesic distance metric to better model skeleton branching.

Principal Curves: Hastie and Stuetzle [14] defined principal curves as passing through the middle of a multidimensional data set, as a representation of self-consistency to generalize principal components. For fixed length curves in a geometric setting, Kegl et al. [18] showed how to minimize the squared distance between the curve and points sampled randomly from the encompassing shape. Most recently, [7] and [8] extended this notion of principal curves to 3D, formalizing the problem as an optimization which also seeks to minimize the curve length. Our extension is to incorporate branching and temporal correspondence.

3 Spine Formulation and Estimation

We build on the axial representation of generalized cylinders of [8,7] because of their elegant mathematical formulation. They treat the regression problem of finding a single curve for a surface as the minimization of a global energy function. Much like the previous work on principal curves [14,18], they seek to minimize the total distance from the axial curve to the surface. But in addition, [7] incorporates a term which penalizes the curve’s length. This augmentation helps force the shorter curve to smoothly follow the middle of a surface, instead of, for example, spiraling through all the boundary points.

For our spine formulation, we seek to further incorporate: (a) skeletons S that model branching curves of individual surfaces X and (b) data captured over a period of time T . We propose a discriminative probabilistic approach to computing spines by finding G , S , and limb end effectors E , which maximize:

$$P(G, S_{1:T}, E_{1:T} | X_{1:T}) = P(G|S_{1:T}, E_{1:T}, X_{1:T}) \cdot P(S_{1:T}, E_{1:T} | X_{1:T}) \quad (2)$$

To compute and optimize the joint probability $P(S_{1:T}, E_{1:T} | X_{1:T})$ requires searching over all skeletons over all time simultaneously. In order to make the solution more computationally tractable, we make the assumption that S_t and E_t are independent of $S_{t'}$ and $E_{t'} \forall (t' \neq t)$, given X_t :

$$P(G, S_{1:T}, E_{1:T} | X_{1:T}) \approx P(G | S_{1:T}, E_{1:T}, X_{1:T}) \cdot \prod_{t=1}^T P(S_t, E_t | X_t) \quad (3)$$

This assumption can lead to temporal inconsistencies that can be resolved once G is estimated (as shown in Section 4.2). We use a bottom-up approach that individually approximates each S_t and E_t individually, and then estimates G . Ideally, we would like to estimate G , S , and E using an EM-like algorithm by iterating back and forth between estimates of G and (S_t, E_t) . However, we have found that the greedy estimate of S and E , while noisy, is sufficient to determine a G consistent with the subject's limb topology.

4 Temporally Constrained Branching Spines

In this section, we will start by describing our method for locating the set of end effectors E_t and extracting a branching skeleton graph from a single 3D surface X_t . Using this or other techniques, we can generate an individual skeleton S_t at each time t , $1 \leq t \leq T$. These (S_t, E_t) will be inherently noisy, as a result of being calculated independently for each t . In Section 4.2, we describe how we combine these individual and often overly complex graphs into a consistent, representative spine for the entire time sequence.

The fairly significant attention given to the problem of building a single branching 3D skeleton includes numerous approaches. After experimenting with portions of several of these [20,15], we have developed our own extension to the level-set method of [27]. In theory, any 3D skeleton-finding technique would be suitable, if it meets the following requirements:

1. Is self-initializing by automatically finding extrema E_t .
2. Generates a principal curve leading to each extremum.
3. Constructs internal junctions of curves only as necessary to make a connected tree.

More precision might be achieved with more iterations or other techniques, but these might only further improve the results of applying our general probabilistic framework of (3). We proceed to explain our greedy method for obtaining a 3D branching skeleton S_t from a surface, with just one iteration of maximizing (3)'s second term followed by correspondence tracking.

4.1 Creating a Skeleton for a Single Surface

Once we have a 3D surface X_t for volumetric image (or frame) t , we want to extract a skeleton from it. We accomplish this goal in two stages. First we find the

tip of each extremity and grow a skeleton from it. Then we merge the resulting skeletons to maximize the presence of the highest quality portions of each. In terms of maximizing $P(S_t, E_t | X_t)$, we are first finding a set of candidates for the end effectors of E_t and the limbs of S_t . We then pick from these the combination that is optimal with respect to our probability metric.

Growing Skeletons: This part of our algorithm is based on the work of [27]. Starting at a seed point on an extremity of the mesh, they sweep through the surface vertices, labelling each with its increasing geodesic distance. These distances are treated as a gradient vector field, which is in turn examined for topological critical points. The critical points are used as surface attachment sites for virtual links (non-centered) between the axes when the mesh branches.

But for our purposes, we want a skeleton that always traverses through the middle of the subject’s extremities. Locating meaningful extremal points is itself an open problem, though the difficulties are generally application specific. Much like the above algorithm which has one source, the vertices of a surface mesh can be labelled with their *average* geodesic distance (AGD) to *all* other points. Surface points thus evaluated to be local extrema of the AGD function correspond to protrusions. Knowledge of the expected size of “interesting” protrusions can be used as a threshold on which local maxima qualify as global extrema.

Hilaga et al. [15] address the significant computational cost of finding the AGD by approximating it with uniformly distributed base seed-points. Applying the simpler base-point initialization of [27,10] in a greedy manner located the desired candidates for E_t for our data sets.

Instead of the separate *distance* and *length* terms minimized by [7], we use the isocontours of geodesic distance to build level sets that serve as our error metric. The vertices of the mesh are clustered into those level-sets by quantizing their distances from the seed point into a fixed number of discrete bins (usually 100). Figures 2C-D illustrate this process. Each skeleton node is constructed by minimizing the distance between the vertices in the level set and the node, *i.e.*, the centroid of the vertices.

By walking along edges of the surface graph from the seed point’s level set toward the last one, skeleton-nodes are added and progressively connected to each other. Figure 3A illustrates this process in 2D. This approach successfully creates a tree graph of nodes, or skeleton, which represents the central axes and internal branching points of genus zero meshes.

The skeleton-generation algorithm is repeated for each of the other limb-tips, producing a total of five skeleton-graphs for the starfish example (see Figure 2). These are our candidates for the best S_t for this X_t . Note that the most compact level-sets usually appear as tidy cylindrical rings on the limb where that respective skeleton was seeded.

Merging Skeletons: All of the constituent skeletons S_t serve as combined estimates of the mesh’s underlying limb structure. The best representation of that structure comes from unifying the most precise branches of those skeletons – the ones with smallest error, or equivalently, maximum $P(S_t, E_t | X_t)$. A high quality skeleton node best captures the shape of its “ring” of vertices when the

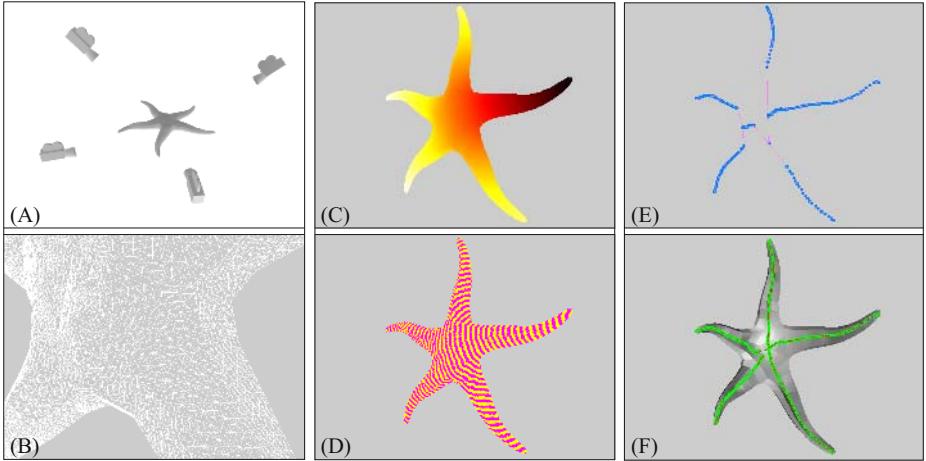


Fig. 2. Example of generating a skeleton for a synthetic starfish mesh. (A) Capture images of the starfish from a variety of vantage points (B) Extract a 3D surface using generalized voxel carving and improved marching cubes (C) Starting at one extremity tip, calculate geodesic distances for each vertex (D) Quantize distances and cluster vertices into bins of the same distance (E) Create a skeleton by walking through the progression of level set rings (F) Repeat C-E for each tip and merge into a single representative skeleton.

ring is short and has small major and minor axes. With this metric, we calculate a cost function C for each node in the constituent skeletons:

$$C_i = \frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}{\# \text{ of points in ring } i}. \quad (4)$$

The σ quantities come from singular values of the decomposition $\bar{\mathbf{P}} = \mathbf{U}_P \Sigma_P \mathbf{V}_P^T$, where $\bar{\mathbf{P}}$ represents the mean-centered coordinates of the points p_i in this ring. Note that the resulting \mathbf{v}_i vectors in $\mathbf{V}_P^T = \{\mathbf{v}_1 | \mathbf{v}_2 | \mathbf{v}_3\}^T$ will usually represent the ring's major, minor, and central axes. Replacing \mathbf{v}_3 with $\mathbf{v}_1 \times \mathbf{v}_2$ produces a convenient local right-hand coordinate frame for each node.

Each chain of bi-connected nodes represents a limb. To assemble the single representative graph of this frame, we copy the best version of each limb available in the constituent skeletons. Limb quality \mathbf{Q}_L is measured as:

$$\mathbf{Q}_L = N - \sum_1^N C_i, \quad (5)$$

where N is the total number of nodes in limb L . Since nodes from different skeletons are being compared through (5), the C_i 's must be normalized by dividing them all by the $\max(C_i)$ of all the skeletons.

Figure 3B illustrates a novel algorithm that we developed to generate limb-correspondences for topologically perturbed tree graphs of the same structure.

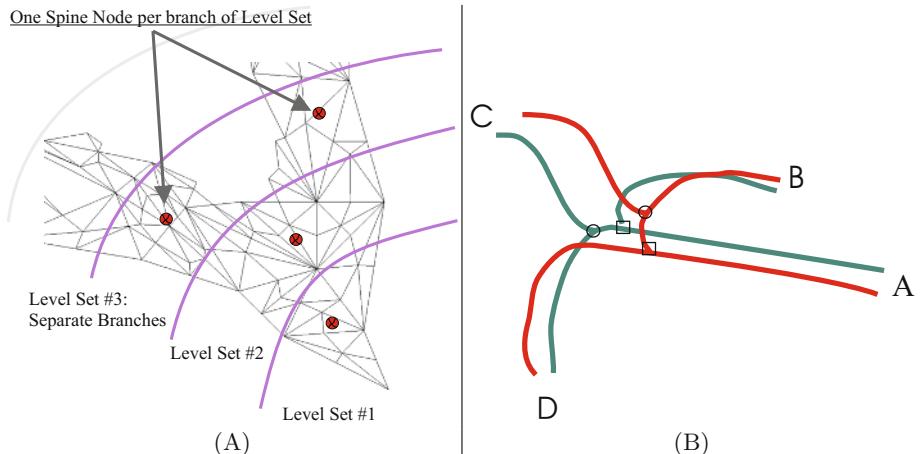


Fig. 3. (A) 2D example of clustering connected vertices into bins of similar geodesic distance and walking through the resulting level set rings. (B) In the right figure, the red and green skeletons represent the same “creature,” possibly seeded from two different places. Wishing to copy nodes from the best limbs each constituent skeleton has to offer, we developed a leaf-node seeking topology matching algorithm that recognizes that these pairs of three-way junctions should be a single four-way junction.

There appears to be no previously established graph theoretic solution for this problem, and our approach is simply:

1. Tag all limb-tips that we are confident of as *Supernodes*; i.e. nodes on both color graphs located at [A, B, C, D] correspond to each other.
2. Traversing inward, the next encountered branch-node in each graph also corresponds to that of the other color: walking from supernode A, the skeleton-nodes at the square-symbols should be grouped into a supernode of their own. From C, the circles will form a supernode. Iterating this process from the outside inward will reveal that the circle and square supernodes should be merged into a four-way *metanode*, which would serve as the point of unification when merging limbs from the red and green skeletons.

4.2 Correspondence Tracking

Now that we can estimate a single skeleton that represents one volumetric image, we adapt the process to handle a sequence of volumes. All the measurements from the sequence of $X_{1:T}$ are now abstracted as $(S_{1:T}, E_{1:T})$, simplifying the first term in (3) to $P(G|S_{1:T}, E_{1:T})$. Finding the G that maximizes this probability eliminates extraneous limbs which might have resulted from overfitting. The danger of overfitting exists because skeleton elements may be created in support of surface-mesh elements that looked like protrusions in that frame only.

Our 3D correspondence problem of finding the best G is significantly easier to automate than trying to perform surface-vertex matching between two dense

meshes of the sequence. Assuming the subject grows no new appendages and with no other priors, we can safely choose the appropriate number of tips to be the most frequently observed number of limb tips. This number of tips, or leaf nodes in G , is $K = \text{the mode of } |E_t|, 1 \leq t \leq T$ (see Figure 7).

Knowing how many appendages to look for, we spatially align each exploratory skeleton from the sequence with respect to its temporal neighbors to reveal the $|E_t| - K$ superfluous tips that should be culled. We start with all the subsequences of frames that already have the correct number of tips K , and tag the frame from the middle of the largest such cluster as the reference frame; allowing that longer sequences may need to automatically select multiple reference frames. Each frame is then processed in turn, constructing a combinatorial list of possible tip-correspondences between the reference tips \mathbf{A} and the tips in the current frame \mathbf{B} . Each possible mapping of $\mathbf{B} \rightarrow \mathbf{A}$ is evaluated using the point-cluster alignment algorithm of [1]. Their technique aligns point clouds as much as possible using only translation and rotation. The combination with the smallest error, E_{\min} , is kept as the correct assignment, where

$$E = \sum_{k=1}^K \|B_k - \hat{\mathbf{R}}A_k - \hat{\mathbf{T}}\|^2. \quad (6)$$

Here $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$ are the least-squares optimal rotation and translation. $\hat{\mathbf{T}}$ simply comes from the alignment of the point clouds' centroids. $\hat{\mathbf{R}}$ is calculated by maximizing the $\text{Trace}(\hat{\mathbf{R}}\mathbf{H})$, where \mathbf{H} is the accumulated point correlation matrix:

$$\mathbf{H} = \sum_{k=1}^K A_k B_k^T. \quad (7)$$

By decomposing $\mathbf{H} = \mathbf{U}_R \Sigma_R \mathbf{V}_R^T$, the optimal rotation is:

$$\hat{\mathbf{R}} = \mathbf{V}_R \mathbf{U}_R^T. \quad (8)$$

After assigning the tips of all these frames, we apply the same error metric to try out the combinations of tip-assignments with frames having alternate numbers of tips. However, these frames are compared to both the reference frame and the frame nearest in time with K tips. This brute-force exploration of correspondence is computationally tractable and robust for creatures that exhibit some asymmetry and have a reasonable number of limbs (typically < 10).

4.3 Imposing a Single Graph on the Spine

With the known trajectories of corresponding limb tips throughout the sequence, we can re-apply the skeleton merging technique from Section 4.1. This time however, we do not keep all the limbs as we did in the exploratory phase, only those that correspond to the K limb-tips. The results of this portion of the algorithm are pictured in Figure 4 and discussed further in Section 5.

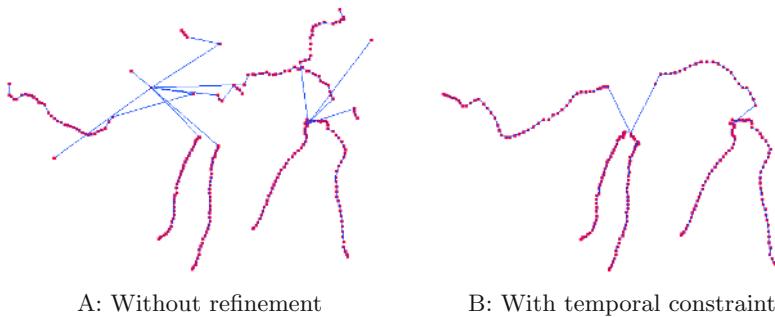


Fig. 4. Refinement through imposing of correspondence into the sequence.

Except for the frames of the sequence where the subject’s limbs were hidden or tucked too close to the body, we can expect the topology of skeletons throughout the sequence to be identical. The most frequently occurring topology is established as G , and corresponds to the first term in 3. This correspondence and trajectory information allows us to construct a single character spine for playback of the whole sequence of poses by parameterizing on each limb’s length. Each topologically consistent limb of the skeleton sequence is resampled at the same interval producing a single spine.

5 Experiments and Results

We tried our algorithm on a variety of small creatures after building a data-capture stage that would both be comfortable for our subjects and minimize the need for video segmentation beyond chromakeying. Twenty video cameras were attached to an aluminum exoskeleton shaped roughly like a cylinder 3 meters in diameter. Their viewing angles were chosen heuristically to maximize viewing coverage and to minimize instances of cameras seeing each other’s lenses. The capture volume itself is $(75\text{cm})^3$, and can accommodate creatures that stay within the space (Figure 5). Our subjects often required human proximity and were too heavy for our transparent flooring, so we were only able to leverage a subset of the cameras present.

With this setup, we are able to obtain video from a dome of inward facing, calibrated and synchronized cameras [29,6]. This allowed us to employ the Generalized Voxel Carving (GVC) algorithm of [11]. Their system functions as a hybrid form of wide-baseline stereo and voxel-carving, enabling the resulting voxel model to reflect concavities found on parts of the subject’s surface. Each second of multi-view footage produces 30 voxel models similar to the system of [26].

5.1 Real Subjects

Baby: The baby data is the result of filming an 11-month old infant using nine cameras. The sequence is 45 frames long because that was the speed with which



Fig. 5. Our Capture Setup: Twenty video cameras were attached to an aluminum exoskeleton shaped roughly like a cylinder 3 meters in diameter. Their viewing angles were chosen heuristically to maximize viewing coverage of subjects raised in the middle, and to minimize instances of cameras seeing each other's lenses. The capture volume itself is $(75\text{cm})^3$.

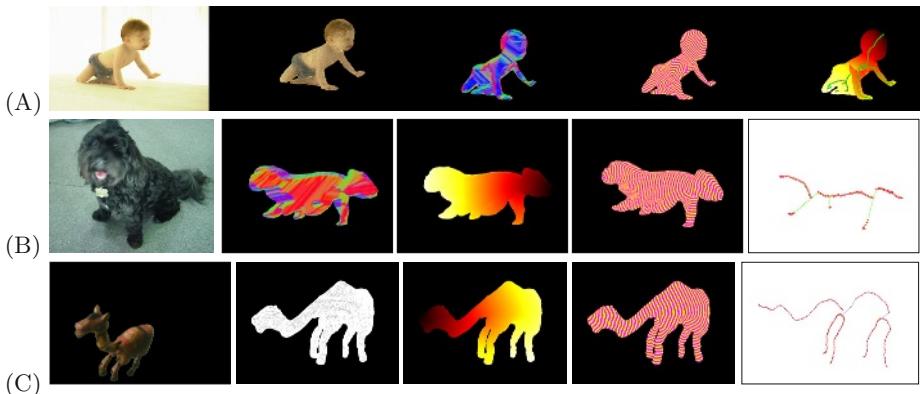


Fig. 6. (A) BABY DATASET: From left to right, one of the views, voxels, polygonal model, level sets, and skeleton with distance function. (B) DOG DATASET: subject, polygonal model, distance function, level sets, and resulting spine. (C) CAMEL PUPPET DATASET: one view, wireframe, distance function, level sets, and resulting spine.

she crawled down the length of the stage. Her progress forward is mostly due to her arms and right leg, while she tends to drag her left leg which causes frequent merging of her voxel-model from the waist down. The spine generation models her head and arms very consistently, but the correspondence tracker cannot resolve her legs and mis-assigns one leg or the other for the majority of frames.

Dog: The dog was the most challenging of our test-subjects simply because we had only seven cameras that could operate without also filming the dog's handlers. The volume reconstructions are all close to their average of 1.04M

voxels. Examination of the polygonal-mesh sequence reveals that much of this bulk comes from the ghost-voxels under his stomach that were carved successfully in the previous and subsequent test subjects when more cameras were running.

Camel Puppet: The camel marionette, pictured in Figure 6C, is 26 cm long and stretches to a height of 42 cm. While the subject didn't change in volume throughout shooting, its representation varied throughout the sequence between 600k and 800k voxels, largely due to self-occlusions. The polygonal representations averaged 200k polygons. The sequence has 495 frames, and was filmed using 12 color cameras. The camel's motion changes in the sequence from leg-jostling at the start to vigorous kicking and raising of the neck by the end. Our system was only hindered by the occasional "merging" of legs as they tucked underneath or appeared close enough to each other to be joined in the voxel stage. With mostly good frames, the exploratory skeleton-generation fed the correspondence tracker, which in turn determined that there were five limbs. The resulting creature spine is pictured in Figure 4B. As illustrated, the correspondence tracking balances out the greedy limb inclusion of the exploratory skeletons. The online video also demonstrates this.

The average processing times for skeleton-generation using our unoptimized implementation of the algorithms were consistently under four minutes per mesh on a Pentium 4 PC with one or more GB of memory. The correspondence-tracking portion of our algorithm (Section 4.2) took ten minutes on our 495 frame camel sequence, and less than three minutes on all our other sequences. The preprocessing stage leading to input meshes is an implementation of GVC that adds approximately 12 minutes to each frame, with 3-8 seconds for Marching Cubes. GVC is not part of our contribution, and can be exchanged for other dense stereo or silhouette-carving algorithms, some of which may, though we have not yet tested this, have superior run-time performance without impacting quality. We have data of other example subjects that will be posted on our website, and the volumetric data has already been shared with other researchers.

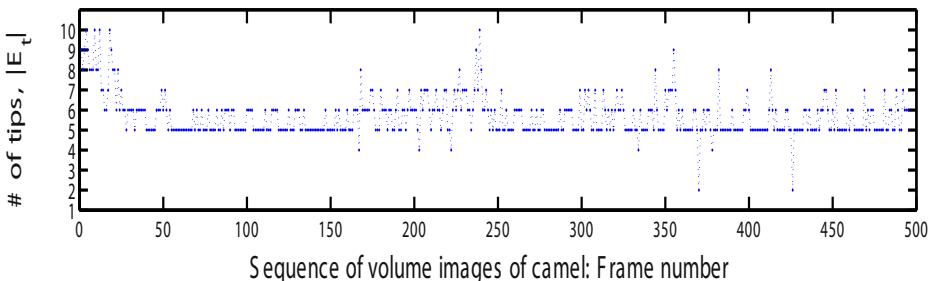


Fig. 7. Number of skeleton tips found per-frame during greedy search.

6 Conclusion and Future Work

We have proposed *spines* as a novel 3D spatio-temporal representation for sequences of volume images. This shape and motion descriptor introduces a method for imposing temporal correspondence on limb topologies when dealing with articulated subjects. We also present an algorithm for efficiently extracting branching spines from surface data. Finally, we have presented example data where the temporally integrated canonical graph improves the quality of individual skeletons.

Where the current fully bottom-up work leaves off, extensions are planned that will allow a prior skeleton estimate to be forced on the data. This will especially apply to meshes where the limbs tuck in or become genus 1+. While the current results reflect that fairly noisy data, without priors, still reveals the real end effectors and underlying structure, further work is needed to track pose even in very poor data.

Acknowledgements. The authors are grateful to Greg Slabaugh and Hewlett-Packard Laboratories for his assistance and for sharing their GVC code. Data capture and processing was possible thanks to the assistance provided by Jonathan Shaw, Steve Park, Stephen Du, Anil Rohatgi, and the indomitable Spencer Reynolds. We also thank Bella Steedly as the baby, Hilary and Davis King for bringing their dog Barnaby, and Odest Chadwicke Jenkins, Quynh Dinh, and the anonymous reviewers.

References

1. ARUN, K. S., HUANG, T. S., AND BLOSTEIN, S. D. 1987. Least squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9*, 2 (March), 698–700.
2. ATTALI, D. AND MONTANVERT, A. 1997. Computing and simplifying 2d and 3d continuous skeletons. *Computer Vision and Image Understanding* **67**(3): 261–273.
3. BETELU, S., SAPIRO, G., TANNENBAUM, A., GIBLIN, P. J. 2000. Noise-resistant affine skeletons of planar curves, *ECCV00*, pp. I: 742–754.
4. BINFORD, T. 1987 (first presented in 1971). Generalized cylinder representation, *Encyclopedia of A. I.*, John Wiley & Sons, pp. 321–323.
5. BLUM, H. 1973. Biological shape and visual science (part I), *Journal of Theoretical Biology* **38**: 205–287.
6. BRADKSI, G., AND PISAREVSKY, V. 2000. Intel’s computer vision library: Applications in calibration, stereo, segmentation, tracking, gesture, face, and object recognition. In *Proceedings of IEEE CVPR 2000*, vol. II, II:796–797. Demonstration Paper.
7. CAO, Y. 2003. *Axial Representations of 3D Shapes*, PhD thesis, Brown University.
8. CAO, Y., AND MUMFORD, D. 2002. Geometric structure estimation of axially symmetric pots from small fragments, *Proc. IASTED SPPRA*.
9. CHU, C., JENKINS, O., AND MATARIC, M. 2003. Markerless kinematic model and motion capture from volume sequences, *CVPR03*, pp. II: 475–482.

10. CORMEN, T. H., LEISERSON, C. E., AND RIVEST, R. L. 1990. *Introduction to Algorithms*. MIT Press/McGraw-Hill.
11. CULBERTSON, W. B., MALZBENDER, T., AND SLABAUGH, G. 1999. Generalized voxel coloring. In *ICCV Vision Algorithms Workshop*, Springer-Verlag, no. 1883 in LNCS, 100–115.
12. DEY, T. K., AND ZHAO, W. 2002. Approximate medial axis as a voronoi subcomplex, *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications*, ACM Press, pp. 356–366.
13. FERLEY, E., CANI, M.-P., AND ATTALI, D. 1997. Skeletal reconstruction of branching shapes, *Computer Graphics Forum* **16**(5): 283–293.
14. HASTIE, T., AND STUETZLE, W. 1989. Principal curves, *Journal of the American Statistical Association* **84**: 502–516.
15. HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 203–212.
16. HUBBARD, P. M. 1996. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics* **15**, 3 (July), 179–210.
17. KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts, *ACM Transactions on Graphics* **22**.
18. KÉGL, B., KRZYŻAK, A., LINDER, T., AND ZEGER, K. 2000. Learning and design of principal curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(3): 281–297.
19. LEYMARIE, F. F., AND KIMIA, B. B. 2001. The shock scaffold for representing 3d shape, in G. S. d. B. C. Arcelli, L.P. Cordella (ed.), *Visual Form 2001*, number LNCS 2059 in *Lecture Notes in Computer Science*, Springer-Verlag, pp. 216–229.
20. LI, X., TOON, T. W., AND HUANG, Z. 2001. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 Symposium on Interactive 3D graphics*, ACM Press, 35–42.
21. MARR, D., AND NISHIHARA, H. 1978. Representation and recognition of the spatial organization of three-dimensional shapes, *Proc. of the Royal Society of London, series B*, Vol. 200, pp. 269–294.
22. NAIN, D., HAKER, S., KIKINIS, R., AND GRIMSON, W. E. L. 2001. An interactive virtual endoscopy tool, *Workshop on Interactive Medical Image Visualization and Analysis satellite symposia of MICCAI, IMIVA'01*, Utrecht, The Netherlands.
23. SEBASTIAN, T. B., KLEIN, P. N., AND KIMIA, B. B. 2001. Recognition of shapes by editing shock graphs. In *ICCV*, I: 755–762.
24. SIDDIQI, K., BOUIX, S., TANNENBAUM, A., AND ZUCKER, S. W. 2002. Hamilton-jacobi skeletons. *IJCV* **48**, 3 (July/August), 215–231.
25. TEICHMANN, M., AND TELLER, S. 1998. Assisted articulation of closed polygonal models. In *Proceeding of Eurographics Workshop on Computer Animation and Simulation 1998*.
26. VEDULA, S., BAKER, S., SEITZ, S., AND KANADE, T. 2000. Shape and motion carving in 6D. In *Proceedings of Computer Vision and Pattern Recognition (CVPR2000)*, 592–598.
27. VERROUST, A., AND LAZARUS, F. 2000. Extracting skeletal curves from 3d scattered data. *The Visual Computer* **16**, 1.
28. WADE, L., AND PARENT, R. E. 2002. Automated generation of control skeletons for use in animation. *The Visual Computer* **18**(2): 97–110.
29. ZHANG, Z. 1998. A flexible new technique for camera calibration. Tech. Rep. 98-71, Microsoft Research. www.research.microsoft.com/~zhang/Calib/.

An Accuracy Certified Augmented Reality System for Therapy Guidance

S. Nicolau^{2,1}, X. Pennec¹, L. Soler², and N. Ayache¹

- ¹ INRIA Sophia, Epidaure, 2004 Rte des Lucioles, F-06902 Sophia-Antipolis Cedex
`{Stephane.Nicolau,Xavier.Pennec,Nicholas.Ayache}@sophia.inria.fr`
<http://www-sop.inria.fr/epidaure/Epidaure-eng.html>
- ² IRCAD-Hopital Civil, Virtual-surg, 1 Place de l'Hopital, 67091 Strasbourg Cedex
`{stephane.nicolau,luc.soler}@ircad.u-strasbg.fr`

Abstract. Our purpose is to provide an augmented reality system for Radio-Frequency guidance that could superimpose a 3D model of the liver, its vessels and tumors (reconstructed from CT images) on external video images of the patient. In this paper, we point out that clinical usability not only need the best affordable registration accuracy, but also a certification that the required accuracy is met, since clinical conditions change from one intervention to the other. Beginning by addressing accuracy performances, we show that a 3D/2D registration based on radio-opaque fiducials is more adapted to our application constraints than other methods. Then, we outline a lack in their statistical assumptions which leads us to the derivation of a new extended 3D/2D criterion. Careful validation experiments on real data show that an accuracy of 2 mm can be achieved in clinically relevant conditions, and that our new criterion is up to 9% more accurate, while keeping a computation time compatible with real-time at 20 to 40 Hz.

After the fulfillment of our statistical hypotheses, we turn to safety issues. Propagating the data noise through both our criterion and the classical one, we obtain an explicit formulation of the registration error. As the real conditions do not always fit the theory, it is critical to validate our prediction with real data. Thus, we perform a rigorous incremental validation of each assumption using successively: synthetic data, real video images of a precisely known object, and finally real CT and video images of a soft phantom. Results point out that our error prediction is fully valid in our application range. Eventually, we provide an accurate Augmented Reality guidance system that allows the automatic detection of potentially inaccurate guidance.

1 Introduction

The treatment of liver tumors by Radio-Frequency (RF) ablation is a new technique which begins to be widely used [11]. However, the guidance procedure to reach the tumors with the electrode is still made visually with per-operative 2D cross-sections of the patient using either Ultra-Sound (US) or Computed Tomography (CT) images. Our purpose is to build an augmented reality system that

could superimpose reconstructions of the 3D liver and tumors onto video images in order to improve the surgeon's accuracy during the guidance step. According to surgeons, the overall accuracy of such a system has to be less than 5 mm to provide significant help.

In our setup, a CT-scan of the patient is acquired just before the intervention (RF is a radiological act), and an automatic 3D-reconstructions of his skin, his liver and the tumors is performed [2]. Two cameras (jointly calibrated) are viewing the patient's skin from two different points of view. The patient is intubated during the intervention, so the volume of gas in his lungs can be controlled and monitored. Then, it is possible to fix the volume at the same value during a few seconds repetitively and to perform the electrode's manipulation almost in the same volume's condition than the one obtained during the preliminary CT-scan. Balter [1] and Wong [20] indicates that the mean tumor repositioning at exhalation phase in a respiratory-gated radiotherapy context is under 1 mm. Thus, it is reasonable to assume that a rigid registration is sufficient to register accurately the 3D-model extracted from the CT with the 2D video images.

Critical issues for computer-guided therapy systems are accuracy and reliability. Indeed, the surgeon has no other source of information than the augmented reality system during the guidance step: he has to rely fully on it. As many parameters can change from one intervention to the other (angle between the cameras, cameras focal, curvature of the patient abdomen), the accuracy provided can sharply vary. For instance, in a point-based registration context, there can be a factor two on the accuracy when the cameras angle goes from 20° to 60° [12]. In accordance with this fact, we cannot afford providing a system without assessing its accuracy during any possible intervention. Consequently, we need to tackle both the system accuracy and the capability to assess its value *before* the intervention. Moreover, every gain in accuracy may be exploited to release some constraints in the system setup (position of the cameras, ergonomics, computation time...).

To answer these requirements, we review in Section 2 the existing registration techniques, and we focus more particularly on 3D/2D points based methods. As their statistical assumptions are not fully satisfied in our application (our 3D point measurements cannot be considered as noise-free), we derive a new criterion that extends the classical one. Experimental results on synthetic and phantom data show that it provides a registration up to 20% more accurate. To be able to quantify online this accuracy, we apply in Section 3 the general theory of error propagation to our new criterion and its standard version. This gives us an *analytical* formulation of the covariance matrix of the sought transformation. But this is only the first part of the job: we then need to validate this prediction w.r.t. the statistical assumptions used to derive the theoretical formula (small non-linearity of the criterion, perfect calibration, unbiased Gaussian noise on points, etc.). Incremental tests with synthetic data, real cameras, and finally real data of a soft phantom, show that our prediction is reliable for our current setup, but may require the inclusion of calibration and skin motion errors if it was to become more accurate.

2 A New 3D/2D Point-Based Registration Criterion

This section aims at finding the most accurate registration method for our application.

2.1 Surfacic, Iconic, 3D/3D, or 3D/2D Registration?

Surface and iconic registration using mutual information have been used to register the 3D surface of the face to either video images [19] or another 3D surface acquired with a laser range scanner [5]. In both cases, thanks to several highly curved parts on the model (nose, ears, eyes), the reported accuracy was under 5 mm. We believe that in our case, the “cylindrical” shape of the human abdomen is likely to lead to much larger uncertainties along the cranio-caudal axis.

Landmarks 3D/3D or 3D/2D registration can be performed when several precisely located points are visible both in the 3D-model and in the video images. Since the landmarks are really homologous, the geometry of the underlying abdomen surface is not any more a problem. As there are no visible anatomical landmarks in our case, we chose to stick to the patient skin some radio-opaque markers that are currently localized interactively (an automatic segmentation is currently being tested). The matching is performed thanks to epipolar geometry between video points, and using a prediction/verification (alignment) algorithm between video and CT points.

As our system is based on a stereoscopic video acquisition, one could think of using a stereoscopic reconstruction. In our case, the main problem is the possible occlusion of some 2D points in one of the cameras, which would lead to discarding the information provided by this point in the other camera. Moreover, one would need to compute non-isotropic uncertainty of the reconstructed 3D points [8] to optimize a 3D/3D registration criterion fitting well the statistical assumptions. Thus, we believe that it is better to rely on LSQ 3D/2D registration criteria.

The 3D/2D registration problem was largely considered in a wide variety of cases. Briefly, we can classify the different methods in 3 groups: closed-form, linear and non-linear. The two first method classes were proposed in the last decades to find the registration as quickly as possible to fulfill real-time constraints [6, 3,7]. However they are very sensitive to noise because they assume that data points are exact, contrary to non-linear method. Consequently, non-linear methods provides better accuracy results [10,18]. As the accuracy is crucial in our application, we think that a LSQ criterion optimization has a definite advantage among the other methods because it can take into account the whole information provided by the data. However, all of the existing methods [4,9,15,10] implicitly consider that 2D points are noisy, but that 3D points of the model to register are exact. In our case, this assumption is definitely questionable, which lead to the development of a new maximum likelihood (ML) criterion generalizing the standard 3D/2D LSQ criterion.

2.2 Maximum Likelihood 3D/2D Registration

Notations. Let M_i ($i \in \{1 \dots N\}$) be the 3D points that represent the exact localization of the radio-opaque fiducials in the CT-scan reference frame and $m_i^{(l)}$ be the 2D points that represent its exact position in the images of camera (l) . To account for occlusion, we use a binary variable ξ_i^l equal to 1 if M_i is observed in camera (l) and 0 otherwise. We denote by $\langle \cdot | \cdot \rangle$ the cross-products, by $T \star M$ the action of the rigid transformation T on the 3D point M and by P_l ($1 \leq l \leq S$) the camera's projective functions from 3D to 2D such that $m_i^{(l)} = P^{(l)}(T \star M_i)$. In the following sections, \hat{A} will represent an *estimation* of a perfect data A , and \tilde{A} will represent an *observed measure* of a perfect data A .

Standard Projective Points Correspondences (SPPC) Criterion. Assuming that the 3D points are exact ($\tilde{M}_i = M_i$) and that the 2D points only are corrupted by an isotropic Gaussian noise η_i of variance σ_{2D}^2 , the probability of measuring the projection of the 3D point M_i at the location $\tilde{m}_i^{(l)}$ in image (l) , knowing the transformation parameters $\theta = \{T\}$ is given by:

$$p(\tilde{m}_i^{(l)} | \theta) = \frac{1}{2\pi\sigma_{2D}^2} \cdot \exp\left(-\frac{\|P^{(l)}(T \star M_i) - \tilde{m}_i^{(l)}\|^2}{2 \cdot \sigma_{2D}^2}\right)$$

Let χ be the data vector regrouping all the measurements, in this case the 2D points $\tilde{m}_i^{(l)}$ only. Since the detection of each point is performed independently, the probability of the observed data is $p(\chi | \theta) = \prod_{l=1}^S \prod_{i=1}^N p(\tilde{m}_i^{(l)} | \theta) \xi_i^l$. In this formula, unobserved 2D points (for which $\xi_i^l = 0$) are implicitly taken out of the probability. Now, the *Maximum likelihood* transformation $\hat{\theta}$ maximizes the probability of the observed data, or equivalently, minimizes its negative log:

$$C_{2D}(T) = \sum_{l=1}^S \sum_{i=1}^N \xi_i^l \cdot \frac{\|P^{(l)}(T \star M_i) - \tilde{m}_i^{(l)}\|^2}{2 \cdot \sigma_{2D}^2} + \left(\sum_{l=1}^S \sum_{i=1}^N \xi_i^l \right) \cdot \log[2\pi\sigma_{2D}^2] \quad (1)$$

Thus, up to a constant factor, this ML estimation boils down to the classical 3D/2D points LSQ criterion.

Extended Projective Points Correspondences (EPPC) Criterion. To introduce a more realistic statistical hypothesis on the 3D data, it is thus safer to consider that we are measuring a noisy version of the exact points: $\tilde{M}_i = M_i + \varepsilon_i$ with $\varepsilon_i \sim N(0, \sigma_{3D})$.

In this case, the exact location M_i of the 3D points is considered as a parameter, just as the transformation T . In statistics, this is called a *latent or hidden variable*, while it is better known as an *auxiliary variable* in computer vision. Thus, knowing the parameters $\theta = \{T, M_1, \dots, M_N\}$, the probability of measuring respectively a 2D and a 3D point is:

$$p(\tilde{m}_i^{(l)} | \theta) = G_{\sigma_{2D}}\left(P^{(l)}(T \star M_i) - \tilde{m}_i^{(l)}\right) \quad \text{and} \quad p(\tilde{M}_i | \theta) = G_{\sigma_{3D}}\left(M_i - \tilde{M}_i\right).$$

One important feature of this statistical modeling is that we can safely assume that all 3D and 2D measurements are independent. Thus, we can write the probability of our observation vector $\chi = (\tilde{m}_1^1, \dots, \tilde{m}_N^1, \dots, \tilde{m}_1^S, \dots, \tilde{m}_N^S, \tilde{M}_1, \dots, \tilde{M}_N)$ as the product of the above individual probabilities. The ML estimation of the parameters is still given by the minimization of $-\log(p(\chi|\theta))$:

$$C(T, M_1, \dots, M_N) = \sum_{i=1}^N \frac{\|\tilde{M}_i - M_i\|^2}{2 \cdot \sigma_{3D}^2} + \sum_{l=1}^S \sum_{i=1}^N \xi_i^l \cdot \frac{\|\tilde{m}_i^{(l)} - P^{(l)}(T * M_i)\|^2}{2 \cdot \sigma_{2D}^2} + K$$

where K is a normalization constant depending on σ_{2D} and σ_{3D} . The convergence is insured since we minimize the same positive criterion at each step.

The obvious difference between this criterion and the simple 2D ML is that we now have to solve for the hidden variables (the exact locations M_i) in addition to the previous rigid transformation parameters. An obvious choice to modify the optimization algorithm is to perform an alternated minimization w.r.t. the two groups of variables, starting from a transformation initialization T_0 , and an initialization of the M_i with the \tilde{M}_i . The algorithm is stopped when the distance between the last two estimates of the transformation become negligible.

Discussion. We highlight in [12] that the EPPC can be viewed as a generalization of either the standard criterion (when $\sigma_{3D} \rightarrow 0$), or a stereoscopic points reconstruction followed by a 3D/3D registration (if σ_{3D} is largely overestimated). A quantitative study on synthetic data showed that accuracy gain brought by EPPC depends essentially on the angle between the cameras and ratio of 2D and 3D SNR [12]. For instance, with data simulating our clinical conditions, EPPC brings up to 10% gain accuracy if the cameras angle is 50° and 18% if the angle is 20° . Finally, as simulation does not take into account calibration errors and possible noise modeling errors, we made a careful validation on real data from a phantom. This showed that a mean accuracy of 2 mm can be reached with a maximum error of 4 mm (obtained when the parameters configuration are not optimal: weak angle between the cameras and/or markers occlusion) and that we can rely on an accuracy gain of 9% with computations time that can still fulfill real-time constraints.

3 Theoretical Uncertainty and Prediction Validation

Now that we have provided a criterion that perfectly fulfills the statistical conditions of our application, we still face the problem of the varying accuracy w.r.t. the various system parameters. In order to propose a safe product to radiologists, we should provide a statistical study that would give the mean *Target Registration Error* (TRE) w.r.t. the number of markers, the angle between the cameras, the focus, and the relative position of the target w.r.t. the markers. This is the equivalent of the direction for use and the secondary effects list mandatory for all proposed drugs in the therapeutic field, and the reliability and accuracy tables of robotics tools: these tables give a usability range to assess under which condition a particular feature (for example accuracy) could be reached.

As increasing the number of experiments is very expensive and time-consuming, it is almost infeasible to *measure* the accuracy provided for each experimental condition. Moreover, as we want a real-time system, the conditions may change during the operation (e.g. markers can be occluded by the radiologist), and the accuracy assessment has to be constantly updated to avoid a potentially dangerous gesture. Consequently, we think that predicting the TRE by studying the theoretical noise propagation is the best way to ensure the safety of our system.

3.1 Uncertainty Propagation through SPPC and EPPC Criteria

In the sequel, we firstly remind the general covariance propagation theory through a criterion. Then, following the methodological framework introduced in [14,13], we present for SPPC and EPPC analytical formulations of the transformation covariance matrix.

General Theory of Error Propagation. Let the criterion $C(\chi, \theta)$ be a smooth function of the data vector χ and the parameters θ . We are looking for the optimal parameter vector $\hat{\theta} = \arg \min_{\theta} (C(\chi, \theta))$. A local minima is reached and well defined if $\Phi(\chi, \theta) = (\frac{\partial C}{\partial \theta}(\chi, \theta))^T = 0$ and $H = \frac{\partial^2 C}{\partial \theta^2}(\chi, \theta)$ is positive definite. The function Φ defines $\hat{\theta}$ as an implicit function of χ . A Taylor expansion gives:

$$\Phi(\chi + \delta\chi, \theta + \delta\theta) = \Phi(\chi, \theta) + \frac{\partial \Phi}{\partial \chi} \cdot \delta\chi + \frac{\partial \Phi}{\partial \theta} \cdot \delta\theta + O(\delta\chi^2, \delta\theta^2)$$

which means that around an optimum $\hat{\theta}$ we have:

$$\hat{\theta}(\chi + \delta\chi) = \hat{\theta}(\chi) - \left(\frac{\partial \Phi}{\partial \theta} \right)^{(-1)} \cdot \frac{\partial \Phi}{\partial \chi} \cdot \delta\chi + O(\delta\chi^2)$$

Thus, if χ is a random vector of mean $\bar{\chi}$ and covariance $\Sigma_{\chi\chi}$, the optimal vector $\hat{\theta}$ is (up to the second order), a random vector with mean $\bar{\theta} = \arg \min_{\theta} (C(\bar{\chi}, \theta))$ and covariance $\Sigma_{\theta\theta} = H^{-1} \left(\frac{\partial \Phi}{\partial \chi} \right) \Sigma_{\chi\chi} \left(\frac{\partial \Phi}{\partial \chi} \right)^T H^{-1}$. Thus, to propagate the covariance matrix from the data to the parameters optimizing the criterion, we need to compute $H = \frac{\partial^2 C(\chi, \theta)}{\partial \theta^2}$, $J_{\Phi_\chi} = \frac{\partial^2 C(\chi, \theta)}{\partial \chi \partial \theta}$ and $\Gamma = J_{\Phi_\chi} \cdot \Sigma_{\chi\chi} \cdot J_{\Phi_\chi}^T$.

SPPC Transformation Covariance. Our analytical analysis needs the block-decomposition of the 3×4 projection matrix 3 as shown below:

$$P^{(l)} = \left[\begin{array}{c|c} Q_{2 \times 3}^{(l)} & b_{2 \times 1}^{(l)} \\ \hline C_{(l)}^T & 1 \end{array} \right] \quad \text{so that} \quad m_i^{(l)} = P^{(l)}(T \star M_i) = \frac{Q^{(l)} \cdot (T \star M_i) + b^{(l)}}{1 + C_{(l)}^T \cdot (T \star M_i)}$$

The second order derivatives H and $J_{\Phi_\chi}^T$ are computed using the chain rule, and after some calculations, the uncertainty of the transformation may be summarized as $\Sigma_{TT} = H^{-1} \cdot \Gamma \cdot H^{-1}$ with

$$\Gamma = \sum_{i=1}^N D_i^T (\sigma_{3D}^2 \cdot K_i \cdot K_i + L_i) \cdot D_i \quad \text{and} \quad H = \sum_{i=1}^N D_i^T \cdot K_i \cdot D_i$$

$$\text{where } D_i = \frac{\partial(T \star M_i)}{\partial T}, \quad L_i = \sum_{l=1}^S \xi_i^l \cdot \frac{(Q - m_i^{(l)} \cdot C_{(l)}^T)^T \cdot (Q - m_i^{(l)} \cdot C_{(l)}^T)}{\sigma_{2D}^2 \cdot (1 + <C_{(l)}|T \star M_i>)^2} \quad \text{and}$$

$$K_i = L_i - \sum_{l=1}^S \xi_i^l \cdot \frac{C_{(l)} \cdot (m_i^{(l)} - \bar{m}_i^{(l)})^T \cdot (Q^{(l)} - m_i^{(l)} \cdot C_{(l)}^T) + (Q^{(l)} - m_i^{(l)} \cdot C_{(l)}^T)^T \cdot (m_i^{(l)} - \bar{m}_i^{(l)}) \cdot C_{(l)}^T}{\sigma_{2D}^2 \cdot (1 + <C_{(l)}|T \star M_i>)^2}$$

EPPC Transformation Covariance. For this case the calculations are not usual because the vector of sought parameters is $\theta = (T, M_1 \dots M_N)$ so that:

$$\delta\theta = \begin{bmatrix} \delta T \\ \delta M \end{bmatrix} = - \left(\frac{\partial \Phi}{\partial \theta} \right)^{-1} \cdot \frac{\partial \Phi}{\partial \chi} \cdot \delta\chi = - \begin{bmatrix} \frac{\partial \Phi_T}{\partial T} & \frac{\partial \Phi_T}{\partial M} \\ \frac{\partial \Phi_M}{\partial T} & \frac{\partial \Phi_M}{\partial M} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \frac{\partial \Phi_T}{\partial \chi} \\ \frac{\partial \Phi_M}{\partial \chi} \end{bmatrix} \cdot \delta\chi$$

Since we only focus on the covariance of the transformation T alone, we need to extract Σ_{TT} from $\Sigma_{\theta\theta} = \begin{bmatrix} \Sigma_{TT} & \Sigma_{TM} \\ \Sigma_{MT} & \Sigma_{MM} \end{bmatrix}$. This is done using a block matrix inversion, and after long calculations, we end up with $\Sigma_{TT} = H^{-1} \cdot \Omega \cdot H^{-1}$, where:

$$\Omega = \sum_{i=1}^N D_i^T (\sigma_{3D}^2 Id + K_i^{-1})^{-1} \cdot (\sigma_{3D}^2 Id + K_i^{-1} \cdot L_i \cdot K_i^{-1}) \cdot (\sigma_{3D}^2 Id + K_i^{-1})^{-1} \cdot D_i$$

$$H = \sum_{i=1}^N D_i^T \cdot (\sigma_{3D}^2 Id + K_i^{-1})^{-1} \cdot D_i$$

One can check that for the limit case where $\sigma_{3D} = 0$, the transformation uncertainty given by Σ_{TT} is equal for both criteria.

Target Registration Error (TRE). Finally, to obtain the final covariance matrix on a target point C_i after registration, we simply have to propagate the uncertainty through the transformation action: $\Sigma_{T \star C_i} = \frac{\partial(T \star C_i)}{\partial T} \cdot \Sigma_{TT} \cdot \frac{\partial(T \star C_i)}{\partial T}^T$.

3.2 Validation of the Prediction

With the previous formulas, we are able to predict the accuracy of the transformation after the convergence of the algorithm of Section 2. But this is only one part of the job: we now have to validate the statistical assumptions used to derive the theoretical formula (small non-linearity of the criterion, perfect calibration, unbiased Gaussian noise on points, etc.). The goal of this section is to verify incrementally that these assumptions hold within our application domain. This will be done using synthetic data (for the non-linearities of the criterion), real video images of a precisely defined 3D object (for camera calibration and distortions), and finally real CT and video images of a soft phantom of the abdomen (for noise assumptions on point measurements).

Synthetic Data. Experiments are realized with two synthetic cameras jointly calibrated with a uniform angle from 5° to 120° , and focusing on 7 to 25 points M_i randomly distributed in a volume of about $30 \times 30 \times 30 \text{ cm}^3$. The cameras are located at a distance of 20 to 50 times the focal length. We add

Table 1. Validation of the uncertainty prediction with 20000 registrations.

	Mean μ^2 (3.0)	Var μ^2 (6.0)	KS-test ($p > 0.01$)
SPPC	3.020	6.28	0.353
EPPC	3.016	6.18	0.647

to the 2D and 3D points a Gaussian noise with σ varying from 0.5 to 4.0 (which corresponds to a SNR of 60 dB to 90 dB¹). The registration error is evaluated using control points C_i to assess a TRE instead of a *Fiducial Localization Error* (FLE).

Since each experiment is different, we need to evaluate the relative fit of the *Predicted TRE* (PTRE) vs. the *Experimental TRE* (ETRE) to quantitatively measure the quality of the uncertainty prediction. Due to the significant anisotropy, we did not use the basic ratio $ETRE^2/PTRE^2$, but rather the validation index [14], which weights the observed error vector with the inverse of its predicted covariance matrix to yield a Mahalanobis distance μ^2 . Assuming a Gaussian error on test points after registration, this validation index should follows a χ_3^2 law. Repeating this experiment with many different “parameters” configurations, we can verify that μ^2 is actually χ_3^2 distributed using the Kolmogorov-Smirnov (K-S) test [16]. We also verify that the empirical mean and variance matches the theoretical ones (resp. 3 and 6 for a χ_3^2 distribution).

Table 1 summarizes the statistics obtained for 20000 registrations where all the parameters randomly vary as previously described. The values obtained for both the validation index and the KS-test fully validate the reliability of the transformation’s accuracy prediction.

Real Calibration and Synthetic Noise. The perfect validation of our accuracy prediction on synthetic data does not take into account possible calibration errors of the cam eras and excludes likely distortions from the pinhole model. The goal of this experiment is to address the validity of these assumptions using a real video system. We used a 54 points calibration grid that allows for a very accurate detection of the points ($\sigma_{3D} < 0.1$ mm, $\sigma_{2D} < 0.2$ pixel). Such an accuracy is obviously far below the current detection of real markers positions ($\sigma_{2D} \simeq 2$ pixel, $\sigma_{3D} \simeq 1$ mm). To simulate the range of variability of our application, we still add a Gaussian noise on the collected data points.

Ideally, the *Experimental TRE* should be assessed by comparing each registration result with a gold-standard that relates both the CT and the camera coordinate systems to the same physical space, using an external and highly accurate apparatus. As such a system is not available, we adapted the registration loops protocol introduced in [13,17], that enables to measure the TRE error for a given set of test points.

The principle is to acquire several couples of 2D images with jointly calibrated cameras so that we can compare *independent* 3D/2D registration of the same object (different 2D and 3D images) using a statistical Mahalanobis distance μ^2 .

¹ $SNR_{dB} = 10 \log_{10}(\frac{\sigma_s}{\sigma_n})$ where σ_s (resp. σ_n) is the variance of the signal (resp. noise).

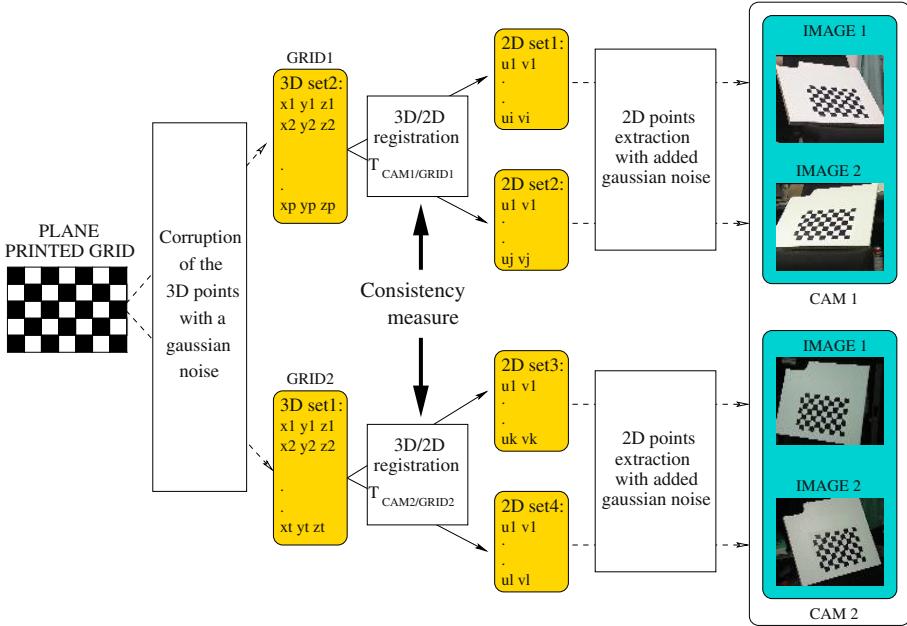


Fig. 1. Registration loops used to estimated the registration consistency: a test point C chosen at a certain distance of the printed grid (typically 20 cm) is transformed into the CAM1 coordinate system using a first 3D/2D registration T_1 , then back into the grid coordinate system using a second 3D/2D registration T_2 provided by the other couple of cameras (the coordinate system of CAM1 and CAM2 are identical since cameras are jointly calibrated). If all transformations were exact, we would obtain the same position for the test point. Of course, since the transformations are not perfect, we measure an error which variance $\sigma_{loop}^2 = 2\sigma_{CAM/GRID}^2$ corresponds to a TRE. In fact, to take into account anisotropies we compute a covariance matrix and a statistical Mahalanobis distance μ^2 between C and $T_1 \star T_2^{-1} \star C$.

A typical loop, sketched in Fig. 1, described the method to get a μ^2 -value. This experiment providing only one error measurement, we still need to repeat it with different datasets to obtain statistically significant measures. In order to take into account possible calibration error and/or bias, it is necessary to change the cameras calibrations and positions, and not only to move the object in the physical space. Likewise, to decorrelate the two 3D/2D transformations, we need to use two differently noised 3D data sets. Indeed, when using the same set of 3D points to register the 2D points, the error on 3D points similarly affects both transformations, and the variability of the 3D points extraction (and any possible bias) is hidden.

Finally, varying each set of parameters (different configuration of our four cameras, different positions/orientations of the calibration grid), we got 144 μ^2 -values. The cameras were placed 10° to 60° apart, at a distance of the object of 25 to 30 times the focal length. Figures 2 shows the mean, standard deviation and K-S test value of the validation index w.r.t. the number of points used

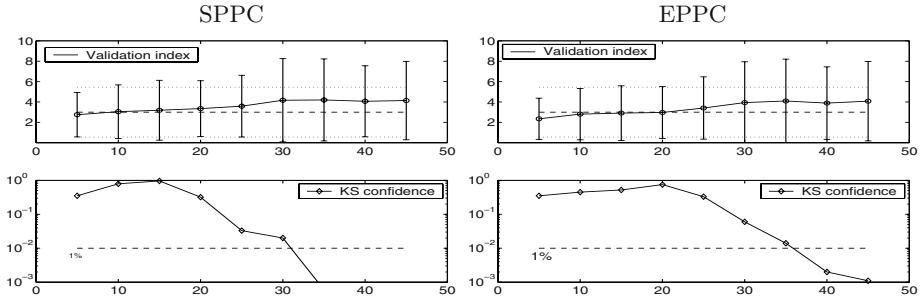


Fig. 2. Validation of the uncertainty prediction on the calibration grid w.r.t. the number of points used for the registration. Top: mean and standard deviation of the validation index. Bottom: KS confidence. Higher scores are more confident.

(randomly chosen among the 54 available). One can see that the prediction is correct up to 40 points (which spans our range of application). This critical value is due to the progressive reduction of the registration error that finally meets the ignored calibration error (about 0.5 mm). Likewise, we observed the same behavior when the feature noise becomes too small (σ_{3D} and σ_{2D} below 0.7).

Real Data (Phantom). To test the last assumption of our prediction (unbiased Gaussian noise), we now turn to a validation experiment on real 3D and 2D measurements of a plastic phantom (designed in [12]), on which are stick about 40 radio-opaque markers (see the incrusted top left image in Fig. 4). The set up is almost the same as for the previous calibration grid (for further details see [12]). However, target points C_i are now randomly chosen within the phantom liver, and markers in the CT and on the video images are interactively localized.

The markers used were randomly chosen among the 40 available, and we obtained 80 μ^2 -values for each experiment. As we experimentally observed that there was a consistent but non-rigid motion of the soft skin (about 1mm), we chose $\sigma_{3D} \simeq 2.0$ mm (instead of 1 mm) to take into account this additional uncertainty. Figure 3 presents the mean and variance of μ^2 w.r.t. the number of points. Firstly, we notice that the mean value slowly increases with the number of points. This can be explained by the biases introduced by the calibration error and the correlated motion of the markers on the skin. Indeed, the measured accuracy figures do not converge to 0 mm with a large number of points but rather towards 1 mm, which corresponds to the motion of the skin.

Nevertheless, it appears that the prediction is well validated for a range of 15 to 25 points. As $\bar{\mu}^2$ can be interpreted as a relative error or the error prediction (see [14]), Fig. 3 shows that we over-estimate the mean TRE by a factor 1.7 for a small number of points ($\bar{\mu}^2 \simeq 1$), and that we under-estimate it by a factor of 1.3 for more than 25 points ($\bar{\mu}^2 \simeq 5$). For our application, in which the number of visible points should not exceed 20, this means that we predict correctly the amplitude of the error on the transformation. In the worst case, we over-estimate it, which can be considered as a good safety measure. One can visually assess

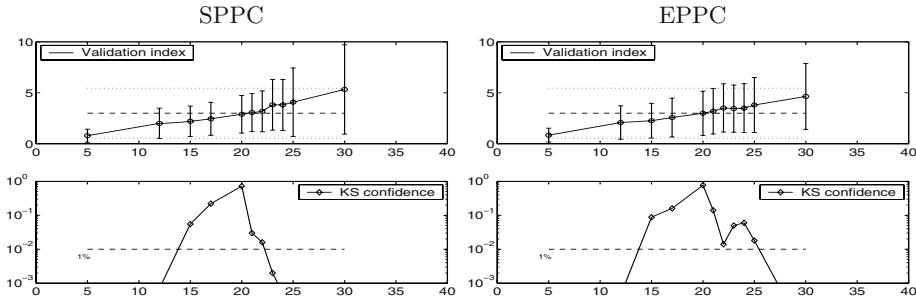


Fig. 3. Validation of the uncertainty prediction on the phantom w.r.t. the number of points used for the registration. Top: mean and standard deviation of the validation index. Bottom: KS confidence. Higher scores are more confident.



Fig. 4. Left image: the top image shows the phantom with radio-opaque markers on its skin. The main image shows the phantom without its skin and we can see the radio-opaque markers on the fake liver. Right image: we superimpose the reconstructions of the fiducials whose predicted accuracy is around 2 mm. One can visually assess the quality of the registration.

the validation of our prediction error on one case among the 160 registrations we performed (Fig.4).

4 Conclusion

We devised in this paper an augmented reality system for Radio-Frequency ablation guidance based on a new 3D/2D registration criterion with a validated error prediction. We argue the necessity to provide not only the best affordable registration accuracy but also an accurate assessment of the TRE for safety consideration.

To reach the best accuracy performances, we firstly derived a new 3D/2D Maximum Likelihood registration criterion (EPPC) based on better adapted statistical hypotheses than the classical 3D/2D least-square registration criterion

(SPPC). Experiments on real data showed that EPPC provides an accuracy of about 2mm within the liver, which fits the initial requirements of less than 5mm. Moreover, EPPC is up to 9% more accurate than SPPC with a refreshment rate that can reach real-time constraints. We underline an alternative interpretation of this gain: we can typically reach the same accuracy with 20 markers for EPPC where 24 are needed for SPPC. As we face possibilities of markers occlusion because of the surgeon's hand and cumbersoness constraints on the placement of the markers, this gain should not be taken with the light one. In addition, as clinical conditions do not allow a free camera positioning, we could meet situation where an angle between the cameras could decrease below 20° , which would mean an accuracy gain of 18%.

In order to assess the system accuracy for all configurations, we propose in a second step a theoretical propagation of the target covariance through SPPC and EPPC w.r.t the experimental configuration parameters. To verify the validity of all the assumptions of that method, we conducted a careful validation study that assess in turn the range of validity of each hypothesis. We firstly verified that non-linearities in the criterion and calibration error are negligible. Then, we use a realistic phantom with a soft and deformable skin to validate the prediction in the range of our application (i.e. for 15 and 25 markers). This study confirmed that we correctly predict the registration error, with a slight over-estimation if too much markers are occluded, which is a good safety rule.

To reach the clinical usability, the whole system still has to be validated on real patients. We are currently conducting experiments (using repeated CT scans at the same point of the breathing cycle) to certify that the motion of the internal structures due to the monitored breathing of the patient cannot bias our accuracy prediction. Preliminary results indicates that this motion is of the order of 1 mm, which is in accordance with the motions we experienced because of the phantom soft skin. Thus, we are pretty confident that our registration error prediction will work properly in the final system. Last but not least, it is possible to estimate broadly the TRE *before* scanning the patient, by using the stereoscopic reconstruction of the markers instead of their positions in the scanner. This will allow a better control of the external conditions (number of markers, angle between the cameras) and the optimization of the intervention preparation.

References

1. J.M. Balter, K.L. Lam, C.J. McGinn, T.S. Lawrence, and R.K. Ten Haken. Improvement of CT-based treatment-planning models of abdominals targets using static exhale imaging. *Int. J. Radiation Oncology Biol. Phys.*, 41(4):939–943, 1998.
2. L. Soler et al. Fully automatic anatomical, pathological, and functional segmentation from CT-scans for hepatic surgery. *Computer Aided Surgery*, 6(3), 2001.
3. M. Dhome et al. Determination of the attitude of 3D objects from a single perspective view. *IEEE Trans. on PAMI*, 11(12):1265–1278, December 1989.
4. R. Haralick et al. Pose estimation from corresponding point data. *IEEE Trans. on Systems, Man. and Cybernetics*, 19(06):1426–1446, December 1989.

5. W. Grimson et al. An automatic registration method for frameless stereotaxy, image-guided surgery and enhanced reality visualization. *IEEE TMI*, 15(2):129–140, April 1996.
6. M. Fischler and R. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Com. of the ACM*, 24(6):381–395, June 1981.
7. S. Ganapathy. Decomposition of transformation matrices for robot vision. *Pattern Recognition Letters*, 2(6):401–412, December 1984.
8. E. Grossmann and J. Santos-Victor. Uncertainty analysis of 3D reconstruction from uncalibrated views. *Image and Vision Computing*, 18:685–696, 2000.
9. Y. Hel-Or and M. Werman. Pose estimation by fusing noisy data of different dimensions. *IEEE Trans. on PAMI*, 17(2):195–201, February 1995.
10. C. Lu, G.D Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Trans. on PAMI*, 22(6):610–622, 2000.
11. J.F. McGahan and G.D. Dodd III. Radiofrequency ablation of the liver: Current status. *American Journal of Roentgenology*, 176(1):3–16, 2001.
12. S. Nicolau, X. Pennec, L. Soler, and N. Ayache. Evaluation of a new 3D/2D registration criterion for liver radio-frequencies guided by augmented reality. In *IS4TM'03*, volume 2673 of *LNCS*, pages 270–283, France, 2003. Springer-Verlag.
13. X. Pennec, C.R.G. Guttmann, and J.-P. Thirion. Feature-based registration of medical images: Estimation and validation of the pose accuracy. In *MICCAI'98*, LNCS 1496, pages 1107–1114, October 1998.
14. X. Pennec and J.-P. Thirion. A framework for uncertainty and validation of 3D registration methods based on points and frames. *IJCV*, 25(3):203–229, 1997.
15. T. Phong, R. Horaud, and P. Tao. Object pose from 2-D to 3-D points and line correspondences. *IJCV*, 15:225–243, 1995.
16. W.H. Press, B.P. Flannery, S.A Teukolsky, and W.T. Vetterling. *Numerical Recipices in C*. Cambridge Univ. Press, 1991.
17. A. Roche, X. Pennec, G. Malandain, and N. Ayache. Rigid registration of 3D ultrasound with MR images: a new approach combining intensity and gradient information. *IEEE TMI*, 20(10):1038–1049, 2001.
18. J. Salvi, X. Armangu, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617–1635, 2002.
19. P. Viola and W.M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.
20. J. Wong, M. Sharpe, D. Jaffray, V. Kini, J. Robertson, J. Stromberg, and A. Martinez. The use of active breathing control (ABC) to reduce margin for breathing motion. *Int. J. Radiation Oncology Biol. Phys.*, 44(4):911–919, 1999.

3D Human Body Tracking Using Deterministic Temporal Motion Models

Raquel Urtasun and Pascal Fua*

Computer Vision Laboratory

EPFL

CH-1015 Lausanne, Switzerland

{raquel.urtasun,pascal.fua}@epfl.ch

Abstract. There has been much effort invested in increasing the robustness of human body tracking by incorporating motion models. Most approaches are probabilistic in nature and seek to avoid becoming trapped into local minima by considering multiple hypotheses, which typically requires exponentially large amounts of computation as the number of degrees of freedom increases.

By contrast, in this paper, we use temporal motion models based on Principal Component Analysis to formulate the tracking problem as one of minimizing differentiable objective functions. The differential structure of these functions is rich enough to yield good convergence properties using a deterministic optimization scheme at a much reduced computational cost. Furthermore, by using a multi-activity database, we can partially overcome one of the major limitations of approaches that rely on motion models, namely the fact they are limited to one single type of motion.

We will demonstrate the effectiveness of the proposed approach by using it to fit full-body models to stereo data of people walking and running and whose quality is too low to yield satisfactory results without motion models.

1 Introduction

In recent years, much work has been devoted to increasing the robustness of people tracking algorithms by introducing motion models. Most approaches rely on probabilistic methods, such as the popular CONDENSATION algorithm [1, 2], to perform the tracking. While effective, such probabilistic approaches require exponentially large amounts of computation as the number of degrees of freedom in the model increases, and can easily become trapped into local minima unless great care is taken to avoid them [3,4,5,6].

By contrast, in this paper, we use temporal motion models based on Principal Component Analysis (PCA) and inspired by those proposed in [7,8,9] to formulate the tracking problem as one of minimizing differentiable objective functions.

* This work was supported in part by the Swiss National Science Foundation.

Our experiments show that the differential structure of these objective functions is rich enough to take advantage of standard *deterministic* optimization methods [10], whose computational requirements are much smaller than those of *probabilistic* ones and can nevertheless yield very good results even in difficult situations. Furthermore, in practice, we could combine both kinds of approaches [5].

We will further argue that we can partially overcome one of the major limitations of approaches that rely on motion-models, namely that they limit the algorithms to the particular class of motion from which the models have been created. This is achieved by performing PCA on motion databases that contain multiple classes of motions as opposed to a single one, which yields a decomposition in which the first few components can be used to classify the motion and can evolve during tracking to model the transition from one kind of motion to another.

We will demonstrate the effectiveness of the proposed approach by using it to fit full-body models to stereo data of people walking and running and whose quality is too low to yield satisfactory results without models. This stereo data simply provides us with a convenient way to show that this approach performs well on real data. However, any motion tracking algorithm that relies on minimizing an objective function is amenable to the treatment we propose. We therefore view the contribution of this paper as the proposed formulation that produces results using a deterministic, as opposed to probabilistic optimization method, which yields good performance at a reduced computational cost.

In the remainder of this paper, we first discuss related approaches and our approach to body and motion modeling. We then introduce our deterministic optimization scheme and show its effectiveness using real data.

2 Related Work

Modeling the human body and its motion is attracting enormous interest in the Computer Vision community, as attested by recent and lengthy surveys [11,12]. However, existing techniques remain fairly brittle for many reasons: Humans have a complex articulated geometry overlaid with deformable tissues, skin and loosely-attached clothing. They move constantly, and their motion is often rapid, complex and self-occluding. Furthermore, the 3-D body pose is only partially recoverable from its projection in one single image. Reliable 3-D motion analysis therefore requires reliable tracking across frames, which is difficult because of the poor quality of image-data and frequent occlusions.

When a person is known *a priori* to be performing a given activity, such as walking or running, an effective means to constrain the search and increase robustness is to introduce a motion model. Of particular interest to us are models that represent motion vectors as linear sums of principal components and have become widely accepted in the Computer Animation community as providing realistic results [13,14,15]. The PCA components are computed by capturing as many people as possible performing a specific activity, for example by means of an optical motion capture system, representing each motion as a temporally quantized vector of joint angles, and performing a Principal Component Analysis on the resulting set of vectors.

In practice, the position of a person, or body pose, in a given image frame can be defined by the position and orientation of a root node and a vector of joint angles. A motion can then be represented by an angular *motion vector*, that is a set of such joint angle vectors measured at regularly sampled intervals. Given a large enough database of motion vectors for different motion classes and the corresponding principal components $\Theta_j, 1 \leq j \leq m$, at a given time t , the joint angle vector $\Theta(\mu_t)$ can then be written as

$$\Theta(\mu_t) = \Theta_0(\mu_t) + \sum_{j=1}^m \alpha_j \Theta_j(\mu_t) \quad \text{with } 0 \leq \mu_t \leq 1 , \quad (1)$$

where μ_t is a normalized temporal variable that indicates to what stage of the motion the pose corresponds, Θ_0 represents an average motion, and the α_j are scalar coefficients. In short, the vector $(\mu_t, \alpha_1, \dots, \alpha_m)$, where m is much smaller than the number of joint angles, can be used as the state vector that completely describes the body pose. Recovering this pose then amounts to minimizing an image-based objective function F with respect to this more compact representation, and can be expected to be much more robust than minimizing it with respect to the full set of joint angles.

This representation has already been successfully used in our community, but almost always in a statistical context [7,8,9] and without exploiting the fact that F is easily differentiable with respect to μ_t and the α_j coefficients of Eq. 1. Here, we propose to use this fact to formulate the fitting problem as a traditional optimization problem with respect to the $(\mu_t, \alpha_1, \dots, \alpha_N)$ state vector. Instead of generating many “particles” by randomly choosing values for the α_j , we will compute the Jacobian of F and use it in conjunction with standard least-squares techniques [16]. Our *deterministic* approach to motion tracking is therefore related to an earlier technique [17] that also uses PCA to model the set of 2-D flow vectors that can be seen in video-sequences of a walking subject and to recognize specific 2-D poses without requiring a probabilistic framework. However, this approach relies on an initial segmentation of the body parts and is viewpoint dependent. By contrast, we fit a global 3-D model to the whole body, which lets us fit over a whole sequence and recover accurate 3-D poses.

3 Models

In this section, we introduce the models we use to describe both body pose and shape at a given time as well as its motion over time.

3.1 Body Model

In earlier work [18], we have developed a body-modeling framework that relies on attaching implicit surfaces, also known as soft objects, to an articulated skeleton. Each primitive defines a field function and the skin is taken to be a level set of the sum of these fields, as shown in Fig. 1a. Defining surfaces in this manner lets us

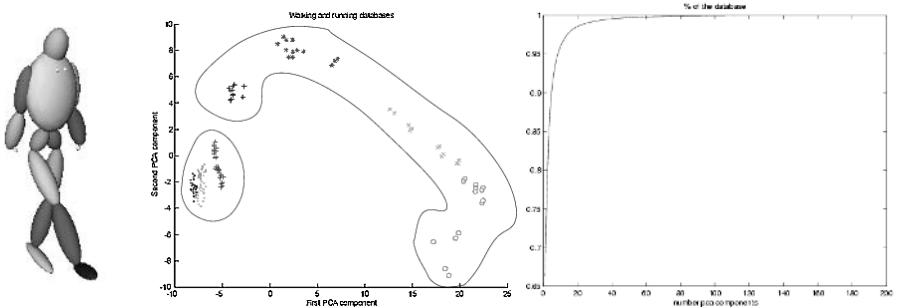


Fig. 1. Shape and motion models. a) Volumetric primitives attached to an articulated skeleton. b) First two PCA components for 4 different captures of 4 subjects walking at speeds varying from 3 to 7km/h, and running at speeds ranging from 6 to 12km/h. The data corresponding to different subjects is shown in different styles. c) Percentage of the database that can be generated with a given number of eigenvectors.

define a distance function of data points to the model that is differentiable. We will take advantage of this to implement our minimization scheme, as discussed in Section 4.

As in Section 2, let us assume that, at a given time, the pose of the skeleton is entirely characterized by the global position and orientation G of a root node and a set of joint angles Θ . To avoid undue blending of primitives, the body is divided into several body parts. Each body part b includes n_b ellipsoidal primitives attached to the skeleton. To each primitive is associated a field function f_i of the form $f_i(G, \Theta, X) = b_i \exp(-a_i d_i(G, \Theta, X))$, where X is a 3-D point, a_i, b_i are constant values, and d_i is the algebraic distance to this ellipsoid. The complete field function for body part b is taken to be

$$f^b(G, \Theta, X) = \sum_{i=1}^{n_b} f_i(X, G, \Theta) , \quad (2)$$

and the skin is the set $\mathcal{S}(G, \Theta) = \bigcup_b \{X \in \mathbb{R}^3 | f^b(G, \Theta, X) = C\}$, where C is a constant. A point X is said attached to body part b if

$$f^b(G, \Theta, X) = \min_{1 \leq i \leq B} |f^i(G, \Theta, X) - C| \quad (3)$$

Fitting the model to stereo-data acquired at time t then amounts to minimizing

$$F_t(G_t, \Theta_t) = \sum_{b=1}^B \sum_{X_t \in b} (f^b(G_t, \Theta_t, X_t) - C)^2 , \quad (4)$$

where the X_t are the 3-D points derived from the data, each one being attached to one of the B body parts. Note that F_t is a differentiable function of the global position G_t and of the joint angles in Θ_t and that its derivatives can be computed fast [18].

3.2 Motion Models

To create a motion database, we used a Vicontm optical motion capture system and a treadmill to capture 4 people, 2 men and 2 women,

- walking at 9 different speeds ranging from 3 to 7 km/h, by increments of 0.5 km/h;
- running at 7 different speeds ranging from 6 to 12 km/h, by increments of 1.0 km/h.

The data was then segmented into cycles and normalized so that each one is represented by the same number of samples. To this end, spherical interpolation in quaternion space was used because it is the space in which a distance measuring the proximity of two orientations can be naturally defined. It therefore lets us interpolate with a meaningful angular velocity measure on an optimal path splining among orientation key frames [19]. Since people never perform the same motion twice in exactly the same fashion, we included in the database four walking or running cycles for each person and speed. The mean motion of the set of examples was subtracted and the M eigenvectors of Eq. 1 were obtained by SVD. Retaining only $m \leq M$ eigenvectors, gave us a reduced base of the most significant subspace of the motion space, that is the one that contains σ % of the database, where λ_i is the i-th bigger eigenvalue.

$$\sigma = \frac{\sum_{i=1}^m \lambda_i}{\sum_{i=1}^M \lambda_i} \quad (5)$$

In our experiments, we chose $\sigma = 0.9$, which means that for the multi-activity database we need only 5 out of 256 coefficients, which corresponds to the total number of examples in the database. In Fig. 1c we display σ as a function of the number of eigenvectors. The same method was used for the walking and running databases independently. The estimation problem is thus reduced from the $\simeq 80$ degrees of freedom for the 28 joints in our body model at each time step, to 5 coefficients plus the time.

Fig. 1b shows the first two PCA components of the original examples used to create the joint walking and running database. The two activities produce separate clusters. The walking components appear on the left of the plot and form a relatively dense set. By contrast, running components are sparser because inter-subject variation is larger, indicating that more examples are required for a complete database.

Note that varying only the first two components along the curve corresponding to the path from one subset to another, yields very natural transitions between walking and running motions.

4 Deterministic Approach to Tracking

In this section we introduce our deterministic approach to tracking that relies on describing motion as a linear combination of the motion eigenvectors of Section 3.2 and choosing optimal weights for these vectors. As before, we represent

the angular component of motion Θ as $\Theta = \Theta_0 + \sum_{i=1}^m \alpha_i \Theta_i$, where Θ_0 is the average motion and the Θ_i are the eigenvectors of Section 3.2. Evaluating Θ at a particular time μ_t yields the pose

$$\Theta(\mu_t) = \Theta_0(\mu_t) + \sum_{i=1}^m \alpha_i \Theta_i(\mu_t) = [\theta^1(\mu_t), \dots, \theta^{ndof}(\mu_t)]^T , \quad (6)$$

where the θ^j are the actual joint angles at time μ_t for the $ndof$ degrees of freedom of the body model we use.

Note that the complete motion is described not only by the angular motion discussed above, but also by the motion G_t of the root body model node with respect to which the angles are expressed. This adds six degrees of freedom to our model, which are not represented at all in our motion database since the data was acquired on a treadmill on which the subjects were forced to walk straight. Furthermore, even if the global motion had been acquired, it would make no sense to include it in the database because similar motions would then have been considered as different just because of the orientation or position of the body.

Let us assume that we have acquired image data, which here is the stereo data depicted by Fig. 2, but could just as well be anything else, in T consecutive frames. Our goal is to recover the motion by minimizing an objective function F over all frames and, therefore, fitting the model to the image data. Tracking is achieved in two main steps. First the global motion G_t is recovered in a recursive way. Results from frame t are used as initialization for frame $t+1$. We initialize using the average motion Θ_0 , positioning the global motion for the first

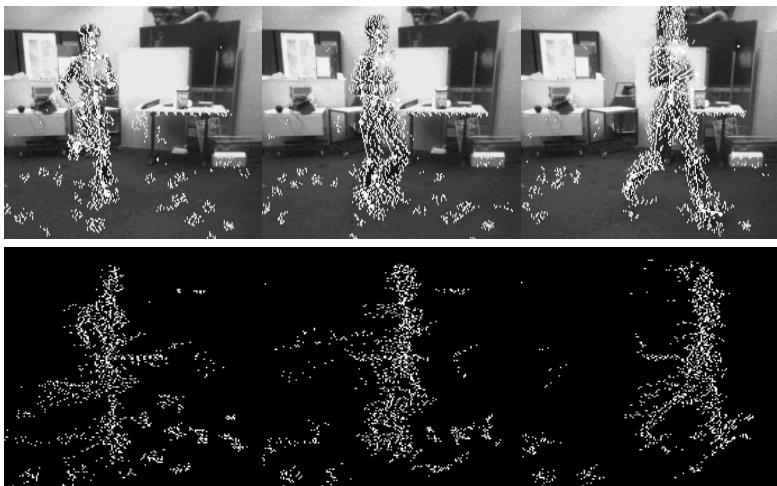


Fig. 2. Input stereo data. Top row: First image of a synchronized trinocular video sequence at three different times. The 3-D points computed by the Digiclops™ system are reprojected onto the images. Bottom row: Side views of these 3-D points. Note that they are very noisy and lack depth because of the low quality of the video sequence.

frame by hand, where the time $\mu_t, 1 \leq t \leq T$, is a linear interpolation between initial values μ_1 and μ_T for the first and last frames. For each frame we minimize $F_t(G_t, \Theta_0(\mu_t, \alpha_i))$ with respect to $G = G(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$, where F_t is defined in equation 4. Given this global motion estimate, we then fit the data over all frames simultaneously by minimizing F with respect to the μ_t , α_i and G_t :

$$F = \sum_{1 \leq t \leq T} F_t(G_t, \Theta(\mu_t, \alpha_i)), \quad (7)$$

In the remainder of this Section, for comparison purposes, we first show the result of fitting the stereo data we use without using motion models. We then introduce in more detail our approach to enforcing the motion models, with or without assuming that the style remains constant. Finally, we discuss the computational requirements of our scheme and contrast them with those of more traditional probabilistic approaches.

4.1 Tracking without Motion Models

In this paper we use stereo data acquired using a Digiclopstm operating at a 640×480 resolution and a 14Hz framerate, which is relatively slow when it comes to capturing a running motion. The quality of the data is poor for several reasons. First, to avoid motion blur, we had to use a high shutter speed that reduces exposure too much. Second, because the camera is fixed and the subject must remain within the capture volume, she appears to be very small at the beginning of the sequence. As a result the data of Fig. 2 is very noisy and lacks both resolution and depth.



Fig. 3. Tracking without a motion model. Given the low framerate, motion between frames is large enough to provoke erroneous attachments of data points to body parts and, as a consequence, very poor fitting behavior. The whole sequence is shown.

To establish a baseline, in Fig. 3, we show the unsatisfactory result of fitting our model to this data without using motion models, that is by minimizing the objective function of Eq. 4 in each frame separately. We simply use the recovered pose in frame $t - 1$ as the starting point in frame t .

Careful analysis shows that tracking fails chiefly due to the low framerate, as the interframe motion is too large. This prevents the process of “attaching” data points to body parts discussed in Section 3.1 from functioning properly. In the fifth image of Fig. 3, both legs end up being “attracted” to the same data points.

4.2 Tracking a Steady Motion

To remedy the problems discussed above, we can first assume that the motion is steady over T data frames and, therefore, that the α_i coefficients of Eq. 6 are invariant. The motion state vector is taken to be

$$\phi = [\vec{\mu}, \vec{\alpha}] = [\mu_1, \dots, \mu_T, \alpha_1, \dots, \alpha_m] \quad (8)$$

To effectively minimize the objective function F of Eq. 7 using a standard least-squares technique [16], we need to evaluate its Jacobian. Bearing in mind that the derivatives of F with respect to the individual joints angles $\frac{\partial F}{\partial \theta_j}$ can be easily computed [18], this can be readily done as follows:

$$\frac{\partial F}{\partial \alpha_i} = \sum_{j=1}^{ndof} \frac{\partial \theta_j}{\partial \alpha_i} \cdot \frac{\partial F}{\partial \theta_j} , \quad \frac{\partial F}{\partial \mu_t} = \sum_{j=1}^{ndof} \frac{\partial \theta_j}{\partial \mu_t} \cdot \frac{\partial F}{\partial \theta_j} . \quad (9)$$

Because the θ_j are linear combinations of the Θ_i eigenvectors, $\frac{\partial \theta_j}{\partial \alpha_i}$ is simply the Θ_{ij} , the j th coordinate of Θ_i . Similarly, we can write

$$\frac{\partial \theta_j}{\partial \mu_t} = \sum_{i=1}^m \alpha_i \frac{\partial \Theta_{ij}}{\partial \mu_t} ,$$

where the $\frac{\partial \Theta_{ij}}{\partial \mu_t}$ can be evaluated using finite differences and stored when building the motion database, as shown in Fig. 4.

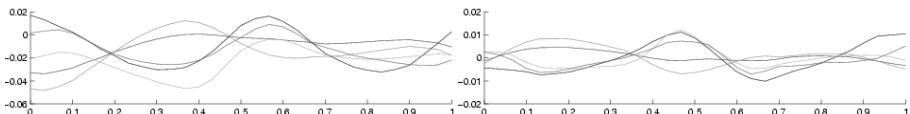


Fig. 4. Motion vector and its temporal derivatives. Left: First 5 eigenvectors for the flexion-extension in the sagittal plane of the left knee. Right: Temporal derivatives $\frac{\partial \Theta_{ij}}{\partial \mu_t}$.

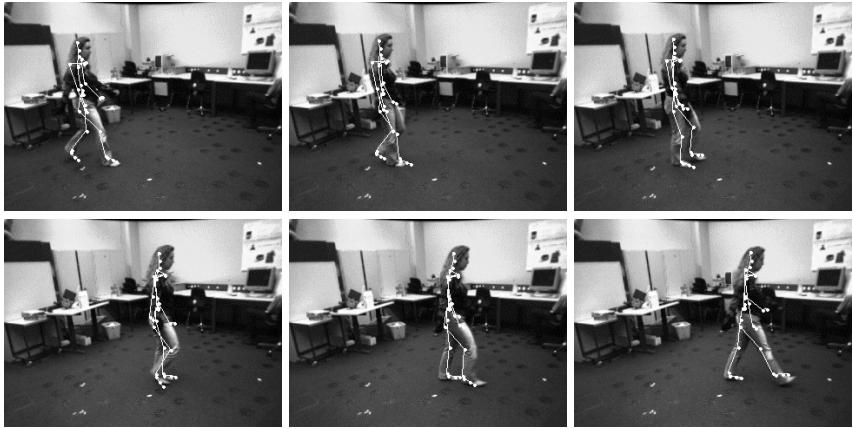


Fig. 5. Using low resolution stereo data to track a woman whose motion was recorded in the database. The recovered skeleton poses are overlaid in white. The legs are correctly positioned.



Fig. 6. Tracking a walking motion assuming a constant style. The legs are correctly positioned.

Figure 5 depicts results on a walking sequence performed by a subject whose motion was captured when building the database. Note that the legs are correctly positioned. The errors in the upper-body are due to the noisiness of the stereo cloud.

Fig. 6 displays the results on a walking sequence performed by a subject who was not recorded when building the database. To validate our results, he is wearing four gyroscopes on his legs, one for each sagittal rotation of the hip and knee joints. The angular speeds they measure are used solely for comparison purposes and we show their integrated values in Fig. 7. We overlay on the corresponding

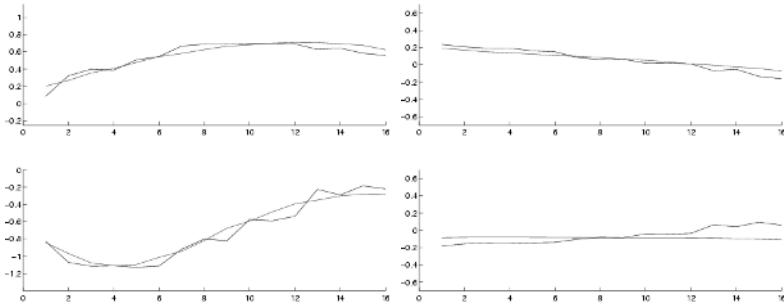


Fig. 7. Comparing recovered rotation angles using visual tracking (solid curve), and by integrating gyroscopic data (smooth curve) for the walk of Fig. 6. Left column: Right hip and knee sagittal rotations. Right Column: Same thing for the left leg. Note that both curves are very close in all plots, even though the left leg is severely occluded.



Fig. 8. Tracking a running motion assuming a constant style. The legs are correctly positioned except the left one in the first frame.

plots the values recovered by our tracker. Note that they are very close, even though the left leg is severely occluded.

Fig. 8 depicts results on the running sequence of Fig. 2 using the running database of Section 3.2, which are much better than those of Fig. 3. The pose of the legs is now correctly recovered, except the one of the left leg in the first frame. This is due in part to the fact that the database was acquired using a treadmill and is therefore too sparse to model a motion in which the leg is raised that high, and in part to the fact that the motion is not truly steady. We address these issues below.



Fig. 9. Tracking a running motion while allowing the style to vary. The legs are now correctly positioned in the whole sequence.

4.3 Tracking a Variable Style and Speed Motion

In the sequences shown in this paper, speed and style are not truly constant. Because of space constraints, the subject starts, accelerates and stops over a short distance. This is true for walking and running, and even more so for transitions from one to the other. Using a single set of α_i parameters for the whole sequence as in Section 4.2 therefore overconstraints the problem. We relax these constraints by introducing a set of α_i per frame or per set of frames and the state vector then becomes:

$$\phi = \phi(\vec{\mu}, \vec{\alpha}^1, \dots, \vec{\alpha}^T) \text{ where } \vec{\alpha}^i = (\alpha_1^i, \dots, \alpha_m^i) . \quad (10)$$

Improved tracking results from the running sequence of Fig. 2 are shown in Fig. 9. The system now has enough freedom to raise the leg in the first frame while still positioning the legs correctly everywhere else. Upper body tracking remains relatively imprecise because average errors in the stereo data are larger than the distance between torso and arms. Improving this would require the use of additional information, such as silhouette information, which could easily be done within the proposed framework. Similar results for walking are shown in Fig. 10. Small errors in foot positioning are due to the fact that ankle flexion has not been recorded in the motion database.

Having a set of PCA parameters per frame gives the system the freedom to automatically evolve from one activity to another. To demonstrate this, in Fig. 11, we use our full motion database to track a transition from walking to running. In the first few frames the subject is walking, then for a couple of frames she performs the transition and runs for the rest of the sequence. The arms are not tracked because we focus on estimating the motion parameters



Fig. 10. Tracking a walking motion while allowing the style to vary. The sequence has a total of 18 frames, we show one in two.

of the lower body only. Here again, the legs are successfully tracked with small errors in foot positioning that are due to the fact that ankle flexion is not part of the motion database.

4.4 Computational Requirements

Probabilistic approaches such as the one of [8] rely on randomly generating “particles” and evaluating their fitness. Assuming the cost of creating the particles to be negligible, the main cost of each iteration comes from evaluating an objective function, such as the function F of Eq. 7 for each particle. In the classical implementation of the condensation, where the state vector has $ndofs$ degrees of freedom, the cost is therefore in the order of $\mathcal{O}(npart(ndofs))$ times the cost of computing F , where $npart$ is the number of particles, which tends to grow very fast if good convergence properties are to be achieved. On the other hand, if we use our motion models to perform the condensation, the cost is of the order of $\mathcal{O}(npart(n))$, which also grows fast with n , the state’s vector dimension.

By contrast, the main cost of each iteration of our optimization scheme comes from evaluating F and its Jacobian, which is of course more expensive than evaluating F alone. However, through careful implementation, we have found



Fig. 11. Tracking the transition between walking and running. In the first four frames the subject is running. The transition occurs in the following three frames and the sequence ends with running. The whole sequence is shown.

that it can be done at a cost in the order of $\mathcal{O}(\lg(ndof))$ times the cost of computing F alone, since evaluating F and its derivatives for the $ndof$ degrees of freedom in the body model involves many similar computations, and computing $\frac{\partial F}{\partial \theta_j}$ once per iteration is what is costly. It took less than 15 iterations to achieve convergence. As a consequence, the cost of the methods of Section 4.2 and 4.3 are of the same order and smaller than the probabilistic approach.

5 Conclusion

We have presented an approach using motion models that allows us to formulate the tracking problem as one of minimizing a differential objective function with respect to relatively few parameters. We take them to be the first few coefficients

of the principal components of the joint angle space for motions captured using an optical motion capture device.

Using walking and running as examples, we have shown that this representation, while having a fairly low dimension, nevertheless has a rich enough differential structure to yield good performance at a low computational cost. It also has the ability to capture the transition from one motion to another.

We have demonstrated that our approach can simultaneously handle two different activities. Our method seems perfectly adapted to 3-D analysis of sport activities such as a golf swing or a tennis serve. The same can be said of capturing the motion of orthopedic patients when they are asked to perform a particular routine designed to evaluate their conditions. Applying our method to such activities will be a subject for future research.

Currently, the major limitation comes from the small size of the database we use, which we will endeavor to complete. This should allow us to precisely track a wider range of styles, perhaps at the cost of adding some regularization constraints that we presently do not need. We also plan to add additional motion types, such as jumping, for which motion capture data is fairly easy to acquire. In the current database, the samples corresponding to different people tend to cluster. If this remains true when the database is completed, this may become a promising approach not only for tracking but also for recognition.

References

1. Isard., M., Blake, A.: CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision* **29** (1998) 5–28
2. Deutscher, J., Blake, A., Reid, I.: Articulated Body Motion Capture by Annealed Particle Filtering. In: CVPR, Hilton Head Island, SC (2000)
3. Davison, A.J., Deutscher, J., Reid, I.D.: Markerless motion capture of complex full-body movement for character animation. In: Eurographics Workshop on Computer Animation and Simulation, Springer-Verlag LNCS (2001)
4. Choo, K., Fleet, D.: People tracking using hybrid monte carlo filtering. In: International Conference on Computer Vision, Vancouver, Canada (2001)
5. Sminchisescu, C., Triggs, B.: Covariance Scaled Sampling for Monocular 3D Body Tracking. In: Conference on Computer Vision and Pattern Recognition, Hawaii (2001)
6. Sminchisescu, C., Triggs, B.: Kinematic Jump Processes for Monocular 3D Human Tracking. In: Conference on Computer Vision and Pattern Recognition, Madison, WI (2003)
7. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3D human figures using 2D image motion. In: European Conference on Computer Vision. (2000)
8. Sidenbladh, H., Black, M.J., Sigal, L.: Implicit Probabilistic Models of Human Motion for Synthesis and Tracking. In: European Conference on Computer Vision, Copenhagen, Denmark (2002)
- 9.Ormoneit, D., Sidenbladh, H., Black, M.J., Hastie, T.: Learning and tracking cyclic human motion. In: Advances in Neural Information Processing Systems 13. (2001) 894–900
10. Bregler, C., Malik, J.: Tracking People with Twists and Exponential Maps. In: Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA (1998)

11. Gavrila, D.: The Visual Analysis of Human Movement: A Survey. *Computer Vision and Image Understanding* **73** (1999)
12. Moeslund, T., Granum, E.: A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding* **81** (2001)
13. Alexa, M., Mueller, W.: Representing animations by principal components. In: Eurographics. Volume 19. (2000)
14. Brand, M., Hertzmann, A.: Style Machines. *Computer Graphics, SIGGRAPH Proceedings* (2000) 183–192
15. Blanz, V., Basso, C., Poggio, T., Vetter, T.: Reanimating Faces in Images and Video. In: Eurographics, Granada, Spain (2003)
16. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes, the Art of Scientific Computing. Cambridge U. Press, Cambridge, MA (1992)
17. Yacoob, Y., Davis, L.S.: Learned Models for Estimation of Rigid and Articulated Human Motion from Stationary or Moving Camera. *International Journal of Computer Vision* **36** (2000) 5–30
18. Plänkers, R., Fua, P.: Articulated Soft Objects for Multi-View Shape and Motion Capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2003)
19. Shoemake, K.: Animating Rotation with Quaternion Curves. *Computer Graphics, SIGGRAPH Proceedings* **19** (1985) 245–254

Robust Fitting by Adaptive-Scale Residual Consensus

Hanzi Wang and David Suter

Department of Electrical and Computer Systems Engineering
Monash University, Clayton Vic. 3800, Australia
`{hanzi.wang, d.suter}@eng.monash.edu.au`

Abstract. Computer vision tasks often require the robust fit of a model to some data. In a robust fit, two major steps should be taken: i) robustly estimate the parameters of a model, and ii) differentiate inliers from outliers. We propose a new estimator called Adaptive-Scale Residual Consensus (ASRC). ASRC scores a model based on both the residuals of inliers and the corresponding scale estimate determined by those inliers. ASRC is very robust to multiple-structural data containing a high percentage of outliers. Compared with RANSAC, ASRC requires no pre-determined inlier threshold as it can simultaneously estimate the parameters of a model and the scale of inliers belonging to that model. Experiments show that ASRC has better robustness to heavily corrupted data than other robust methods. Our experiments address two important computer vision tasks: range image segmentation and fundamental matrix calculation. However, the range of potential applications is much broader than these.

1 Introduction

Unavoidably, computer vision data is contaminated (e.g., faulty feature extraction, sensor noise, segmentation errors, etc) and it is also likely that the data include multiple structures. Considering any particular structure, outliers to that structure can be classified into gross outliers and pseudo outliers [16], the latter being data belonging to other structures. Computer vision algorithms should be robust to outliers including pseudo outliers [6]. Robust methods have been applied to a wide variety of tasks such as optical flow calculation [1, 22], range image segmentation [24, 15, 11, 10, 21], estimating the fundamental matrix [25, 17, 18], etc.

The breakdown point is the smallest percentage of outliers that can cause the estimator to produce arbitrarily large values ([13], pp.9.). Least Squares (LS) has a breakdown point of 0%. To improve on LS, robust estimators have been adopted from the statistics literature (such as M-estimators [9], LMedS and LTS [13], etc) but they tolerate no more than 50% outliers, limiting their suitability [21]. The computer vision community has also developed techniques to cope with outliers: e.g., the Hough Transform [8], RANSAC [5], RESC [24], MINPRAN [15], MUSE [11], ALKS [10], pbM-estimator [2], MSAC and MLESAC [17]. The Hough Transform determines consensus for a fit from “votes” in a binned parameter space: however one must choose the bin size wisely and, in any case, this technique suffers from high cost when the number of parameters is large. Moreover, unlike the other techniques, it returns a limited precision result (limited by the bin size). RANSAC requires a user-

supplied error tolerance. RESC attempts to estimate the residual probability density function but the method needs the user to tune many parameters and we have found that it overestimates the scale of inliers. MINPRAN assumes that the outliers are randomly distributed within a certain range, making MINPRAN less effective in extracting multiple structures. MUSE requires a lookup table for the scale estimator correction and ALKS is limited in its ability to handle extreme outliers.

In this paper, we propose (section 0) a new robust estimator: Adaptive-Scale Residual Consensus (ASRC), which is based on a robust two-step scale estimator (TSSE) (section 0). We apply ASRC to range image segmentation and fundamental matrix calculation (section 0) demonstrating that ASRC outperforms other methods.

2 A Robust Scale Estimator: TSSE

TSSE [23] is derived from kernel density estimation techniques and the mean shift/mean shift valley method. Kernel estimation is a popular method for probability density estimation [14]. For n data points $\{X_i\}_{i=1,\dots,n}$ in a 1-dimensional residual space, the kernel density estimator with kernel K and bandwidth h is ([14], p.76):

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \quad (1)$$

The Epanechnikov kernel ([14], p.76)

$$K(X) = \begin{cases} \frac{3}{4}(1 - X^2) & \text{if } (1 - X^2) > 0; 0 & \text{otherwise} \end{cases} \quad (2)$$

is optimum in terms of minimum mean integrated square error (MISE), satisfying various conditions ([19], p.95). Using such a kernel, the mean shift vector $M_h(x)$ is:

$$M_h(x) \equiv \frac{1}{n_x} \sum_{X_i \in S_h(x)} (X_i - x) = \frac{1}{n_x} \sum_{X_i \in S_h(x)} X_i - x \quad (3)$$

where $S_h(x)$ is a hypersphere of the radius h , having the volume $h^d c_d$ (c_d is the volume of the unit d -dimensional sphere, e.g., $c_1=2$), centered at x , and containing n_x data points.

Marching in the direction of this vector we perform gradient ascent to the peak. However, for TSSE we also need to find the valleys. Based upon the Gaussian kernel, a saddle-point seeking method was published in [4] but we employ a more simple method [20], based upon the Epanechnikov kernel and, for our purposes, in 1-D residual space. The basic idea is to define the mean shift valley vector as:

$$MV_h(x) = -M_h(x) = x - \frac{1}{n_x} \sum_{X_i \in S_h(x)} X_i \quad (4)$$

In order to avoiding the oscillations, we modify the step size as follows. Let $\{y_i\}_{i=1,2,\dots}$ be the sequence of successive locations of the mean shift valley procedure, then we have, for each $i=1,2,\dots$,

$$y_{i+1} = y_i + \tau \cdot MV_h(y_i) \quad (5)$$

where τ is a correction factor, and $0 < \tau \leq 1$. If the shift step at y_i is too large, it causes y_{i+1} to jump over the local valley and thus oscillate over the valley. This problem can be avoided by adjusting τ so that $MV_h(y_i)^T MV_h(y_{i+1}) > 0$.

A crucial issue in implementing the TSSE is the kernel bandwidth choice [19, 3]. A simple over-smoothed bandwidth selector can be employed [19].

$$\hat{h} = \left[\frac{243R(K)}{35u_2(K)^2 n} \right]^{1/5} S \quad (6)$$

where $R(K) = \int_{-1}^1 K(\zeta)^2 d\zeta$ and $u_2(K) = \int_{-1}^1 \zeta^2 K(\zeta) d\zeta$. S is the sample standard deviation.

The median [13], MAD [12] or robust k [10] scale estimator can be used to yield an initial scale estimate. It is recommended that the bandwidth be set as $c \hat{h}$, ($0 < c < 1$) to avoid over-smoothing ([19], p.62).

We can now describe the TSSE process:

1. Use mean shift, with initial center zero, to find the local peak, and then we use the mean shift valley to find the valley next to the peak: all in ascending ordered absolute residual space.
2. Estimate the scale of the fit by the median scale estimator [13] on the points whose residuals are within the obtained band centered at the local peak.

Based on TSSE, a new robust estimator (ASRC) will be provided in the next section.

3 Robust Adaptive-Scale Residual Consensus Estimator

We assume that when a model is correctly found, two criteria should be satisfied:

- The (weighted) sum of absolute residuals (r_i) of the inliers should be small.
- The scale (S) (standard variance) of the inliers should be small.

Given S , the inliers are those that satisfy:

$$|r_{\vartheta i}/S_{\vartheta}| < T \quad (7)$$

where T is a threshold. If T is 2.5(1.96), then 98%(95%) percent of a Gaussian distribution will be identified as inliers. In our experiments, $T=2.5$ (except for section 0 where $T=1.96$)

$$\hat{\theta} = \arg \max_{\hat{\theta}} \left(\frac{\sum_{i=1}^{n_{\hat{\theta}^{in}}} (1 - |r_{\hat{\theta} i}/(S_{\hat{\theta}} T)|)}{S_{\hat{\theta}}} \right) \quad (8)$$

where $n_{\hat{\theta}^{in}}$ is the number of inliers which satisfies equation (7) for the fitted $\hat{\theta}$.

No priori knowledge about the scale of inliers is necessary as the proposed method yields the estimated parameters of a model and the corresponding scale simultaneously.

The ASRC estimator algorithm is as follows (for fitting models with p parameters):

1. Randomly choose one p -subset from the data points, estimate the model parameters using the p -subset, and calculate the ordered absolute residuals.

2. Choose the bandwidth by equation (6). A robust k scale estimator [10] ($k=0.2$) is used to yield a coarse initial scale S_o .
3. Apply TSSE to the absolute sorted residuals to estimate the scale of inliers S_1 . Because the robust k scale estimator is biased for data with multiple structures, use S_1 in equation (6) to apply TSSE again for the final scale of inliers S_2 .
4. Validate the valley. The probability density at the local peak $\hat{f}(\text{peak})$ and local valley $\hat{f}(\text{valley})$ are obtained by equation (1). Let $\hat{f}(\text{valley})/\hat{f}(\text{peak})=\lambda$ (where $1 > \lambda \geq 0$). Because the inliers are assumed having a Gaussian-like distribution, the valley is not sufficiently deep when λ is too large (say, larger than 0.8). If the valley is sufficiently deep, go to step (5); otherwise go to step (1).
5. Calculate the score, i.e., the objective function of the ASRC estimator.
6. Repeat step (1) to step (5) m times. Finally, output the parameters and the scale S_2 with the highest score.

Let ε be the fraction of outliers, P the probability that at least one of the m p -tuples is “clean”; then one can determine m by ([13], pp.198):

$$m = \frac{\log(1-P)}{\log[1 - (1-\varepsilon)^p]} \quad (9)$$

In [23], we propose a robust Adaptive Scale Sample Consensus (ASSC) estimator:

$$\hat{\theta} = \arg \max_{\hat{\theta}} (n_{\hat{\theta}_{in}} / S_{\hat{\theta}}) \quad (10)$$

From equation (8) and (10), we can see that the difference between ASRC and our recently proposed ASCS [23] is: in ASCS, all inliers are treated as the same, i.e., each inlier contributes 1 to the object function of ASCS. However, in ASRC, the sizes of the residuals of inliers are influential.

4 Experiments

4.1 Synthetic Examples on Line Fitting and Plane Fitting

The proposed method is compared with LMedS, RESC, ALKS, and our recently proposed method: ASCS. We generated four examples: roof, ‘F’-figure, one-step, and three-step linear signals (the signals are in the magenta color), each with a total of 500 data points, corrupted by Gaussian noise with zero mean and standard variance σ . Among the 500 data points, α data points were randomly distributed in the range of (0, 100). The i 'th structure has n_i data points: **(a)** Roof: $x:(0-50)$, $y=2x$, $n_1=65$; $x:(50-100)$, $y=200-2x$, $n_2=50$; $\alpha=385$; $\sigma=1$. **(b)** F-figure: $x:(25-75)$, $y=85$, $n_1=40$; $x:(25-75)$, $y=70$, $n_2=35$; $x=25$, $y:(30-85)$, $n_3=35$; $\alpha=390$; $\sigma=1.2$. **(c)** Step: $x:(0-50)$, $y=75$, $n_1=45$; $x:(50-100)$, $y=60$, $n_2=45$; $\alpha=410$; $\sigma=1$. **(d)** Three-step: $x:(0-25)$, $y=20$, $n_1=45$; $x:(25-50)$, $y=40$, $n_2=30$; $x:(50-75)$, $y=60$, $n_3=30$; $x:(75-100)$, $y=80$, $n_4=30$; $\alpha=365$; $\sigma=1$.

From Fig. 1 we can see that ASRC correctly fits all four signals. LMedS (50% breakdown point) failed to fit all four. Although ALKS is sometimes more robust, it also failed. RESC and ASCS succeeded in the roof signal (87% outliers), however, they both failed in the other three cases. It should be emphasized that both the

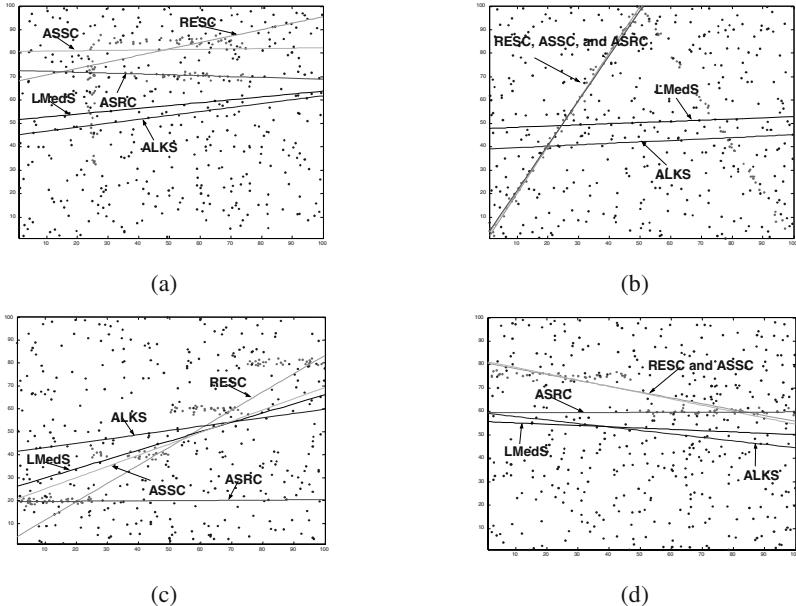


Fig. 1. Comparing the performance of five methods: (a) fitting a roof with a total of 87% outliers; (b) fitting F-figure with a total of 92% outliers; (c) fitting a step with a total of 91% outliers; (d) fitting three-step with a total of 91% outliers.

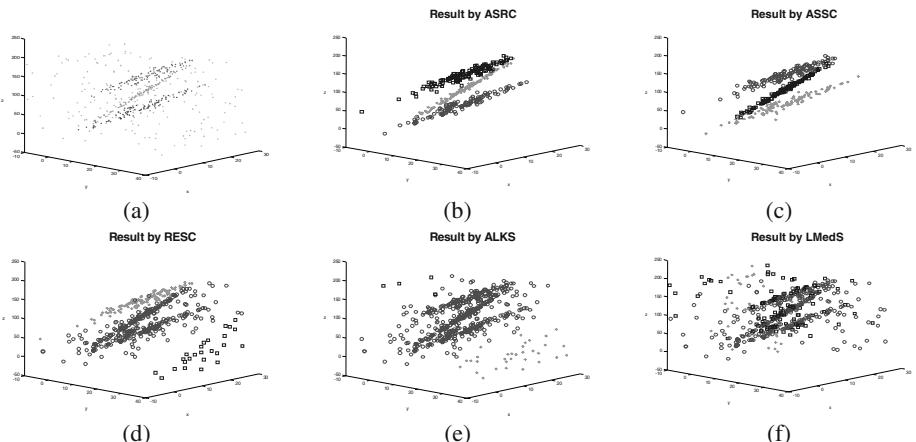


Fig. 2. (a) the 3D data with 80% outliers; the extracted results by (b) ASRC; (c) ASSC; (d) RESC; (e) ALKS; and (f) LMedS.

bandwidth choice and the scale estimation in ASRC are data-driven: an improvement over RANSAC where the user sets a priori scale-related error bound.

Next, two 3D signals were used: 500 data points and three planar structures with each plane containing n points corrupted by Gaussian noise with standard variance σ (=3.0); 500- $3n$ points are randomly distributed. In the first example, n =100; in the

second $n = 65$. We repeat: (1) estimate the parameters and scale of a plane (2) extract the inliers and remove them from the data set - until all planes are extracted. The red circles denote the first plane extracted; green stars the second; and blue squares the third (Fig. 2 and Fig. 3).

From Fig. 2 (d) and (e), we can see that RESC and ALKS, which claim to be robust to data with more than 50% outliers, failed to extract all the three planes. This is because the estimated scales (by RESC and ALKS) for the first plane were wrong, which caused these two methods to fail to fit the second and third planes. Because the LMedS (in Fig. 2 (d)) has only a 50% breakdown point, it completely failed to fit data with such high contamination — 80% outliers. The proposed method and ASSC yielded the best results (Fig. 2 (b) and (c)). Similarly, in the second 3D experiment (Fig. 3), RESC, ALKS and LMedS completely broke down. ASSC, although it correctly fitted the first plane, wrongly fitted the second and the third planes. Only the proposed method correctly fitted and extracted all three planes (Fig. 3 (b)).

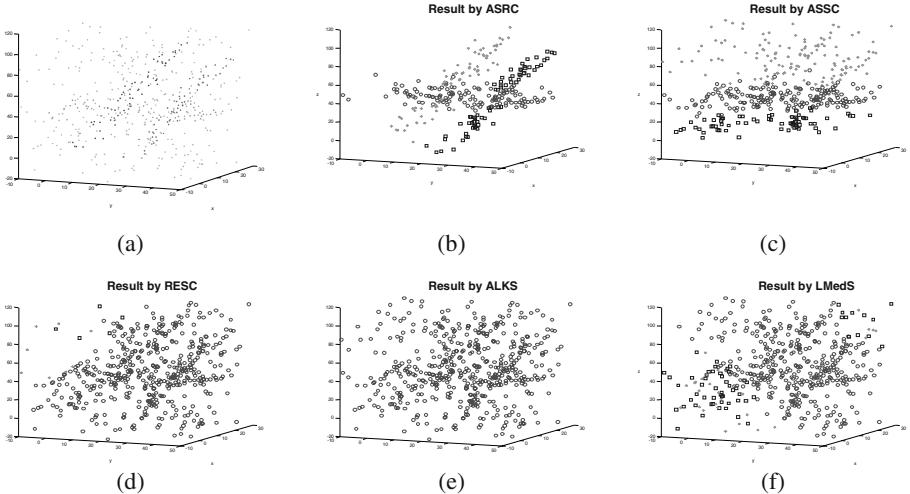


Fig. 3. (a) the 3D data with 87% outliers; the extracted results by (b) ASRC; (c) ASSC; (d) RESC; (e) ALKS; and (f) LMedS.

4.2 Range Image Segmentation

Many robust estimators have been employed to segment range images ([24, 11, 10, 21], etc.). Here, we use the ABW range images (obtained from <http://marathon.csee.usf.edu/seg-comp/SegComp.html>.) The images have 512x512 pixels and contain planar structures. We employ a hierarchical approach with four levels [21]. The bottom level of the hierarchy contains 64x64 pixels that are obtained by using regular sampling on the original image. The top level of the hierarchy is the original image. We begin with bottom level. In each level of the hierarchy, we:

- (1) Apply the ASRC estimator to obtain the parameters of plane and the scale of inliers. If the number of inliers is less than a threshold, go to step (6).
- (2) Use the normals to the planes to validate the inliers obtained in step (1). When the angle between the normal of the data point that has been classified as an

inlier, and the normal of the estimated plane, is less than a threshold value, the data point is accepted. Otherwise, the data point is rejected and will be left for further processing. If the number of the validated inliers is small, go to step (6).

- (3) Fill in the holes, which may appear due to sensor noise, inside the maximum connected component (CC) from the validated inliers.
- (4) In the top hierarchy, assign a label to the points corresponding to the CC from step (3) and remove these points from the data set.
- (5) If a point is unlabelled and it is not a jump edge point, the point is a "left-over" point. After collecting all these, use the CC algorithm to get the maximum CC. If the number data points of the maximum CC of "left-over" points is smaller than a threshold, go to step (6); otherwise, sample the maximum CC obtained in this step, then go to step (1).
- (6) Terminate the processing in the current level of the hierarchy and go to the higher-level hierarchy until the top of the hierarchy.

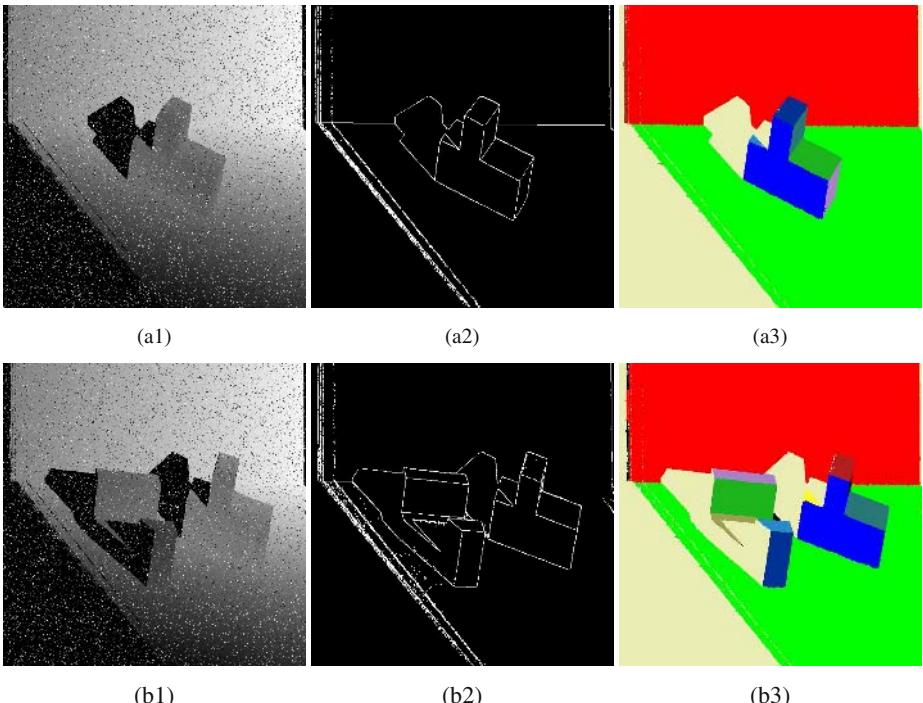


Fig. 4. Segmentation of ABW range images from the USF database. (a1, b1) Range image with 26214 random noise points; (a2, b2) The ground truth results for the corresponding range images without adding random noise; (a3, b3) Segmentation result by ASRC.

The proposed range image segmentation method is very robust to noise. We added 26214 random noise points to the range images (in Fig. 4) taken from the USF ABW range image database (“test 11” and “test 3”). No separate noise filtering is performed. All of the main surfaces were recovered by our method.

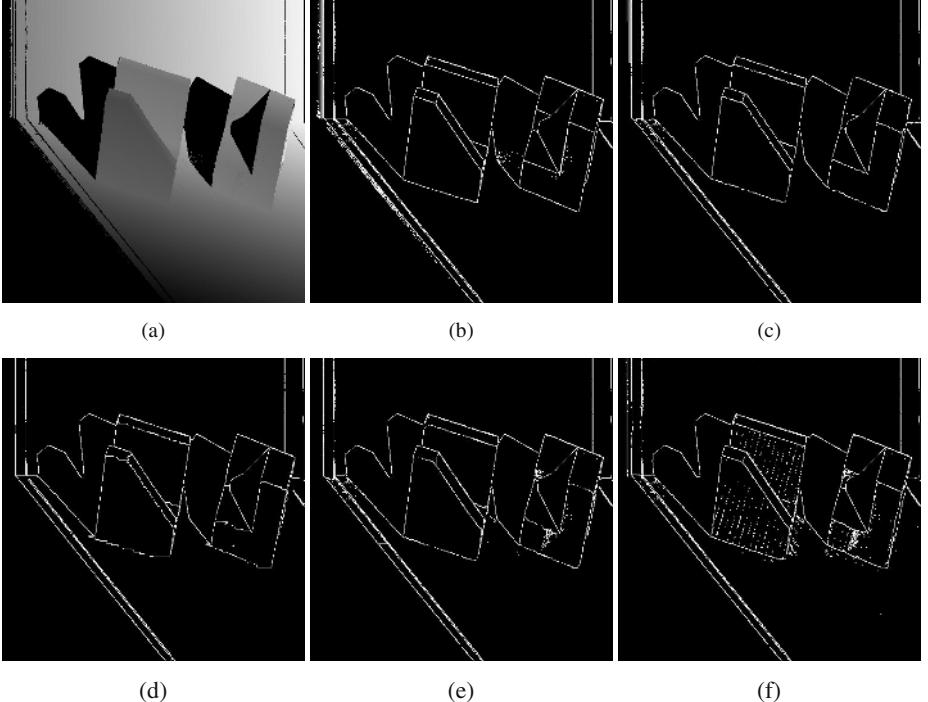


Fig. 5. Comparison of the segmentation results for ABW range image (train 7). (a) Range image; (b) The result of ground truth; (c) The result by the ASRC; (d) The result by the UB; (e) The result by the WSU; (f) The result by the USF.

We also compared our results with those of three state-of-the-art approaches of USF, WSU, and UB [7]. Fig. 5 (c-f), showing the results obtained by the four methods should be compared with the results of the ground truth (Fig. 5 (b)).

From Fig. 5, we can see that the proposed method achieved the best results: all surfaces are recovered and the segmented surfaces are relatively “clean”. In comparison, some boundaries on the junction of the segmented patch by the UB were seriously distorted. The WSU and USF results contained many noisy points and WSU over segmented one surface. The proposed method takes about 1-2 minutes (on an AMD800MHz personal computer in C interfaced with MATLAB language).

4.3 Fundamental Matrix Estimation

Several robust estimators, such as M-estimators, LMedS, RANSAC, MSAC and MLESAC, have been applied in estimating the fundamental matrix [17]. However, M-estimators and the LMedS have a low breakdown point, RANSAC and MSAC need a priori knowledge about the scale of inliers. MLESAC performs similar to MSAC.

The proposed ASRC can tolerate more than 50% outliers; and no priori scale information about inliers is required.

Let $\{x_i\}$ and $\{x'_i\}$ (for $i=1,\dots,n$) to be a set of homogeneous image points viewed in image 1 and image 2. We have the following constraints for the fundamental matrix F :

$$x_i^T F x_i = 0 \text{ and } \det[F] = 0 \quad (11)$$

We employ the 7 points algorithm [17] to solve for candidate fits using Simpson distance - for the i 'th correspondence r_i using Simpson distance is:

$$r_i = \frac{k_i}{(k_x^2 + k_y^2 + k_{x'}^2 + k_{y'}^2)^{1/2}} \quad (12)$$

where $k_i = f_1 x_i x_i + f_2 x_i y_i + f_3 x_i \zeta + f_4 y_i x_i + f_5 y_i y_i + f_6 y_i \zeta + f_7 x_i \zeta + f_8 y_i \zeta + f_9 \zeta^2$.

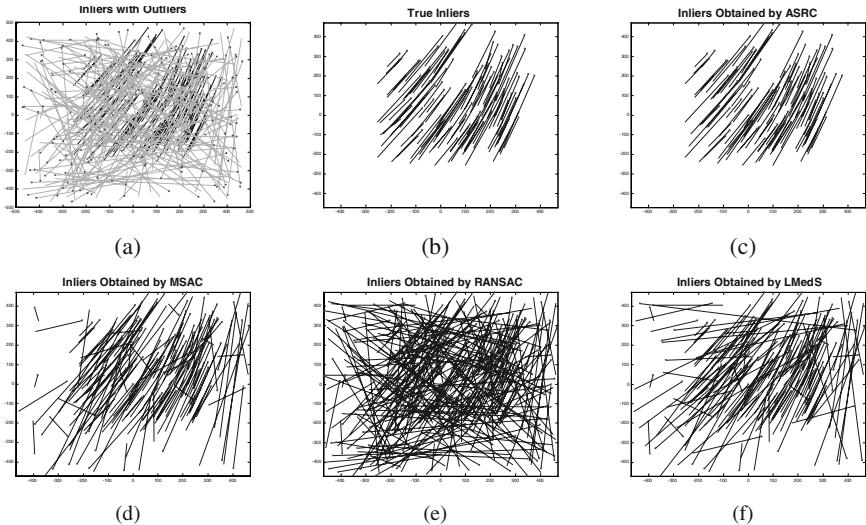


Fig. 6. An experimental comparison of estimating fundamental matrix for data with 60% outliers. (a) The distributions of inliers and outliers; (b) The distribution of true inliers; The inliers obtained by (c) ASRC; (d) MSAC; (e) RANSAC; and (f) LMedS.

Table 1. An experimental comparison for data with 60% outliers.

	% of inliers correctly classified	% of outliers correctly classified	Standard variance of inliers
Ground Truth	100.00	100.00	0.9025
ASRC	95.83	100.00	0.8733
MSAC	100.00	65.56	41.5841
RANSAC	100.00	0.56	206.4936
LMedS	100.00	60.00	81.1679

We generated 300 matches including 120 point pairs of inliers with unit Gaussian variance (matches in blue color in Fig. 6(a)) and 160 point pairs of random outliers (matches in cyan color in Fig. 6(a)). Thus the outliers occupy 60% of the whole data.

The scale information about inliers is usually not available, thus, the median scale estimator, as recommended in [17], is used for RANSAC and MSAC to yield an initial scale estimate. The number of random samples is set to 10000. From Fig. 6 and Table 1, we can see that our method yields the best result.

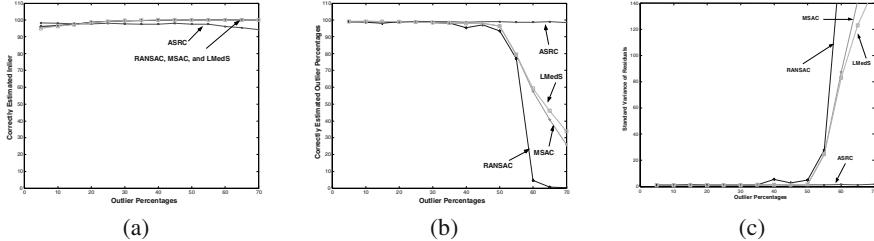


Fig. 7. A comparison of correctly identified percentage of inliers (a), outliers (b), and the comparison of standard variance of residuals of inliers (c).

Table 2. Experimental results on two frames of the Corridor sequence.

	Number of inliers	Mean error of inliers	Standard variance of inliers
ASRC	269	-0.0233	0.3676
MSAC	567	-0.9132	7.5134
RANSAC	571	-1.2034	8.0816
LMedS	571	-1.1226	8.3915

Next, we investigate the behavior for data involving different percentages of outliers (PO). We generated the data (in total 300 correspondences) similar to that in Fig. 6. The percentage of outliers varies from 5% to 70% in increments of 5%. The experiments were repeated 100 times for each percentage of outliers. If a method is robust enough, it should resist the influence of outliers and the correctly identified percentages of inliers should be around 95% (T is set 1.96 in equation (7)) and the standard variance of inliers should be near to 1.0 despite of the percentages of outliers.

We set the number of random samples, m , to be: $m = 1000$ when $PO \leq 40$; 10000 when $40 < PO \leq 60$; and 30000 when $PO > 60$ to ensure a high probability of success.

From Fig. 7, we can see that MSAC, RANSAC, and LMedS all break down when data involve more than 50% outliers. The standard variance of inliers by ASRC is the smallest when $PO > 50\%$. Note: ASRC succeeds to find the inliers and outliers even when outliers occupied 70% of the whole data.

Next, two frames of the Corridor sequence (bt.000 and bt.004), were obtained from <http://www.robots.ox.ac.uk/~vgg/data/> (Fig. 8(a) and (b)). Fig. 8(c) shows the matches involving 800 point pairs in total. The inliers (269 correspondences) obtained by the proposed method are shown in Fig. 8(d). The epipolar lines and epipole using the estimated fundamental matrix by ASRC are shown in Fig. 8(e) and (f). In Fig. 8(e) and (f), we draw 30 epipolar lines. We can see that most of the point pairs correspond to the same feature in the two images except for one case: the 30th point pair, which is pointed out by the two arrows. The reason is that the residual of the point pair corresponding to the estimated fundamental matrix is small: the epipolar constraint is

a weak constraint and ANY method enforcing ONLY the epipolar constraint scores this match highly. Because the camera matrices of the two frames are available, we can obtain the ground truth fundamental matrix and thus evaluate the errors (Table 2). We can see that ASRC performs the best.

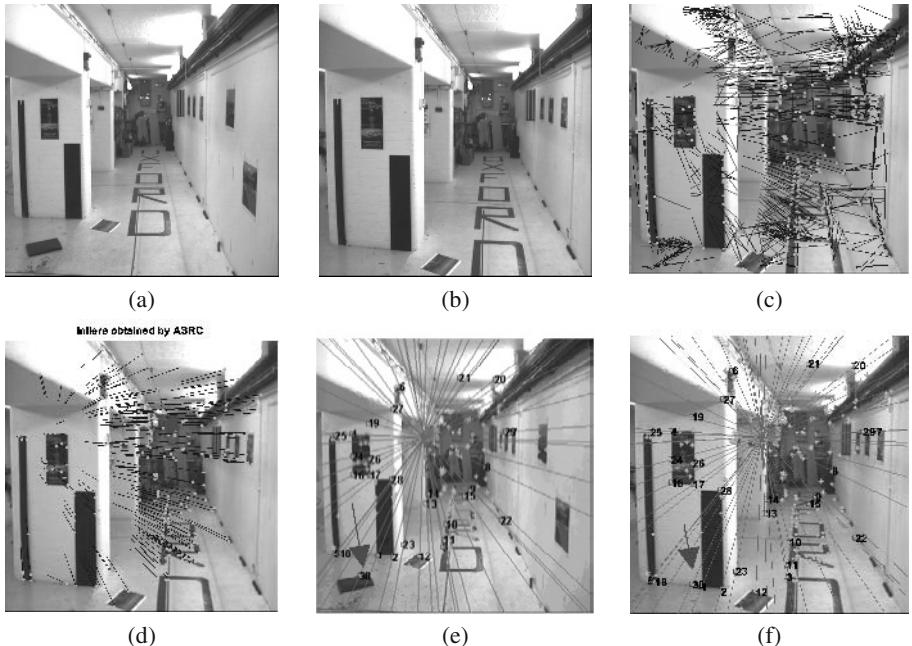


Fig. 8. (a)(b) image pair (c) matches (d) inliers by ASRC; (e)(f) epipolar geometry.

5 Conclusion

The proposed ASRC method exploits both the residuals of inliers and the corresponding scale estimate using those inliers, in determining the merit of model fit. This estimator is very robust to multiple-structural data and can tolerate more than 80% outliers. The ASRC estimator is compared to those of several popular and recently proposed robust estimators: LMedS, RANSAC, MSAC, RESC, ALKS, and ASSC, showing that the ASRC estimator achieves better results (Readers may download the paper from <http://www-personal.monash.edu.au/~hanzi>, containing the corresponding colors figure/images, to understand the results better). Recently, a “pbM-estimator”[2], also using kernel density estimation was announced. However, this employs projection pursuit and orthogonal regression. In contrast, we consider the density distribution of the mode in the residual space.

Acknowledgements. The authors thank Andrew Zisserman, Hongdong Li, Kristy Sim, and Haifeng Chen for their valuable helps/suggestions. This work is supported by the Australia Research Council (ARC), under the grant A10017082.

References

1. Bab-Hadiashar, A., Suter, D.: *Robust Optic Flow Computation*. International Journal of Computer Vision. **29**(1) (1998) 59-77
2. Chen, H., Meer, P.: *Robust Regression with Projection Based M-estimators*. in ICCV. Nice, France (2003) 878-885
3. Comaniciu, D., Meer, P.: *Mean Shift Analysis and Applications*. in ICCV, Kerkyra, Greece. (1999) 1197-1203
4. Comaniciu, D., Ramesh, V., Bue, A.D.: *Multivariate Saddle Point Detection for Statistical Clustering*. in ECCV. Copenhagen, Danmark (2002) 561-576
5. Fischler, M.A., Rolles, R.C.: *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Commun. ACM. **24**(6) (1981) 381-395
6. Haralick, R.M.: *Computer vision theory: The lack thereof*. CVGIP. **36** (1986) 372-386
7. Hoover, A., Jean-Baptiste, G., Jiang., X.: *An Experimental Comparison of Range Image Segmentation Algorithms*. IEEE Trans. PAMI. **18**(7) (1996) 673-689
8. Hough, P.V.C.: Methods and means for recognising complex patterns. U.S. Patent 3 069 654. (1962)
9. Huber, P.J.: *Robust Statistics*. New York, Wiley. (1981)
10. Lee, K.-M., Meer, P., Park, R.-H.: *Robust Adaptive Segmentation of Range Images*. IEEE Trans. PAMI. **20**(2) (1998) 200-205
11. Miller, J.V., Stewart, C.V.: *MUSE: Robust Surface Fitting Using Unbiased Scale Estimates*. in CVPR, San Francisco (1996) 300-306
12. Rousseeuw, P.J., Croux, C.: *Alternatives to the Median Absolute Deviation*. Journal of the American Statistical Association. **88**(424) (1993) 1273-1283
13. Rousseeuw, P.J., Leroy, A.: *Robust Regression and outlier detection*. John Wiley & Sons, New York. (1987)
14. Silverman, B.W.: *Density Estimation for Statistics and Data Analysis* London: Chapman and Hall. (1986).
15. Stewart, C.V.: *MINPRAN: A New Robust Estimator for Computer Vision*. IEEE Trans. PAMI. **17**(10) (1995) 925-938
16. Stewart, C.V.: *Bias in Robust Estimation Caused by Discontinuities and Multiple Structures*. IEEE Trans. PAMI. **19**(8) (1997) 818-833
17. Torr, P., D. Murray: *The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix*. International Journal of Computer Vision. **24** (1997) 271-300
18. Torr, P., Zisserman, A.: *MLESAC: A New Robust Estimator With Application to Estimating Image Geometry*. Computer Vision and Image Understanding. **78**(1) (2000) 138-156
19. Wand, M.P., Jones, M.: *Kernel Smoothing*. Chapman & Hall. (1995)
20. Wang, H., Suter, D.: *False-Peaks-Avoiding Mean Shift Method for Unsupervised Peak-Valley Sliding Image Segmentation*. in *Digital Image Computing Techniques and Applications*. Sydney, Australia (2003) 581-590
21. Wang, H., Suter, D.: *MDPE: A Very Robust Estimator for Model Fitting and Range Image Segmentation*. International Journal of Computer Vision. (2003) to appear
22. Wang, H., Suter, D.: *Variable bandwidth QMDPE and its application in robust optic flow estimation*. in ICCV. Nice, France (2003) 178-183
23. Wang, H., Suter, D.: *Robust Adaptive-Scale Parametric Model Estimation for Computer Vision*. submitted to IEEE Trans. PAMI. (2003)
24. Yu, X., Bui, T.D., Krzyzak, A.: *Robust Estimation for Range Image Segmentation and Reconstruction*. IEEE Trans PAMI. **16**(5) (1994) 530-538
25. Zhang, Z., et al.: *A Robust Technique for Matching Two Uncalibrated Image Through the Recovery of the Unknown Epipolar Geometry*. Artificial Intelligence. **78** (1995) 87-11

Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures

Tal Nir and Alfred M. Bruckstein

Computer Science Department, Technion, Haifa 32000, ISRAEL
`{taln,freddy}@cs.technion.ac.il`

Abstract. The problem of Simultaneous Localization And Mapping (SLAM) originally arose from the robotics community and is closely related to the problems of camera motion estimation and structure recovery in computer vision. Recent work in the vision community addressed the SLAM problem using either active stereo or a single passive camera. The precision of camera based SLAM was tested in indoor static environments. However the extended Kalman filters (EKF) as used in these tests are highly sensitive to outliers. For example, even a single mismatch of some feature point could lead to catastrophic collapse in both motion and structure estimates. In this paper we employ a robust-statistics-based condensation approach to the camera motion estimation problem. The condensation framework maintains multiple motion hypotheses when ambiguities exist. Employing robust distance functions in the condensation measurement stage enables the algorithm to discard a considerable fraction of outliers in the data. The experimental results demonstrate the accuracy and robustness of the proposed method.

1 Introduction

While the vision community struggled with the difficult problem of estimating motion and structure from a single camera generally moving in 3D space (see [5]), the robotics community independently addressed a similar estimation problem known as Simultaneous Localization and Mapping (SLAM) using odometry, laser range finders, sonars and other types of sensors together with further assumptions such as planar robot motion. Recently, the vision community has adopted the SLAM name and some of the methodologies and strategies from the robotics community. Vision based SLAM has been proposed in conjunction with an active stereo head and odometry sensing in [7], where the stereo head actively searched for old and new features with the aim of improving the SLAM accuracy. In [6] the more difficult issue of localization and mapping based on data from a single passive camera is treated. The camera is assumed to be calibrated and some features with known 3D locations are assumed present and these features impose a metric scale on the scene, enable the proper use of a motion model, increase the estimation accuracy and avoid drift. These works on vision based SLAM employ an Extended Kalman Filter (EKF) approach where camera motion parameters are packed together with 3D feature locations to form a large and tightly coupled estimation problem. The main disadvantage of this approach is that even a single outlier in measurement data can lead to a collapse of the whole estimation problem. Although there are means for excluding problematic

feature points in tracking algorithms, it is impossible to completely avoid outliers in uncontrolled environments. These outliers may result from mismatches of some feature points which are highly likely to occur in cluttered environments, at depth discontinuities or when repetitive textures are present in the scene. Outliers may exist even if the matching algorithm performs perfectly when some objects in the scene are moving. In this case multiple-hypothesis estimation as naturally provided by particle filters is appropriate. The estimation of the stationary scene structure together with the camera ego-motion is the desired output under the assumption that most of the camera's field of view looks at a static scene. The use of particle filters in SLAM is not new. Algorithms for FastSLAM [19] employed a particle filter for the motion estimation, but their motivation was mainly computational speed and robust estimation methodology was neither incorporated nor tested. In [18] a version of FastSLAM addressing the problem of data association between landmarks and measurements is presented. However, the solution to the data association problem provided there does not offer a solution to the problem of outliers since all landmarks are assumed stationary and every measurement is assumed to correctly belong to one of the real physical landmarks. Other works like e.g. [6] employed condensation only in initialization of distances of new feature points before their insertion into the EKF. However the robustness issue is not solved in this approach since the motion and mapping are still provided by the EKF. In [23] the pose of the robot was estimated by a condensation approach. However, here too the algorithm lacked robust statistics measures to effectively reject outliers in the data. Furthermore the measurements in this work were assumed to be provided by laser range finders and odometric sensors. In this work we propose a new and robust solution to the basic problem of camera motion estimation from known 3D feature locations, which has practical importance of its own. The full SLAM problem is then addressed in the context of supplementing this basic robust camera motion estimation approach for simultaneously providing additional 3D scene information. The paper is organized as follows: Section 2 formulates the basic motion estimation problem. Section 3 presents the proposed framework for robust motion from structure. Section 4 discusses methods for incorporating the proposed framework for the solution of SLAM. Section 5 presents results on both synthetic data and real sequences and compares the performance to that of EKF based methods.

2 Problem Formulation

Throughout this work it is assumed that the camera is calibrated. This assumption is commonly made in previous works on vision based SLAM. A 3D point indexed by i in the camera axes coordinates, $(X_i(t) \ Y_i(t) \ Z_i(t))^T$ projects to the image point $(x_i(t) \ y_i(t))^T$ at frame time t via some general projection function Π as follows:

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \Pi \begin{pmatrix} X_i(t) \\ Y_i(t) \\ Z_i(t) \end{pmatrix} \quad (1)$$

The camera motion between two consecutive frames is represented by a rotation matrix $R(t)$ and a translation vector $V(t)$. Hence for a static point in the scene:

$$\begin{pmatrix} X_i(t) \\ Y_i(t) \\ Z_i(t) \end{pmatrix} = R(t) \begin{pmatrix} X_i(t-1) \\ Y_i(t-1) \\ Z_i(t-1) \end{pmatrix} + V(t) \quad (2)$$

The rotation is represented using the exponential canonical form $R(t) = e^{\hat{\omega}(t)}$ where $\omega(t)$ represents the angular velocity between frames $t-1$ and t , and the exponent denotes the matrix exponential. The hat notation for some 3D vector q is defined by:

$$q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}; \quad \hat{q} = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix}$$

The matrix exponential of such skew-symmetric matrices may be computed using the Rodrigues' formula:

$$e^{\hat{\omega}} = I + \frac{\hat{\omega}}{\|\omega\|} \sin(\|\omega\|) + \frac{\hat{\omega}^2}{\|\omega\|^2} (1 - \cos(\|\omega\|))$$

Let us denote by $\Omega(t)$ and $T(t)$ the overall rotation and translation from some fixed world coordinate system to the camera axes:

$$\begin{pmatrix} X_i(t) \\ Y_i(t) \\ Z_i(t) \end{pmatrix} = e^{\hat{\Omega}(t)} \begin{pmatrix} X_i^{World} \\ Y_i^{World} \\ Z_i^{World} \end{pmatrix} + T(t) \quad (3)$$

Equation (3) describes the pose of the world relative to the camera. The camera pose relative to the world is given by: $\Omega_{(t)}^{Camera} = -\Omega(t)$; $T_{(t)}^{Camera} = -e^{-\hat{\Omega}(t)}T(t)$

Using equations (2), (3) and (3) written one sample backward:

$$\begin{aligned} \Omega(t) &= \log_{SO(3)} \left(e^{\hat{\omega}(t)} e^{\hat{\Omega}(t-1)} \right) \\ T(t) &= e^{\hat{\omega}(t)} T(t-1) + V(t) \end{aligned} \quad (4)$$

Where, $q = \log_{SO(3)}(A)$ denotes the inverse of the matrix exponential of the skew symmetric matrix A such that $A = e^{\hat{q}}$ (i.e. inverting Rodrigues' formula). Let us define the robust motion from structure estimation problem: given matches of 2D image feature points to known 3D locations, estimate the camera motion in a robust framework accounting for the possible presence of outliers in measurement data.

2.1 Dynamical Motion Model

One can address the camera motion estimation problem with no assumptions on the dynamical behavior of the camera (motion model), thus using only the available geometric information in order to constrain the camera motion. This is equivalent to assuming independent and arbitrary viewpoints at every frame. In most practical applications though, physical constraints result in high correlation of pose between adjacent frames. For example, a camera mounted on a robot traveling in a room produces smooth motion trajectories unless the robot hits some obstacle or collapses. The use of a proper motion model accounts for uncertainties, improves the estimation accuracy, attenuates the influence of measurement noise and helps overcome ambiguities (which may occur if at some time instances, the measurements are not sufficient to uniquely constrain camera pose, see [5] and [6]). Throughout this work, the motion model assumes constant velocity with acceleration disturbances, as follows:

$$\begin{aligned}\omega(t) &= \omega(t-1) + \dot{\omega}(t) \\ V(t) &= V(t-1) + \dot{V}(t)\end{aligned}\tag{5}$$

If no forces act on the camera the angular and translation velocities are constant. Accelerations result from forces and moments which are applied on the camera, and these being unknown are treated as disturbances (recall that the vectors $\omega(t), V(t)$ are velocity terms and the time is the image frame index).

Acceleration disturbances are modeled here probabilistically by independent white Gaussian noises:

$$\begin{aligned}\dot{\omega}(t) &\sim N(0, \sigma_{\dot{\omega}}) \\ \dot{V}(t) &\sim N(0, \sigma_{\dot{V}})\end{aligned}\tag{6}$$

where $\sigma_{\dot{\omega}}$ and $\sigma_{\dot{V}}$ denote expected standard deviations of the angular and linear acceleration disturbances.

3 Robust Motion from Structure by Condensation

In this section we present the proposed condensation based algorithm designed for robust camera 3D motion estimation. A detailed description of condensation in general and its application to contour tracking can be found in [12] and [13]. The state vector of the estimator at time t , denoted by s_t , includes all the motion parameters:

$$s_t = (\Omega(t) \quad T(t) \quad \omega(t) \quad V(t))^T$$

The state vector is of length 12. The state dynamics are generally specified in the condensation framework by the probability distribution function $p(s_t | s_{t-1})$. Our motion model is described by equations (4),(5),(6). All measurements at time t are denoted compactly as $z(t)$. The camera pose is defined for each state s_t separately, with the corresponding expected projections being tested on all the visible points in the current frame:

$$\begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix} = \Pi \left(e^{\hat{\Omega}(t)} \begin{pmatrix} X_i^{World} \\ Y_i^{World} \\ Z_i^{World} \end{pmatrix} + T(t) \right)$$

The influence of the measurements is quantified by $p(z(t) | s_t)$. This is the conditional Probability Distribution Function (PDF) of measuring the identified features $z(t)$ when the true parameters of motion correspond to the state s_t . The conditional PDF is calculated as a function of the geometric error, which is the distance denoted by d_i between the projected 3D feature point location on the image plane and the measured image point. If the image measurement errors are statistically independent random variables with zero mean Gaussian PDF, then up to a normalizing constant:

$$p(z | s) = \exp \left(-\frac{\sum_{i=1}^{N_{points}} d_i^2}{2\sigma^2 N_{points}} \right)$$

Where N_{points} is the number of visible feature points and σ is the standard deviation of the measurement error (about 1 pixel). Since outliers have large d_i values even for the correct motion, the quadratic distance function may be replaced by a robust distance function $\rho(d_i^2)$ see e.g. [20]:

$$p(z | s) = \exp \left(-\frac{\sum_{i=1}^{N_{points}} \rho(d_i^2)}{2\sigma^2 N_{points}} \right) \quad (7)$$

$$\rho(d^2) = \frac{d^2}{1 + d^2/L^2} \quad (8)$$

If some feature point is behind the camera (this occurs when its 3D coordinates expressed in camera axes have a negative Z value), clearly this feature should not have been visible and hence its contribution to the sum is set to the value:

$$\lim_{d_i \rightarrow \infty} \rho(d_i^2) = L^2$$

The influence of every feature point on the PDF is now limited by the parameter L. The choice of L reflects a threshold value between inliers and outliers. In order to understand why robustness is achieved using such distance functions, let us consider the simpler robust distance function, the truncated quadratic:

$$\rho(d^2) = \begin{cases} d^2 & d^2 < A^2 \\ A^2 & Otherwise \end{cases}$$

where, A is the threshold value between inliers and outliers. Using this ρ function in equation (7) yields:

$$p(z | s) = \exp \left(-\frac{\sum_{i \in \text{Inlier points}} d_i^2 + \sum_{i \in \text{Outlier points}} A^2}{2\sigma^2 N_{\text{points}}} \right) = \exp \left(-\frac{\sum_{i \in \text{Inlier points}} d_i^2 + A^2 \cdot (\# \text{Outliers})}{2\sigma^2 N_{\text{points}}} \right)$$

Maximizing this PDF (a maximum likelihood estimate) is equivalent to minimizing the sum of the two terms, the first is the sum of the quadratic distances at the inlier points and the second term is proportional to the number of outliers. The robust distance function of equation (8) is similar to the truncated quadratic, with a smoother transition between the inliers and outliers (see [2] and [3] for an analysis of ρ functions used in robust statistics and their use for image reconstruction and for the calculation of piecewise-smooth optical flow fields). Let us summarize the proposed algorithm for robust 3D motion estimation from known structure:

Initialization- Sample N states $s_0^{(n)}, n=1\dots N$ from the prior PDF of $\omega(0), V(0)$ and $\Omega(0); T(0)$. Initialize $\pi_0^{(n)}$ with the PDF corresponding to each state.

At every time step $t=1,2, \dots$:

- Sample N states $\tilde{s}_{t-1}^{(n)}$ copied from the states $s_{t-1}^{(n)}$ with probabilities $\pi_{t-1}^{(n)}$.
- Propagate the sampled states using equations (6),(5),(4) to obtain $s_t^{(n)}$.
- Incorporate the measurements to obtain $\pi_t^{(n)} = p(z_t | s_t^{(n)})$ using equations (7),(8). Then normalize by the appropriate factor so that: $\sum_{n=1}^N \pi_t^{(n)} = 1$
- Extract the dominant camera motion from the state $s_t^{(n)}$ corresponding to the maximum of $\pi_t^{(n)}$: $\Omega_{(t)}^{\text{Camera}} = -\Omega_{(t)}^{(n)}; T_{(t)}^{\text{Camera}} = -e^{-\hat{\Omega}_{(t)}^{(n)}} T_{(t)}^{(n)}$

Code written in C++ implementing the algorithm of this section can be found in [25]. It can run in real time on a Pentium 4, 2.5GHz processor, with 30Hz sampling rate, 1000 particles and up to 200 instantaneously visible feature points.

4 Application to SLAM

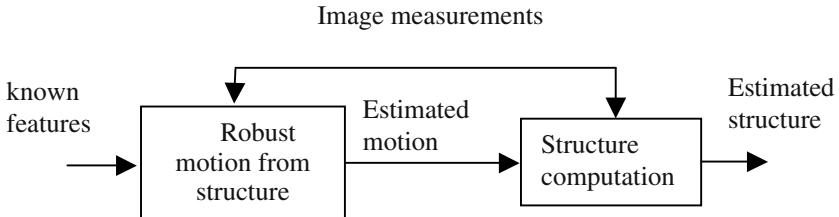
This section describes various possible solutions to the robust SLAM problem.

4.1 SLAM in a Full Condensation Framework

The most comprehensive solution to robust SLAM is the packing of all the estimated parameters into one large state and solve using a robust condensation framework. The state is composed of the motion parameters and each feature contributes three additional parameters for its 3D location. As stated in [7], this solution is very expensive computationally due to the large number of particles required to properly sample from the resulting high dimensional space.

4.2 SLAM in a Partially Decoupled Scheme

The research on vision based SLAM tends to incorporate features with known 3D locations in the scene. The simplest method for incorporating the proposed robust motion from structure algorithm into SLAM is in a partially decoupled block scheme in which the features with known 3D locations are the input to the robust motion estimation block of section 3. Structure of other features in the scene can be recovered using the estimated motion and the image measurements. Assuming known 3D motion, the structure of each feature can be estimated using an EKF independently for each feature (similar to FastSlam in [19]). If enough features with known structure are available in the camera field of view at all times (few can be enough as shown in the experiments section), then this method can work properly. It may be practical for robots moving in rooms and buildings to locate known and uniquely identifiable features (fiducials) at known locations. When the motion estimation is robust, the independence of the estimators for the structure of the different features guarantees the robustness of the structure recovery as well.



4.3 SLAM with Robust Motion Estimation and Triangulation

In this section we propose a solution to the robust SLAM problem in a condensation framework with a state containing motion parameters only. In the measurement phase, features with known locations have their 3D structure projected on the image plane, features with unknown structure have their 3D structure reconstructed using triangulation (see [9] chapter 11) and the geometric error is measured by projecting this structure back on the image plane. The information regarding the camera pose in the current and previous frames is embedded in each state hypothesis of the condensation algorithm which together with the corresponding image measurements form the required information for the triangulation process. Triangulation can be performed from three views, where the third view is the first appearance of the feature.

5 Experimental Results

5.1 Synthetic Tests

It has been experimentally found using synthetic tests that robustness with the proposed method is maintained with up to about 33% of outliers. The proposed

algorithm is compared with the results of the EKF approach in [5] which is run with the code supplied in [15]. The robust motion estimation used triangulation in three frames as described in section 4.3. The 3D structure was unknown to both algorithms. The outlier points are randomly chosen and remain fixed throughout the sequence, these points are given random image coordinates uniformly distributed in the image range (see examples in [25]). The rotation errors are compactly characterized by:

$$\left\| I - \left(e^{\hat{\Omega}_{True}} \right)^T e^{\hat{\Omega}_{Estimated}} \right\|_{Frobenius}^2$$

The estimation results are shown in Fig. 1. With 33% of outliers, the EKF errors are unacceptable while the proposed method maintains reasonable accuracy.

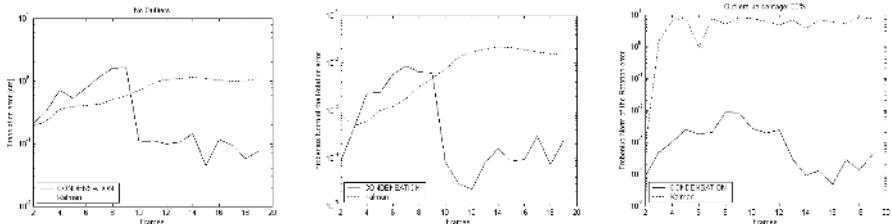


Fig. 1. The translation (left) and rotation (middle) errors with no outliers in the data. Rotation errors with 33% of outliers (right)

5.2 Real Sequence Example

In this section a test consisting of 340 frames is described in detail. More sequences can be found in [25]. Small features (fiducials) were placed in known 3D locations (see Table 1) on the floor and on the walls of a room (see Fig. 2). Distances were measured with a tape having a resolution of 1 millimeter (0.1 cm). The fixed world coordinate system was chosen with its origin coinciding with a known junction on the floor tiles, the X and Z axes on the floor plane and parallel to the floor tiles and the Y axis pointing downwards (with -Y measuring the height above the floor). The balls are 1.4 and the circles are 1cm in diameter, the tiles are squares of 30x30cm.

Table 1. Scene fiducial geometry

Serial number	Type	Color	World axes location [cm]		
			X	Y	Z
1	Ball	Blue	30	-0.7	180
2	Ball	Green	30	-0.7	210
3	Ball	Yellow	-60	-0.7	240
4	Ball	Light blue	30	-0.7	240
5	Ball	Black	0	-0.7	270
6	Ball	Red	-30	-0.7	330
7	Ball	Orange	60	-0.7	360
8	Circle	Light blue	-31	-100.3	388
9	Circle	Light blue	29	-120.7	492.5

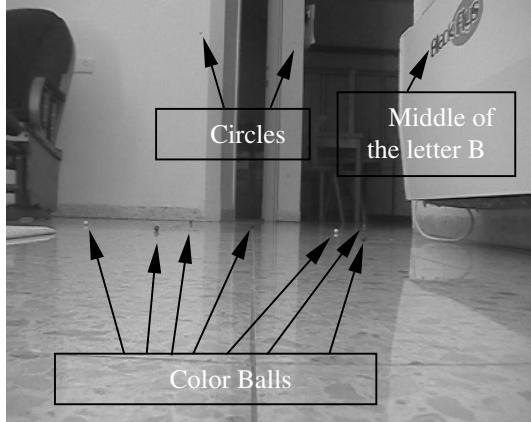


Fig. 2. First frame of the sequence

5.2.1 Camera Setup and Motion

The camera was a Panasonic NV-DS60 PAL color camera with a resolution of 720x576 pixels. The camera zoom was fixed throughout the test at the widest viewing angle. A wide field of view reduces the angular accuracy of a pixel, but enables the detection of more features (overall, [5] has experimentally found that a wide viewing angle is favorable for motion and structure estimation). The camera projection parameters at this zoom were obtained from a calibration process:

$$x = 938X/Z + 360.5 ; \quad y = 1004Y/Z + 288.5$$

The camera was initially placed on the floor with the optical axis pointing approximately in the Z direction of the world. The camera was moved backwards by hand on the floor plane with the final orientation approximately parallel to the initial (using the tile lines). The comparison between the robust and the EKF approach is made with both having the same motion parameters in the estimated state, the same measurements and the same knowledge of the 3D data of table 1. The acceleration disturbance parameters for both methods are: $\sigma_{\dot{\omega}} = 0.003$, $\sigma_{\dot{v}} = 0.0005$. The number of particles is 2000 and the robust distance function parameter is L=4 pixels.

5.2.2 Feature Tracking

The features were tracked with a Kanade-Lucas-Tomasi (KLT) type feature tracker (see [21]). The tracker was enhanced for color images by minimizing the sum of squared errors in all three RGB color channels (the standard KLT is formulated for grayscale images). The tracking windows of size 9x9 pixels were initialized in the first frame at the center of each ball and circle by manual selection. To avoid the fatal effect of interlacing, the resolution was reduced in the vertical image plane by sampling every two pixels (processing one camera field), the sub-pixel tracking results were then scaled to the full image resolution.

5.2.3 Motion Estimation Results

The results obtained by the proposed robust approach and the EKF approach are shown in Fig. 3. Most of the motion is in the Z direction. The final position was at

approximately $Z=60\text{cm}$. The robust approach estimates the value of $Z=61\text{cm}$ at the end of the motion (there is some uncertainty regarding the exact location of the camera focal center), the estimated Y coordinate is almost constant and equal to the camera lens center height above the floor (about -7.4 cm). The trajectory estimated by the EKF is totally different with $Z \approx -80\text{ cm}$ and $Y \approx 30\text{ cm}$ at the end of the motion.

The deviation from the expected final camera position is by two orders of magnitude higher than the expected experimental accuracy, the EKF estimation is therefore erroneous. After observing the tracking results of all the feature points, the points 1,2,4,5,6,8 were manually selected as the inlier points (those which reasonably track the appropriate object throughout the sequence). Running again both estimators with only the inlier points, the proposed approach results are almost unchanged, while the EKF estimation changes drastically, now producing a trajectory similar to the robust approach (see Fig. 4). It should be noted that the EKF estimation produces a smoother trajectory. Image plane errors between the measurements and the projected 3D structure are shown in Fig. 5 (corresponding to the motion estimation of Fig. 3). The robust method exhibits low errors for most of the features and allows high errors for the outliers (this implies that algorithm can automatically separate the inliers from the outliers by checking the projection errors). The EKF approach on the other hand exhibits large errors for both inlier and outlier features. It should be noted that the outlier features are distracted from the true object due to its small size, noise, similar objects in the background and reflections from the shiny floor. It is possible to improve the feature tracking results by using methodologies from [14], [21], [24], but good feature tracking should be complemented with a robust methodology in order to compensate for occasional mistakes. Although the deficiencies of the EKF approach are mentioned in [5], [6], [7], no examples are given and no remedies are suggested in the camera motion estimation literature. As anonymous reviewers have suggested, we

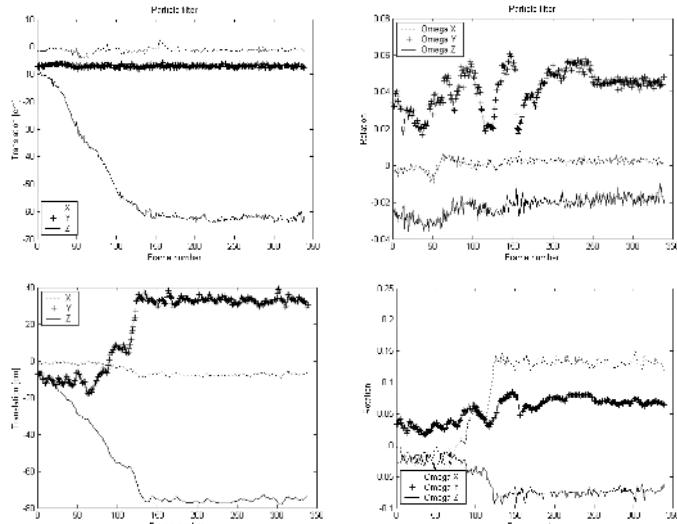


Fig. 3. Estimated camera 3D trajectory using the proposed approach (*upper row*) and the EKF approach (*lower row*)

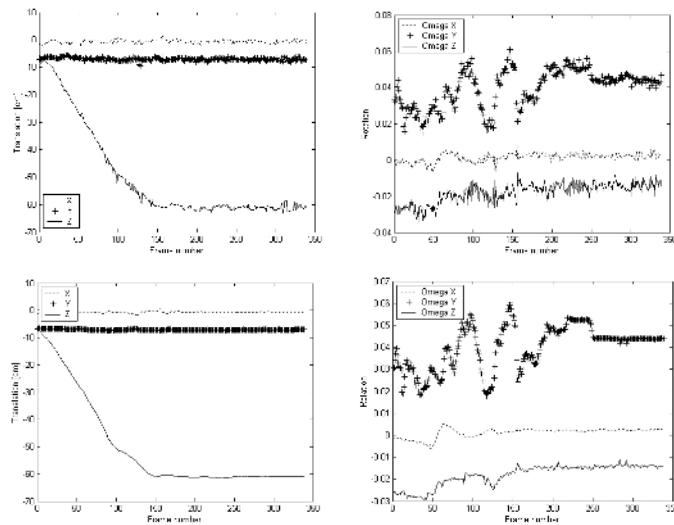


Fig. 4. Estimated camera 3D trajectory using only the inlier points, the proposed approach (upper row) and the EKF approach (lower row)

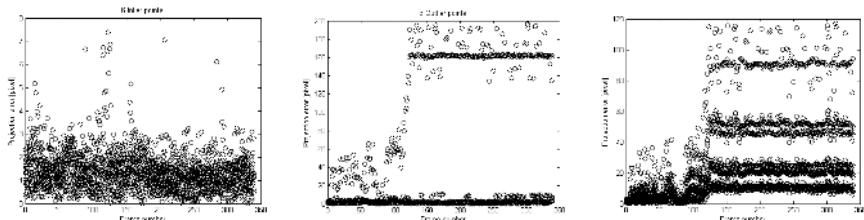


Fig. 5. Image plane errors. Robust approach showing the 6 inliers (left) and 3 outliers (middle). EKF approach with all 9 features (right)

examined two methods of making the EKF solution more robust: 1. By incorporating measurements only from features which have a geometric error norm below a threshold and 2. By applying the robust distance function on the norm of the geometric error of each feature. Both failed to improve the results of the EKF. Rejection of outliers in Kalman filtering may succeed if the outliers appear scarcely or when their proportion is small. In our example these conditions are clearly violated.

5.2.4 Structure Computation Example

Structure of unknown features in the scene can be recovered using the estimated camera motion obtained by the robust method and the image measurements in a partially decoupled scheme as explained in section 4.2. As an example, consider the middle of the letter B appearing on the air conditioner which was tracked from frame 0 to frame 50 (it is occluded shortly afterwards). The reconstructed location of this point in the world axes is: $X=42.3\text{cm}$; $Y=-45.2\text{cm}$; $Z=159.6\text{cm}$. The tape measure world axes location is: $X=42.0\text{cm}$; $Y=-43.7\text{cm}$; $Z=155\text{cm}$. The X , Y , Z differences

are: 0.3, 1.5 and 4.6 [cm] respectively. As expected, the estimation error is larger along the optical axis (approximately the world's Z axis). The accuracy is reasonable, taking into account the short baseline of 19cm produced during the two seconds of tracking this feature (the overall translation from frame 0 to frame 50). As discussed in [5], a long baseline improves the structure estimation accuracy when the information is properly integrated over time.

6 Conclusion

A robust framework for camera motion estimation has been presented with extensions to the solution of the SLAM problem. The proposed algorithm can tolerate about 33% of outliers and it is superior in robustness relative to the commonly used EKF approach. It has been shown that a small number of visible features with known 3D structure are enough to determine the 3D pose of the camera. It may be implied from this work that some degree of decoupling between the motion estimation and structure recovery is a desirable property of SLAM algorithms which trades some accuracy loss for increased robustness. The robust distance function used in this work is symmetric for all the features with the underlying assumption that the probability of a feature to be an inlier or an outlier is independent of time. However, in most cases, a feature is expected to exhibit a more consistent behavior as an outlier or an inlier. This property may be exploited for further improvement of the algorithm's robustness and accuracy. Also, an interesting question for future work is: How to construct fiducials which can be quickly and accurately identified in the scene for camera localization purposes.

References

1. A. Azarbayejani and A. Pentland. Recursive Estimation of Motion, Structure and Focal Length. *IEEE Trans. PAMI*, Vol. 17, no. 6, pp. 562-575, 1995.
2. M. J. Black and P. Anandan. The Robust Estimation of Multiple Motions: Parametric and Piece-wise Smooth Flow Fields. *CVIU*, Vol. 63, No. 1, 1996.
3. M. J. Black and A. Rangarajan. On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications in Early Vision. *IJCV* 19(1), 57-91, 1996.
4. T.J. Broda, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Trans. on AES.*, 26(4):639- 656, 1990.
5. A. Chiuso, P. Favaro, H. Gin. and S. Soatto. Structure from Motion Causally Integrated Over Time. *IEEE. Trans. on PAMI*, Vol. 24, No. 4, 2002.
6. A. J. Davison. Real-Time Simultaneous Localization and Mapping with a Single Camera. *ICCV* 2003.
7. A. J. Davison and D. W. Murray. Simultaneous Localization and Map-Building using Active Vision. *PAMI*, Vol. 24, No. 7, July 2002.
8. O. Faugeras. Three Dimensional Computer Vision, a Geometric Viewpoint. MIT Press, 1993.
9. R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision, Cambridge press 2000.
10. T.S. Huang and A.N. Netravali. Motion and Structure from Feature Correspondences: a review. *Proceeding of The IEEE Communications of the ACM*, 82(2): 252-268, 1994.

11. X. Hu and N. Ahuja. Motion and Structure Estimation Using Long Sequence Motion Models. *Image and Vision Computing*, Vol. 11, no. 9, pp. 549-569, 1993.
12. M. Isard and A. Blake. Visual Tracking by Stochastic Propagation of Conditional Density. Proc. 4th ECCV, Pages 343-356.
13. M. Isard and A. Blake. CONDENSATION - Conditional Density Propagation for Visual Tracking, *Int. J. Computer Vision*, 29, 1, 5-28, 1998.
14. H. Jin, P. Favaro and S. Soatto. Real-time Feature Tracking and Outlier Rejection with Changes in Illumination, ICCV, July 2001.
15. H. Jin. Code from the web site: <http://ee.wustl.edu/~hljin/research/>
16. B.D. Lucas and T. Kanade, An iterative Image Registration Technique with an Application to Stereo Vision, In IJCAI81, pages 674-679, 1981.
17. J. MacCormick and M. Isard, Partitioned Sampling, Articulated Objects and Interface-Quality Hand Tracking. Proc. Sixth European Conf. Computer Vision, 2000.
18. M. Montemerlo and S. Thrun. Simultaneous Localization and Mapping with Unknown Data Association using FastSLAM. Proc. ICRA, 2003, to appear
19. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In AAAI-2002.
20. P.J. Huber. Robust Statistics. Wiley 1981.
21. J. Shi, C. Tomasi. Good Features to Track. CVPR '94, June 1994, pub. IEEE, pp. 593-600.
22. M. Spetsakis and J. Aloimonos. A Multi-Frame Approach to Visual Motion Perception. *Int. J. Computer Vision*, Vol. 6, No. 3, pp. 245-255, 1991.
23. S. Thrun, W. Burgard and D. Fox. A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping. IEEE. International Conference on Robotics and Automation. April 2000.
24. T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making Good Features Track Better. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 178-183, 1998.
25. Web site: <http://www.cs.technion.ac.il/~taln/>

An Adaptive Window Approach for Image Smoothing and Structures Preserving

Charles Kervrann

IRISA - INRIA Rennes / INRA - Mathématiques et Informatique Appliquées
Campus de Beaulieu, 35042 Rennes Cedex, France
ckervran@irisa.fr

Abstract. A novel adaptive smoothing approach is proposed for image smoothing and discontinuities preservation. The method is based on a locally piecewise constant modeling of the image with an adaptive choice of a window around each pixel. The adaptive smoothing technique associates with each pixel the weighted sum of data points within the window. We describe a statistical method for choosing the optimal window size, in a manner that varies at each pixel, with an adaptive choice of weights for every pair of pixels in the window. We further investigate how the I-divergence could be used to stop the algorithm. It is worth noting the proposed technique is data-driven and fully adaptive. Simulation results show that our algorithm yields promising smoothing results on a variety of real images.

1 Introduction

The problem of image recovering is to reduce undesirable distortions and noise while preserving important features such as homogeneous regions, discontinuities, edges and textures. Popular image smoothing algorithms are therefore nonlinear to reduce the amount of smoothing near abrupt changes, i.e. edges. Most of them are based on discrete [3] or continuous [18,21] energy functionals minimization since they are designed to explicitly account for the image geometry. In recent years, a large number of partial differential equations (PDE) and variational methods, including anisotropic diffusion [20,4] and total variation (TV) minimization [21], have shown impressive results to tackle the problem of edge-preserving smoothing [5,4,6,17] and to separate images into noise, texture and piecewise smooth components [16,19].

In this paper, we also address the adaptive image smoothing problem and present a nonparametric estimation method that smooth homogeneous regions and inhibits smoothing in the neighborhood of discontinuities. The proposed *adaptive window approach* differs from previous energy minimization-based methods [3,18,21]. It is conceptually very simple being based on the key idea of estimating a regression function with an adaptive choice of the window size (neighborhood) in which the unknown function is well approximated by a constant. At each pixel, we estimate the regression function by iteratively growing a window and adaptively weighting input data to achieve an optimal compromise

between the bias and variance [14,15,13]. The motivation behind this nonparametric estimation approach is to use a well-established theory in statistics [10,14] for adaptive smoothing, yielding to non-iterative algorithms for 2D-3D imaging. The proposed algorithm complexity is actually controlled by simply restricting the size of the larger window and setting the window growing factor. In contrast to most digital diffusion-based filtering processes for which the input noisy image is “abandoned” after the first iteration [20,4], the adaptive window approach recycles at each step the original data. Other related works to our approach are nonlinear Gaussian filters (iterative or non-iterative bilateral filtering [12,22,7,1,23]) that essentially average values within a local window but changes the weights according to local differences in the intensity [12,22,7,1,23]. However, these weighted schemes use a static window size which can be arbitrarily large, in the both spatial and range domains. Our structure-adaptive smoothing also works in the joint spatial-range domain but has a more powerful adaptation to the local structure of the data since the size of the window and internal parameters are computed from local image statistics.

2 A Statistical Nonparametric Approach

We observe the function u with some additive errors ξ_i :

$$Y_i = u(x_i) + \xi_i, \quad i = 1, \dots, n, \quad (1)$$

were $x_i \in \mathbb{R}^2$ represents the spatial coordinates of the discrete image domain \mathcal{S} of n pixels and $Y_i \in \mathbb{R}$ is the observed intensity at location x_i . We suppose the errors ξ_i to be independent and distributed zero-mean random variables with unknown variances, i.e. $\text{var}(\xi_i) = \sigma_i^2$.

2.1 Image Model and Basic Idea

A classical nonparametric estimation approach is based on the structural assumption that regression function $u(x)$ is constant in a local neighborhood in the vicinity of a point x . An important question under such an approach is first how to determine for each pixel the size and shape of the neighborhood under concern from image data. The regression function $u(x)$ can be then estimated from the observations lying in the estimated neighborhood of x by a local maximum likelihood (ML) method.

Our procedure is iterative and uses this idea. First, suppose we are given a local window $\mathcal{W}_i^{(0)}$ containing the point of estimation x_i . By $\hat{u}_i^{(0)}$ we denote an approximation of $\hat{u}^{(0)}(x_i)$. We can calculate an initial ML estimate $\hat{u}_i^{(0)}$ at point x_i (and its variance $\hat{\sigma}_i^{(0)}$) by averaging observations over a small neighborhood $\mathcal{W}_i^{(0)}$ of x_i as

$$\hat{u}_i^{(0)} = \frac{1}{|\mathcal{W}_i^{(0)}|} \sum_{x_j \in \mathcal{W}_i^{(0)}} Y_j \quad \text{and} \quad \hat{\sigma}_i^{(0)} = \frac{\hat{\sigma}_i^2}{|\mathcal{W}_i^{(0)}|} \quad (2)$$

where $\hat{\sigma}_i^2$ is a pilot estimator which can be plugged in place of the noise variance σ_i^2 and $|\mathcal{W}_i^{(0)}|$ denotes the number of points $x_j \in \mathcal{W}_i^{(0)}$. At the next iteration, a larger neighborhood $\mathcal{W}_i^{(1)}$ with $\mathcal{W}_i^{(0)} \subset \mathcal{W}_i^{(1)}$ centered at x_i is considered, and every point x_j from $\mathcal{W}_i^{(1)}$ gets a weight $w_{ij}^{(1)}$ which is defined by comparing the estimates $\hat{u}_i^{(0)}$ and $\hat{u}_j^{(0)}$ obtained at the first iteration. Then we recalculate the estimate $\hat{u}_i^{(1)}$ as the weighted average of data points lying in the neighborhood $\mathcal{W}_i^{(1)}$. We continue this way, growing with k the considered neighborhood $\mathcal{W}_i^{(k)}$; for each $k \geq 1$, the ML estimate $\hat{u}_i^{(k)}$ and its variance are given by:

$$\hat{u}_i^{(k)} = \sum_{x_j \in \mathcal{W}_i^{(k)}} w_{ij}^{(k)} Y_j \quad \text{and} \quad \hat{\vartheta}_i^{(k)} = \hat{\sigma}_i^2 \sum_{x_j \in \mathcal{W}_i^{(k)}} [w_{ij}^{(k)}]^2 \quad (3)$$

where weights $w_{ij}^{(k)}$ are continuous variables ($0 \leq w_{ij}^{(k)} \leq 1$), computed by comparison of the preceding estimates $\hat{u}_i^{(k-1)}$ and $\hat{u}_j^{(k-1)}$. Note we can also write $\hat{u}_i^{(k)} = \sum_{j=1}^n 1_{\{x_j \in \mathcal{W}_i^{(k)}\}} w_{i,j}^{(k)} Y_j$ where $1_{\{x_j \in \mathcal{W}_i^{(k)}\}}$ is the spatial rectangular kernel. In the following, we use a spatial rectangular kernel (square windows) for mathematical convenience, but the method can be easily extended to the case of a more usual spatial Gaussian kernel [12,22,7,1,23]. Moreover, we choose an optimal window for each pixel x_i by comparing the new estimate $\hat{u}_i^{(k)}$ with the estimate $\hat{u}_i^{(k-1)}$ obtained at the preceding iteration. Finally, a global convergence criterion is introduced to stop the estimation procedure.

At this level, an important connection between our method and local robust M-estimation [7] should be mentioned. In Equation (3), the weight function $w_{ij}^{(k)}$ does not depend on input data but are only calculated from neighboring local estimates, which contributes to the regularization performance. In addition, the method is able to recover a piecewise smooth image even the underlying image model is locally constant as it is confirmed in our experiments (see Fig. 2). The approach is similar also (at least in spirit) to twofold weighting schemes employed in the bilateral filtering [22,1,23] and *mean shift*-based algorithms [8].

2.2 Adaptive Weights

In our approach, we may decide on the basis of the estimates $\hat{u}_i^{(k-1)}$ and $\hat{u}_j^{(k-1)}$, whether a point x_i and $x_j \in \mathcal{W}_i^{(k)}$ are in the same region and then prevent significant discontinuities oversmoothing. In the local Gaussian case, significance is measured using a contrast $|\hat{u}_i^{(k-1)} - \hat{u}_j^{(k-1)}|$. If this contrast is high compared to $\sqrt{\hat{\vartheta}_i^{(k-1)}}$, then x_j is an outlier and should not participate to the estimation of $\hat{u}_i^{(k)}$ and $w_{ij}^{(k)} \rightarrow 0$.

Hence, motivated by the robustness and smoothing properties of the Huber M-estimator in the probabilistic approach of image denoising [2], we introduce

the following related weight function (but other weight functions are possible [4,23])

$$w_{ij}^{(k)} = \frac{g_{ij}^{(k)}}{\sum_{x_j \in \mathcal{W}_i^{(k)}} g_{ij}^{(k)}}, \quad g_{ij}^{(k)} = \begin{cases} 1 & \text{if } |\hat{u}_i^{(k-1)} - \hat{u}_j^{(k-1)}| \leq \lambda \sqrt{\hat{\vartheta}_i^{(k-1)}} \\ \frac{\lambda \sqrt{\hat{\vartheta}_i^{(k-1)}}}{|\hat{u}_i^{(k-1)} - \hat{u}_j^{(k-1)}|} & \text{otherwise.} \end{cases} \quad (4)$$

Here $\lambda \sqrt{\hat{\vartheta}_i^{(k-1)}}$ is related to the spatially varying fraction of contamination of the Gaussian distribution: for the majority of points $x_j \in \mathcal{W}_i$, the values $\hat{u}_i^{(k-1)} - \hat{u}_j^{(k-1)}$ can be approximatively modeled as being constant (zero) with random Gaussian noise. Hence λ is an appropriate quantile of the standard normal distribution and depends on the level of noise in images. In our experiments, we arbitrarily set $\lambda = 3$ according to the well known “rule of 3 sigma”.

2.3 Optimal Window Selection

Statistical inference under such a structural assumption focuses on searching for every point x_i the largest neighborhood (window) \mathcal{W}_i where the hypothesis of structural homogeneity is not rejected. In other words, we seek to estimate a regression function u from the data, while having deal with a so-called nuisance parameter, that is the neighborhood. The classical measure of the closeness of the estimator \hat{u} obtained in the window \mathcal{W}_i to its target value u is the mean squared error (MSE) which is decomposed into the sum of the squared bias $[\text{Bias}(\hat{u}_i)]^2$ and variance $\hat{\vartheta}_i$.

As explained before, we should choose a window that achieves an optimal compromise between the squared bias and variance for each pixel. Accordingly, we make the reasonable assumption that the squared bias is an increasing function of the neighborhood size and the variance is a decreasing function of the neighborhood size. Then, in order to minimize the MSE we search for the window where the squared bias and the variance of the estimate are equal. The corresponding critical MSE is ($E[\cdot]$ denotes the mathematical expectation):

$$\text{MSE} = E \left[\hat{u}_i^{(k)} - u(x_i) \right]^2 = \left[\text{Bias} \left(\hat{u}_i^{(k)} \right) \right]^2 + \hat{\vartheta}_i^{(k)} = 2\hat{\vartheta}_i^{(k)}. \quad (5)$$

Now, let us introduce a finite set of windows $\{\mathcal{W}_i^{(0)}, \dots, \mathcal{W}_i^{(k_M)}\}$ centered at $x_i \in \mathcal{S}$, with $\mathcal{W}_i^{(k)} \subset \mathcal{W}_i^{(k+1)}$, starting with a small $\mathcal{W}_i^{(0)}$ and the corresponding estimates $\hat{u}_i^{(0)}$ of the true image $u(x_i)$. Denote by \hat{k}_i the ideal window size corresponding to the minimum value of the pointwise MSE at location x_i . Accordingly, $\mathcal{W}_i^{(\hat{k}_i)}$ can be obtained according to the following statistical pointwise rule [14,15,13]:

$$\hat{k}_i = \max \left\{ k : \forall k' < k : \left| \hat{u}_i^{(k)} - \hat{u}_i^{(k')} \right|^2 \leq 8\hat{\vartheta}_i^{(k')} \right\}. \quad (6)$$

In other words, as long as successive estimates $\hat{u}_i^{(k)}$ stay close to each other, we decide that the bias is small and the size of the estimation window can be increased to improve the estimation of the constant model (and to decrease the variance of the estimate $\hat{u}_i^{(k)}$). If an estimated point $\hat{u}_i^{(k')}$ appears far from the previous ones, we interpret this as the dominance of the bias over the variance term. For each pixel, the detection of this transition enables to determine the critical size that balances bias and variance. Note the choice of the detection threshold in (6) between 2^2 and 4^2 does not change the result of the algorithm significantly.

2.4 Global Stopping Rule

A stopping rule can be used to save computing time if two successive solutions are very close and prevent an useless setting of the larger window size. Here, we adopt the so-called Csiszár's I-divergence [9] to detect global convergence:

$$I(\hat{u}^{(k)}, \hat{u}^{(k+1)}) = \sum_{i=1}^n \left[\hat{u}_i^{(k)} \log \frac{\hat{u}_i^{(k)}}{\hat{u}_i^{(k+1)}} - \hat{u}_i^{(k)} + \hat{u}_i^{(k+1)} \right]. \quad (7)$$

We choose this criterion to obtain the distance between succeeding iterations since the decorrelation criterion proposed in [17] tends to underestimate the number of necessary iterations in our framework. In addition, the I-divergence criterion can be used for a variety of algorithms, as it does not directly depend on the restoration method. In practice, the I-divergence is normalized with its maximal occurring value at iteration $k = 0$. When $I(\hat{u}^{(k)}, \hat{u}^{(k+1)})$ sinks under a threshold (of the order 10^{-3} for typical images) that sufficiently represents convergence, the algorithm is stopped at the final iteration $k_c = k$, with $k_c \leq k_M$. Finally, the window size increases at each iteration k if the global convergence criterion is not met (or $k \leq k_M$) without changing the estimate $\hat{u}_i^{(k)}$ if the pointwise rule (6) has already been violated at x_i , i.e. $\hat{u}_i^{(k)} = \hat{u}_i^{(\hat{k}_i)}$ if $\hat{k}_i < k$. If the rule (6) has not been violated at x_i , we have $k_i = k$ where k is the current iteration of the algorithm.

3 Implementation

The key ingredient of the procedure is an increasing sequence of neighborhoods $\mathcal{W}_i^{(k)}, k = 0, 1, \dots, k_M$ with $\mathcal{W}_i^{(k)} \subset \mathcal{W}_i^{(k+1)}$ centered at each image pixel x_i . In what follows, $|\mathcal{W}_i^{(k)}|$ denotes the number of points x_j in $\mathcal{W}_i^{(k)}$, i.e. $|\mathcal{W}_i^{(k)}| = \#\{x_j \in \mathcal{W}_i^{(k)}\}$. In our experiments, we arbitrarily use neighborhoods $\mathcal{W}^{(k)}$ corresponding to successive square windows of size $|\mathcal{W}^{(k)}| = (2k+1) \times (2k+1)$ pixels with $k = 0, 1, 2, \dots, k_M$.

The estimation procedure described in Section 2.1 relies also on the preliminary local estimation of the noise variance. This estimation is an (off-line)

pre-processing step to initialize the adaptive smoothing procedure. In most applications, the noise variance σ_i^2 at each image point x_i is unknown and an estimate $\hat{\sigma}_i^2$ can be obtained from data as

$$\hat{\sigma}_i^2 = \frac{1}{|\mathcal{B}_i|} \sum_{x_j \in \mathcal{B}_i} \varepsilon_j^2 \quad (8)$$

where \mathcal{B}_i denotes a square block of pixels centered at x_i and local residuals ε_j are defined as (we note Y_{j_1, j_2} the observation Y_j at site $j = (j_1, j_2)$) [10]:

$$\varepsilon_j = \frac{4Y_{j_1, j_2} - (Y_{j_1+1, j_2} + Y_{j_1-1, j_2} + Y_{j_1, j_2+1} + Y_{j_1, j_2-1})}{\sqrt{20}}. \quad (9)$$

In presence of discontinuities in the block \mathcal{B}_i , a local estimate of the noise variance based on robust statistics is preferable. In this framework, high discontinuities within the region \mathcal{B}_i correspond to statistical outliers with respect to local image contrasts. As in [4], we suggest to define $\hat{\sigma}_i^2 = \max(\hat{\sigma}^2, \hat{\sigma}_{\mathcal{B}_i}^2)$ where $\hat{\sigma}^2$ and $\hat{\sigma}_{\mathcal{B}_i}^2$ are respectively the robust estimates of the noise variance computed for the entire image and within a local block centered at point x_i . We propose the following robust estimates for $\hat{\sigma}$ and $\hat{\sigma}_{\mathcal{B}_i}$:

$$\hat{\sigma} = 1.4826 \text{ median}(|\varepsilon_{\mathcal{S}}| - \text{median}|\varepsilon_{\mathcal{S}}|) \quad (10)$$

$$\hat{\sigma}_{\mathcal{B}_i} = 1.4826 \text{ median}(|\varepsilon_{\mathcal{B}_i}| - \text{median}|\varepsilon_{\mathcal{B}_i}|) \quad (11)$$

where $\varepsilon_{\mathcal{S}} = \{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$ is the set of n local residuals of the entire image, $\varepsilon_{\mathcal{B}_i}$ is the set of $|\mathcal{B}_i|$ local residuals within the local block \mathcal{B}_i and the constant is derived from the fact that the median absolute deviation of a zero-mean normal distribution with unit variance is $0.6745 = 1./1.4826$. The local estimation $\hat{\sigma}_{\mathcal{B}_i}^2$ of noise variance is useful for filtering off spatially varying textures. The global estimate of the noise variance provides also a reasonable lower bound and prevents the amplification of noise in relatively homogeneous areas. From a practical point of view, we have explored the computation of the local estimate of the noise variance within blocks \mathcal{B}_i of size $(2k_M + 1) \times (2k_M + 1)$, i.e. $\mathcal{B}_i = \mathcal{W}_i^{(k_M)}$ at every location x_i in the image.

Below, we give the proposed algorithm:

- **Initialization:** For each point x_i , we calculate initial estimates $\hat{u}_i^{(0)}$ and $\hat{v}_i^{(0)}$ using Equation (2) and set $k = 1$. We naturally choose $|\mathcal{W}^{(0)}| = 1$, i.e. the initial local neighborhood $\mathcal{W}^{(0)}$ contains only x_i . Here $\hat{\sigma}_i^2$ is the noise variance robustly estimated at each point x_i from data as it has been explained before.
- **Estimation:** For all x_j in $\mathcal{W}_i^{(k)}$, we compute weights $w_{ij}^{(k)}$ using Equation (4) and new estimates $\hat{u}_i^{(k)}$ and $\hat{v}_i^{(k)}$ using Equation (3).
- **Pointwise control:** After the estimate $\hat{u}_i^{(k)}$ has been computed, we compare it to the previous estimates $\hat{u}_i^{(k')}$ at the same point x_i for all $k' < k$. If the pointwise rule (6) is violated at iteration k , we do not accept $\hat{u}_i^{(k)}$ and keep

- the estimates $\hat{u}_i^{(k-1)}$ from the preceding iteration as the final estimate at x_i (i.e. $\hat{k}_i = k - 1$ at point x_i). The estimate at x_i is unchanged if $k > \hat{k}_i$.
- **Convergence:** Stop the procedure if $k = k_M$ or if $I(\hat{u}^{(k)}, \hat{u}^{(k+1)}) < 10^{-3}$, otherwise increase k by 1 and continue with the estimation step. We use the parameter k_M to bound the numerical complexity. As expected, increasing k_M allows for additional variance reduction in homogeneous regions but usually does not change the estimates in the neighborhood of discontinuities. In our experiments, $k_M = 15$ satisfies a good compromise and over-estimates the number of necessary iterations.

Using this algorithm, it can be shown that the average gray level of the smoothed image is not affected by the adaptive window procedure, i.e. $\frac{1}{n} \sum_{i=1}^n Y_i = \frac{1}{n} \sum_{i=1}^n \hat{u}_i^{(\hat{k}_i)}$ and the *extremum principle* is guaranteed: $\min_j Y_j \leq \hat{u}^{(k)} \leq \max_j Y_j, \forall x_i \in \mathcal{S}, \forall k \leq k_M$.

4 Image Decomposition

The proposed adaptive window approach is able to extract a piecewise smooth image, but significant textures and oscillating patterns are actually excluded. The purpose of this section is to briefly show that a relatively simple modification of the above global procedure yields an algorithm able to separate images into noise, texture and piecewise smooth (or *cartoon*) parts. In general the image-signal is assumed to be corrupted by an additive zero-mean white Gaussian noise with a constant variance. Therefore, the key idea is first to remove noise from the input image by setting $\hat{\sigma}_i = \hat{\sigma}$ in the estimation procedure described in Section 3.1. If the original image consists of three additive components (noise, texture, *cartoon*) [16,19,11], the texture component is simply obtained by computing the difference between the noise-free image and the piecewise smooth image. The piecewise smooth image is estimated as described earlier in the paper, i.e. by considering local estimations of the noise-texture variances $\hat{\sigma}_i^2$ in the procedure. In areas containing little texture this difference is close to zero since $\hat{\sigma}_{\mathcal{B}_i}^2$ is likely to be less than $\hat{\sigma}^2$ in these areas. While the experimental results are promising using this simple mechanism, we do not pretend that it is able to decompose any images into the three main components under concern ([16,19,11]).

5 Experiments

The potential of the adaptive window method is first shown on a synthetic-natural image (Fig. 1a) corrupted by an additive white-Gaussian noise (Fig. 1b, PSNR = 28.3 db, $\sigma = 10$). In Fig. 1c, the noise and small textures are reduced in a natural manner and significant geometric features such as corners and boundaries, and original contrasts are visually well preserved (PSNR = 31 db). In this experiment, the final iteration $k_c = 11$ was determined autonomously by the I-divergence criterion ($k_M = 15$). To enhance the visibility of the results,

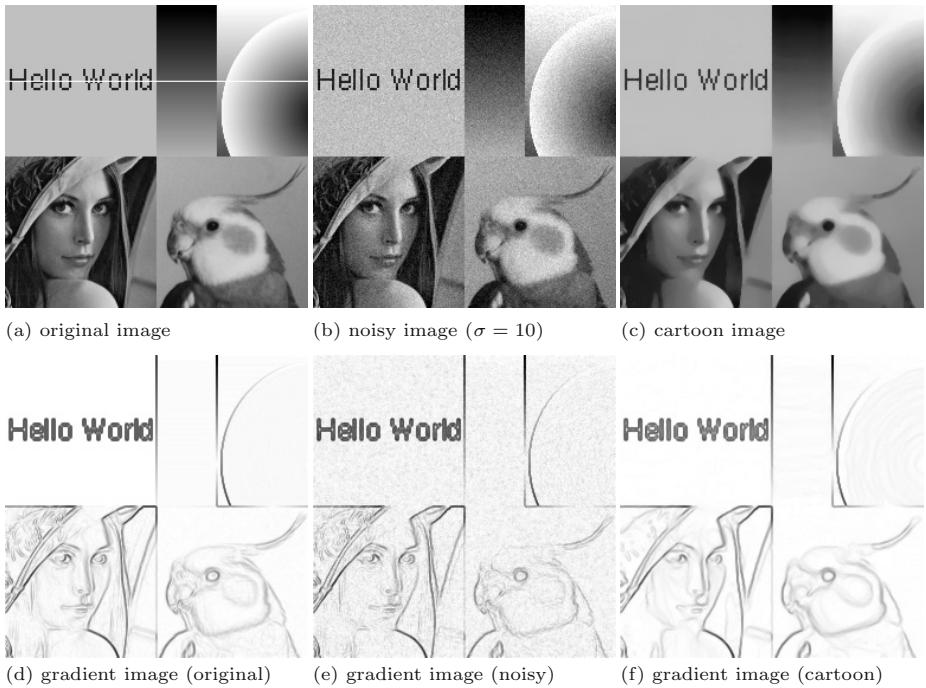


Fig. 1. An example of adaptive smoothing results.

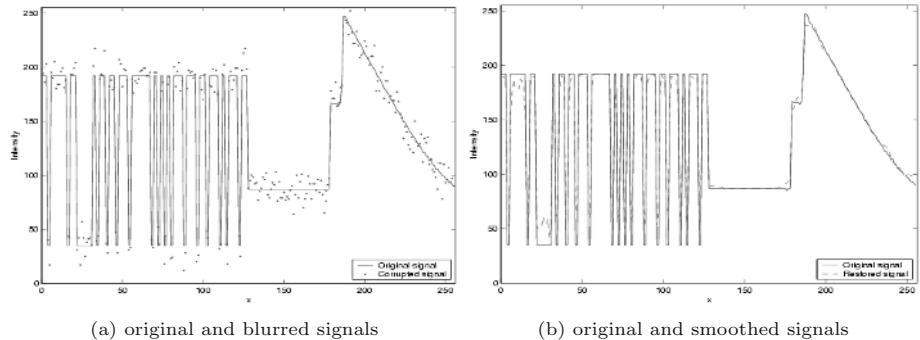
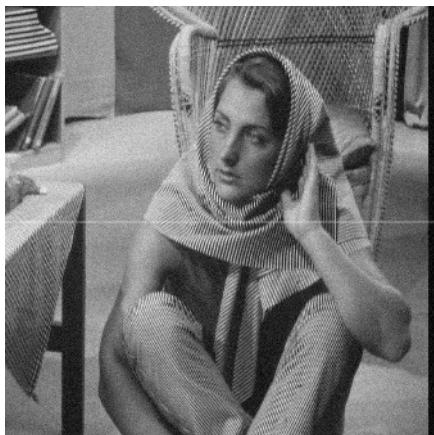


Fig. 2. A horizontal cross-section corresponding to line 128 drawn in white in Fig. 1a.

we have computed the gradient norm on each of the three images; in Figs. 1d-f, high gradient values are coded with black and zero gradient with white. To better appreciate the smoothing process, a horizontal cross-section marked in white in Fig. 1a is graphically depicted in Fig. 2. The abrupt changes are correctly located and satisfying smooth variations of the signal are recovered (Fig. 2b). The processing of a 256×256 image required typically 15 seconds on a PC (2.6 Ghz, Pentium IV) using a standard C++ implementation of the algorithm.



(a) noisy image ($\sigma = 10$)



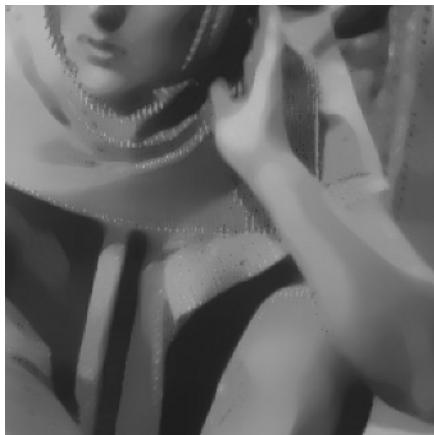
(b) noise removal using our method



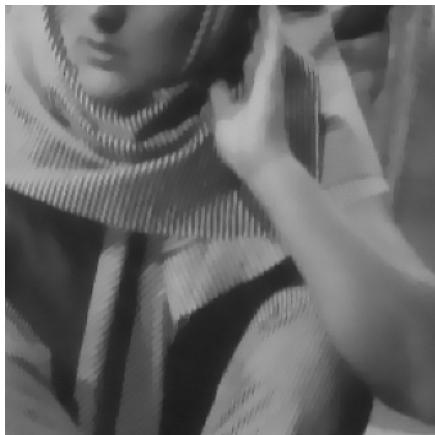
(c) cartoon-like image using our method



(d) noise-free texture component



(e) part of the cartoon-like image using our method



(f) part of the cartoon-like image by TV minimization [21]

Fig. 3. Processing of the noisy 512×512 *Barbara* image.

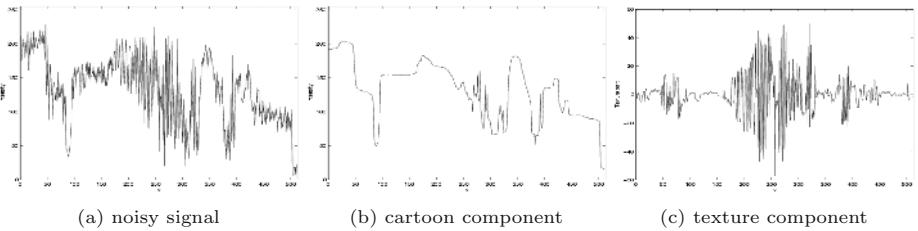


Fig. 4. A horizontal cross-section corresponding to line 256 drawn in Fig. 3a.

In a second example, the effects of the adaptive window approach are illustrated using the well-known 512×512 *Barbara* image (Fig. 3). It turns out that the results of the adaptive window approach and TV minimization are visually similar when they are applied on the original image corrupted by an additive white-Gaussian noise (Fig. 3a, $\sigma = 10$). In this experiment, the noise-free image shown in Fig. 3b, has been obtained by setting $\hat{\sigma}_i = \hat{\sigma}$ in the adaptive window procedure. Additionally, our method is also able to reliably estimate the piecewise smooth component as shown in Fig. 3c and Fig. 3e. Small textures on clothes in the original image are nearly eliminated after $k_c = 11$ iterations (automatically detected). In Fig. 3f, the TV minimizing process [21] does not completely eliminate small textures without blurring edges even if the balance between the fidelity and regularizing terms are modified. Finally, the estimated texture component shown in Fig. 3d correspond to the difference between the piecewise smooth image (Fig. 3b) and the noise-free image (Fig. 3a). In Fig. 4, the horizontal cross-section marked in white in Fig. 3a is depicted to better appreciate the image decomposition. The method provides also some additional structural information about the image. Figure 5a shows the results of local estimations of the noise-texture variance $\hat{\sigma}_i^2$ within local windows ($k_M = 15$). Dark areas have higher values of $\hat{\sigma}_i^2$ and correspond to more textured image regions; bright areas correspond to uniform regions, i.e. $\hat{\sigma}_i^2 = \hat{\sigma}^2$. Figure 5b shows the locations and sizes of optimal estimation windows; we have coded small windows with black and large windows with white. As expected, small windows are in the neighborhood of image gradients shown in Fig. 5e. The histogram of the windows sizes is also shown in Fig. 5f. Finally, Figures 5c and 5d show respectively the values $\{|\mathcal{W}_i^{(k_i)}|^{-1} \sum_{x_j \in \mathcal{W}_i^{(k_i)}} g_{ij}^{(k_i)}\}$ and $\{\hat{\vartheta}_i^{(k_i)}\}$ where dark values correspond to image discontinuities (Fig. 5c) and to low-confidence estimates (Fig. 5d).

We have also tested the algorithm (also implemented for processing 3D data) on a 3D confocal fluorescence microscopy image that contain complex structures. A typical 2D image taken from the 3D stack of 80 images depicts glomeruli in the antennal lobes of the moth olfactory system (Fig. 6a). The smoothed 3D image have larger homogeneous areas than the original 3D image (Fig. 6b). The corresponding noise-free texture image is shown in Fig. 6c. Here, this image decomposition is useful to extract regions/volumes of interest.

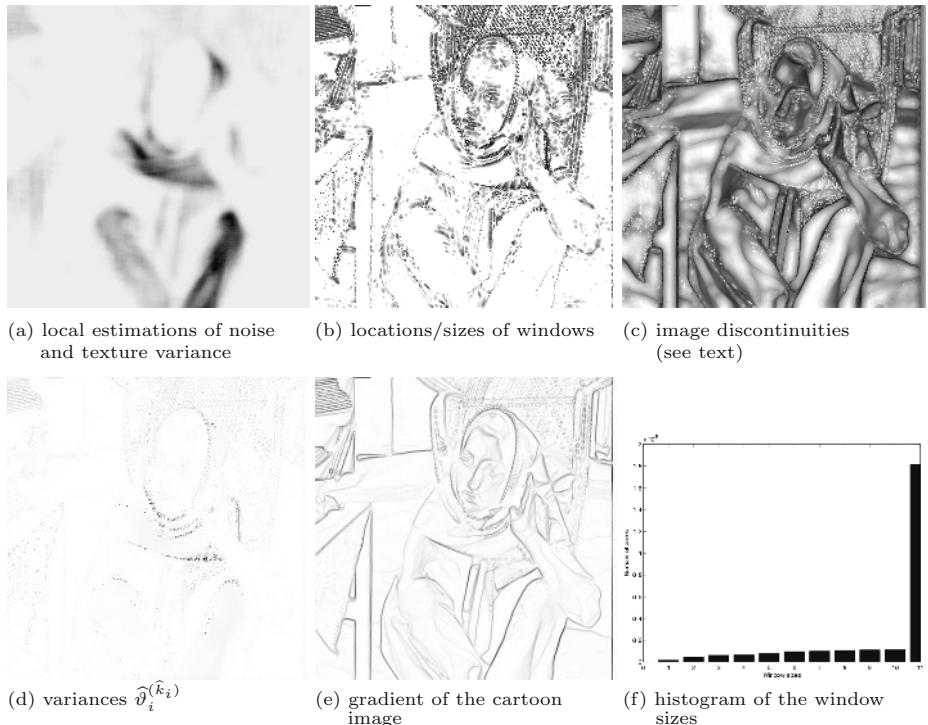


Fig. 5. Results of the cartoon-like 512×512 *Barbara* image.

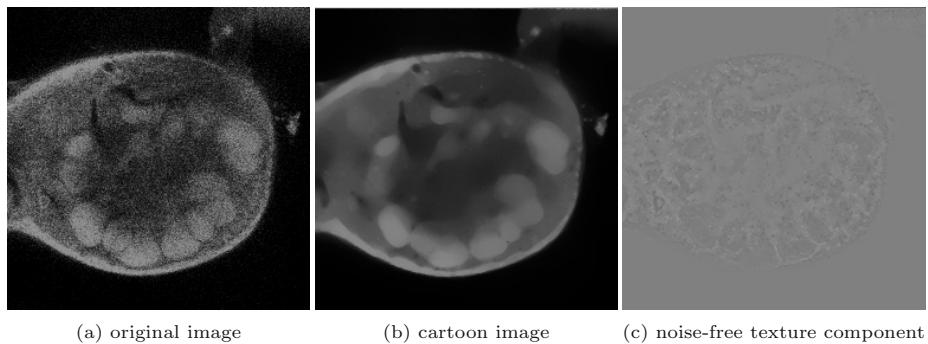


Fig. 6. Decomposition of a 3D confocal microscopy image.

6 Summary and Conclusions

We have described a novel feature-preserving adaptive smoothing algorithm where local statistics and variable window sizes are jointly used. Since $|\mathcal{W}^{(k)}|$ grows exponentially in our set-up, the whole complexity of the proposed algorithm is of order $O(n|\mathcal{W}^{(k_c)}|)$ if an image contains n pixels and $k_c \leq k_M$. In

addition, the proposed smoothing scheme provides an alternative method to the anisotropic diffusion and bilateral filtering or energy minimization methods. An advantage of the method is that internal parameters can be easily calibrated using statistical arguments. Experimental results show demonstrate its potential for image decomposition into noise, texture and piecewise smooth components.

Acknowledgments. We thank Sileye Ba for his contribution to this work.

References

1. D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 24(6): 844-847, 2002.
2. A. Ben Hamza, H. Krim. A variational approach to maximum a posteriori estimation for image denoising. In *Proc. EMMCVPR'01*, LNCS 2134, pp. 19-34, Sophia-Antipolis, France, 2001.
3. A. Blake, A. Zisserman. *Visual Reconstruction*, MIT Press, 1987.
4. M.J. Black, G. Sapiro. Edges as outliers: Anisotropic smoothing using local image statistics. In *Proc. Scale-Space'99*, LNCS 1682, pp. 259-270, Kerkyra, Greece, 1999.
5. F. Catte, P.-L. Lions, J.-M. Morel, T. Coll. Image selective smoothing and edge-detection by nonlinear diffusion. *SIAM J. Numerical Analysis*, 29(1): 182-193, 1992.
6. T. Chan, S. Osher, J. Shen. The digital TV filter and nonlinear denoising. *IEEE Trans. Image Process.*, 10(2): 231-241, 2001.
7. C.K. Chu, K. Glad, F. Godtliebsen, J.S. Marron. Edge-preserving smoothers for image processing. *J. Am. Stat. Ass.*, 93(442): 526-555., 1998.
8. D. Comaniciu, P. Meer. Mean-shift: a robust approach toward feature space analysis. *IEEE Trans. Patt. Anal. Mach. Intel.*, 24(5): 603-619, 2002.
9. I. Csiszár. Why least squares and maximum entropy ? An axiomatic approach to inference for linear inverse problems. *Ann. Statist.*, 19: 2032-2066, 1991.
10. T. Gasser, L. Sroka, C. Jennen Steinmetz. Residual variance and residual pattern in nonlinear regression. *Biometrika*, 73: 625-633, 1986.
11. G. Gilboa, N. Sochen, Y.Y. Zeevi. Texture preserving variational denoising using an adaptive fidelity term. In *Proc. VLSM'03*, Nice, France, 2003.
12. F. Godtliebsen, E. Spjotvoll, J.S. Marron. A nonlinear Gaussian filter applied to images with discontinuities, *J. Nonparametric Statistics*, 8: 21-43, 1997.
13. A. Juditsky. Wavelet estimators: adapting to unknown smoothness. *Math. Methods of Statistics*, 1:1-20, 1997.
14. O. Lepski. Asymptotically minimax adaptive estimation 1: uppers bounds. *SIAM J. Theory of Prob. and Appl.*, 36(4): 654-659, 1991.
15. M. Maurizot, P. Bouthemy, B. Delyon, A. Juditski, J.-M. Odobez. Determination of singular points in 2D deformable flow fields. In *IEEE Int. Conf. Image Processing*, Washington DC, 1995.
16. Y. Meyer. Oscillating patterns in image processing and nonlinear evolution equations, *University Lecture Series*, 22, AMS 2002.
17. P. Mrazek. Selection of optimal stopping time for nonlinear diffusion filtering. *Int. J. Comp. Vis.*, 52(2/3): 189-203, 2003.
18. D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and variational problems. *Comm. Pure and Appl. Math.*, 42(5): 577-685, 1989.

19. S. Osher, A. Solé, L. Vese. Image decomposition and restoration using total variation minimization and the H^{-1} norm. *Multiscale Model. Simul.*, 1(3): 349-370, 2003.
20. P. Perona, J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12(7): 629-239, 1990.
21. L. Rudin, S. Osher, E. Fatemi. Nonlinear Total Variation based noise removal algorithms. *Physica D*, 60: 259-268, 1992.
22. C. Tomasi, R. Manduchi. Bilateral filtering for gray and color images. In *Proc. Int Conf. Comp. Vis. (ICCV'98)*, pp. 839-846, Bombay, India, 1998.
23. J. van de Weijer, R. van den Boomgaard. Local mode filtering. In *Proc. Comp. Vis. Patt. Recogn. (CVPR'01)*, vol. II, pp. 428-433, Kauai, Hawaii, USA, 2001.

Extraction of Semantic Dynamic Content from Videos with Probabilistic Motion Models

Gwenaëlle Piriou¹, Patrick Bouthemy¹, and Jian-Feng Yao^{1,2}

¹ IRISA/INRIA,

² IRMAR,

Campus universitaire de Beaulieu, 35042 Rennes cedex, France
{Gwenaelle.Piriou,Patrick.Bouthemy,Jian-Feng.Yao}@irisa.fr

Abstract. The exploitation of video data requires to extract information at a rather semantic level, and then, methods able to infer “concepts” from low-level video features. We adopt a statistical approach and we focus on motion information. Because of the diversity of dynamic video content (even for a given type of events), we have to design appropriate motion models and learn them from videos. We have defined original and parsimonious probabilistic motion models, both for the dominant image motion (camera motion) and the residual image motion (scene motion). These models are learnt off-line. Motion measurements include affine motion models to capture the camera motion, and local motion features for scene motion. The two-step event detection scheme consists in pre-selecting the video segments of potential interest, and then in recognizing the specified events among the pre-selected segments, the recognition being stated as a classification problem. We report accurate results on several sports videos.

1 Introduction and Related Work

Exploiting the tremendous amount of multimedia data, and specifically video data, requires to develop methods able to extract information at a rather semantic level. Video summarization, video retrieval or video surveillance are examples of applications. Inferring concepts from low-level video features is a highly challenging problem. The characteristics of a semantic event have to be expressed in terms of video primitives (color, texture, motion, shape ...) sufficiently discriminant w.r.t. content. This remains an open problem at the source of active research activities.

In [9], statistical models for components of the video structure are introduced to classify video sequences into different genres. The analysis of image motion is widely exploited for the segmentation of videos into meaningful units or for event recognition. Efficient motion characterization can be derived from the optical flow, as in [8] for human action change detection. In [11], the authors use very simple local spatio-temporal measurements, i.e., histograms of the spatial and temporal intensity gradients, to cluster temporal dynamic events. In [10], a principal component representation of activity parameters (such as translation,

rotation ...) learnt from a set of examples is introduced. The considered application was the recognition of particular human motions, assuming an initial segmentation of the body.

In [2], video abstraction relies on a measure of fidelity of a set of key-frames based on color descriptors and a measure of summarizability derived from MPEG-7 descriptors. In [6], spatio-temporal slices extracted in the volume formed by the image sequence are exploited both for clustering and retrieving video shots. Sport videos are receiving specific attention due to the economical importance of sport TV programs and to future services to be designed in that context. Different approaches have been recently investigated to detect highlights in sport videos. Dominant colour information is used in [3].

In this paper, we tackle the problem of inferring concepts from low-level video features and we follow a statistical approach involving modeling, (supervised) learning and classification issues. Such an attempt was recently undertaken for static images in [5]. We are dealing here with concepts related to events in videos, more precisely, to dynamic content. Therefore, we focus on motion information. Since no analytical motion models are available to account for the diversity of dynamic contents to be found in videos, we have to specify and learn them from the image data. To this end, we introduce new probabilistic motion models. Such a probabilistic modelling allows us to derive a parsimonious motion representation while coping with errors in the motion measurements and with variability in motion appearance for a given type of event. We handle in a distinct way the scene motion (i.e., the residual image motion) and the camera motion (i.e., the dominant image motion) since these two sources of motion bring important and complementary information. As for motion measurements, we consider, on one hand, parametric motion models to capture the camera motion, and on the other hand, local motion features to account for the scene motion. They convey more information than those used in [11], while still easily computable contrary to optic flow. They can be efficiently and reliably computed in any video whatever its genre and its content.

We have designed a two-step event detection method to restrict the recognition issue to a limited and pertinent set of classes since probabilistic motion models have to be learnt for each class of event to be recognized. This allows us to simplify the learning stage, to save computation time and to make the overall detection more robust and efficient. The first step consists in selecting candidate segments of potential interest in the processed video. Typically, for sport videos, it involves to select the “play” segments. The second step handles the recognition of the relevant events (in terms of dynamic content) among the segments selected after the first step and is stated as a classification problem.

The remainder of the paper is organized as follows. In Section 2, we briefly present the motion measurements we use. Section 3 is concerned with the probabilistic models introduced to represent the dominant image motion and the residual motion. We describe in Section 4 the two-step event detection method, while the learning stage is detailed in Section 5. Experiments on sports videos are reported in Section 6, and Section 7 contains concluding remarks.

2 Motion Measurements

Let us first briefly describe the motion measurements that we use. Let us point out that the choice of these measurements is motivated by the goal we are pursuing, that is the recognition of important events in videos. This task is intended as a rather qualitative characterization which does not require a full estimation of the image motion.

It is possible to characterize the image motion as proposed in [4], by computing at each pixel a local weighted mean of the normal flow magnitude. However, the image motion is actually the sum of two motion sources: the dominant motion (which can be usually assumed to be due to camera motion) and the residual motion (which is then related to the independent moving objects in the scene, which will be referred to as the scene motion in the sequel). More information can be recovered if we explicitly consider these two motion components rather than the total motion only. Thus, we first compute the camera motion (more precisely, we estimate the dominant image motion) between successive images of the sequence. Then, we cancel the camera motion (i.e., we compensate for the estimated dominant image motion), which allows us to compute local motion-related measurements revealing the residual image motion only.

The dominant image motion is represented by a deterministic 2D affine motion model which is a usual choice:

$$\mathbf{w}_\theta(p) = \begin{pmatrix} a_1 + a_2x + a_3y \\ a_4 + a_5x + a_6y \end{pmatrix}, \quad (1)$$

where $\theta = (a_i, i = 1, \dots, 6)$ is the model parameter vector and $p = (x, y)$ is an image point. This simple motion model can correctly handle different camera motions such as panning, zooming, tracking, (including of course static shots). Different methods are available to estimate such a motion model. We use the robust real-time multiresolution algorithm described in [7]. Let us point out that the motion model parameters are directly computed from the spatio-temporal derivatives of the intensity function. Thus, the camera-motion flow vector $\mathbf{w}_{\hat{\theta}_t}(p)$ is available at each time t and for each pixel p .

Then, the residual motion measurement $v_{res}(p, t)$ is defined as the local mean of the magnitude of normal residual flows weighted by the square of the norm of the spatial intensity gradient. The normal residual flow magnitude is given by the absolute value of the Displaced Frame Difference $DFD_{\hat{\theta}_t}$, evaluated with the estimated dominant motion, and divided by the norm of the image spatial gradient. We finally get:

$$v_{res}(p, t) = \frac{\sum_{q \in \mathcal{F}(p)} \|\nabla I(q, t)\| \cdot |DFD_{\hat{\theta}_t}(q)|}{\max \left(\eta^2, \sum_{q \in \mathcal{F}(q)} \|\nabla I(q, t)\|^2 \right)}, \quad (2)$$

where $DFD_{\hat{\theta}_t}(q) = I(q + \mathbf{w}_{\hat{\theta}_t}(q), t + 1) - I(q, t)$. $\mathcal{F}(p)$ is a local spatial window centered in pixel p (typically a 3×3 window). $\nabla I(q, t)$ is the spatial intensity gradient of pixel q at time t . η^2 is a predetermined constant related to the noise

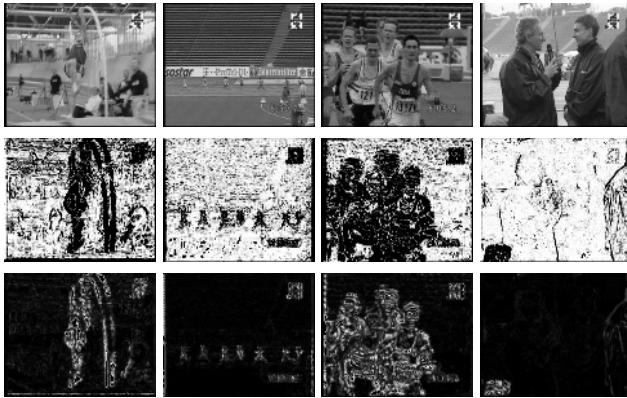


Fig. 1. *Athletics video*: First row: four images of the video. Second row: the corresponding maps of dominant image motion supports (inliers in white, outliers in black). Third row: local residual motion measurements v_{res} (zero-value in black).

level. Such measurements have already been used for instance for the detection of independent moving objects in case of a mobile camera. Figure 1 respectively displays images of an athletic TV program, the corresponding maps of dominant motion support (i.e., the points belonging to the image parts undergoing the estimated dominant motion) and the corresponding maps of residual motion measurements. This example shows that the camera motion is reliably captured even in case of multiple moving elements in the scene since the static background is correctly included in the inliers. It also indicates that the scene motion is correctly accounted for by the residual motion measurements. From relation (2), it can be straightforwardly noted that we only get information related to motion magnitude, and consequently, we lose the motion direction. As demonstrated by the results reported later, this is not a shortcoming since we aim at detecting events, i.e., at determining “qualitative” motion classes. Furthermore, it allows us to manipulate scalar measurements.

3 Probabilistic Modelling of Motion

The proposed method for the detection of important dynamic events relies on the probabilistic modelling of the motion content in a video. Indeed, the large diversity of video contents leads us to favor a probabilistic approach which moreover allows us to formulate the problem of event recognition within a Bayesian framework. Due to the different, nature of the information brought by the residual motion (scene motion) and by the dominant motion (camera motion), two different probabilistic models are defined.

3.1 Residual Motion

We first describe the probabilistic model of scene motion derived from statistics on the local residual motion measurements expressed by relation (2). The histograms of these measurements computed over different video segments were found to be similar to a zero-mean Gaussian distribution (a truncated version since we deal with positive values only, by definition $v_{res}(p, t) \geq 0$) except a usually prominent peak at zero. Therefore, we model the distribution of the local residual motion measurements within a video segment by a specific mixture model involving a truncated Gaussian distribution and a Dirac distribution. It can be written as:

$$f_{v_{res}}(\gamma) = \beta\delta_0(\gamma) + (1 - \beta)\phi_t(\gamma; 0, \sigma^2)\mathbf{I}_{\gamma \neq 0}(\gamma), \quad (3)$$

where β is the mixture weight, δ_0 denotes the Dirac function at 0 ($\delta_0(\gamma) = 1$ if $\gamma = 0$ and $\delta_0(\gamma) = 0$ otherwise) and $\phi_t(\gamma; 0, \sigma^2)$ denotes the truncated Gaussian density function with mean 0 and variance σ^2 . \mathbf{I} denotes the indicator function ($\mathbf{I}_{\gamma \neq 0} = 1$ if $\gamma \neq 0$ and $\mathbf{I}_{\gamma \neq 0} = 0$ otherwise). Parameters β and σ^2 are estimated using the Maximum Likelihood criterion (ML). In order to capture not only the instantaneous motion information but also its temporal evolution over the video segment, the temporal contrasts Δv_{res} of the local residual motion measurements are also considered: $\Delta v_{res}(p, t) = v_{res}(p, t + 1) - v_{res}(p, t)$. They are also modeled by a mixture model of a Dirac function at 0 and a zero-mean Gaussian distribution, but the Gaussian distribution is not truncated here. The mixture weight and the variance of the Gaussian distribution are again evaluated using the ML criterion.

The full probabilistic residual motion model is then defined as the product of these two models as follows: $P_{M_{res}}(v_{res}, \Delta v_{res}) = P(v_{res}).P(\Delta v_{res})$. The probabilistic residual motion model is completely specified by four parameters only which are moreover easily computable. Obviously, this model does not allow us to capture how the motion information is spatially distributed in the image plane, but this is not necessary for the objective we consider here.

3.2 Dominant Image Motion

We have to design a probabilistic model of the camera motion to combine it with the probabilistic model of the residual motion in the recognition process. A first choice could be to characterize the camera motion by the motion parameter vector θ defined in Section 2 and to represent its distribution over the video segment by a probabilistic model. However, if the distribution of the two translation parameters a_1 and a_4 could be easily inferred (these two parameters are likely to be constant within a video segment so that a Gaussian mixture could reasonably be used, the task becomes more difficult when dealing with the other parameters which may be not constant anymore over a segment).

We propose instead to consider another mathematical representation of the estimated motion models, that is the camera-motion flow vectors and to consider the 2D histogram of these vectors. At each time t , the motion parameters θ_t of

the camera motion model (1) are available and the vectors $\mathbf{w}_{\hat{\theta}_t}(p)$ can be computed at any point p of the image plane (we consider the points of the image grid in practice). The values of the horizontal and vertical components of $\mathbf{w}_{\hat{\theta}_t}(p)$ are then finely quantized, and we form the empirical 2D histogram of their distribution over the considered video segment. Finally, this histogram is represented by a mixture model of 2D Gaussian distributions. Let us point out that this modelling can involve several different global motions for events of the same type filmed in different ways. The number of components of the mixture is determined with the Integrated Completed Likelihood criterion (ICL, [1]) and the mixture model parameters are estimated using the Expectation-Maximisation (EM) algorithm.

4 Event Detection Algorithm

We now exploit the designed probabilistic models of motion content for the task of event detection in video. We have to learn the concepts of dynamic content to be involved in the event detection task.

We suppose that the videos to be processed are segmented into homogeneous temporal units. This preliminary step is out of the scope of this paper which focuses on the motion modelling, learning and recognition issues. To segment the video, we can use either a shot change detection technique or a motion-based temporal video segmentation method. Let $\{s_i\}_{i=1,\dots,N}$ be the partition of the processed video into homogeneous temporal segments.

4.1 Selecting Video Segments

The first step of our event detection method permits to sort the video segments in two groups, the first group contains the segments likely to contain the relevant events, the second one is formed by the video segments to be definitely discarded. Typically, if we consider sport videos, we try to first distinguish between “play” and “no play” segments. This step is based only on the residual motion which accounts for the scene motion, therefore only single-variable probabilistic models are used, which saves computation. To this end, several motion models are learnt off-line in a training stage for each of the two groups of segments. This will be detailed in Section 5. We denote by $\{\mathcal{M}_{res}^{1,n}, 1 \leq n \leq N_1\}$ the residual motion models learnt for the “play” group and by $\{\mathcal{M}_{res}^{2,n}, 1 \leq n \leq N_2\}$ the residual motion models learnt for the “no play” group. Then, the sorting consists in assigning the label ζ_i , whose value can be 1 for “play” or 2 for “no play”, to each segment s_i of the processed video using the ML criterion defined as follows:

$$\zeta_i = \arg \max_{k=1,2} \left[\max_{1 \leq n \leq N_k} P_{\mathcal{M}_{res}^{k,n}}(z_i) \right] \quad (4)$$

$z_i = \{v_{res,i}, \Delta v_{res,i}\}$ denote the local residual motion measurements and their temporal contrasts for the video segment s_i .

4.2 Detecting Relevant Events

Problem statement. The second step of the proposed method effectively deals with the detection of the events of interest within the previously selected segments. Contrary to the first step, the two kinds of motion information (scene motion and camera motion) are exploited, since their combination permits to more precisely characterize a specific event. For a given genre of video document, an off-line training stage is required to learn the dynamic content concepts involved in the event detection task. As explained in Section 5, a residual motion model \mathcal{M}_{res}^j and a camera motion model \mathcal{M}_{cam}^j have to be estimated from a given training set of video samples, for each event j to be retrieved. The detection is performed in two sub-steps. First, we assign to each pre-selected segment the label of one of the event classes introduced in the considered task. This issue is stated as a classification problem which avoids the need of detection thresholds and allows all the considered events to be extracted in a single process. Since false segments might be included in the pre-selected segments, a validation step is subsequently applied to confirm or not the assigned labels.

Video segment labeling. We consider only the segments s_i which have been selected as “play” segments after the first step described above. For each video segment s_i , $z_i = \{v_{res\ i}, \Delta v_{res\ i}\}$ are the residual motion measurements and their temporal contrasts, and w_i represent the motion vectors corresponding to the 2D affine motion models estimated between successive images over the video segment s_i .

The video segments are then labeled with one of the J learnt classes of dynamic events according to the ML criterion. More precisely, the label l_i assigned to the segment s_i takes its value in the label set $\{1, \dots, J\}$ and is defined as follows :

$$l_i = \arg \max_{j=1, \dots, J} P_{\mathcal{M}_{res}^j}(z_i) \times P_{\mathcal{M}_{cam}^j}(w_i) \quad (5)$$

Prior on the classes could be introduced in (5) leading to a MAP criterion.

Event label validation. By applying (5), we can label all the segments supplied by the first selection step. However, we have to consider that there might be “no play” segments wrongly labeled as “play” after the first selection step. We call them “intruders”. These segments are forced to be assigned one of the event classes using relation (5), which creates false detection. As a consequence, we propose a validation test, involving only residual motion models. It consists in testing for each segment s_i the hypotheses defined by:

$$\begin{cases} H_0 : "s_i \text{ really belongs to the class } l_i \text{ determined by (5)}" \\ H_1 : "s_i \text{ is labeled as } l_i, \text{ whereas it is an intruder segment}" \end{cases}$$

To this end, a set of models $\overline{\mathcal{M}}_{res}^j$ has to be specified and estimated to represent the intruder segments. This will be explained in Section 5.

The likelihood test to choose between this two hypotheses, is given by:

if $\frac{P_{\mathcal{M}_{res}^j}(z_i)}{P_{\bar{\mathcal{M}}_{res}^j}(z_i)} < \varepsilon$, H_1 is accepted ; else, H_0 is accepted.

In this way, we can correct some misclassifications resulting from the imperfect output of the first selection step, by discarding the video segments which are rejected by the likelihood test.

5 Learning the Dynamic Content Concepts

For a given video genre, a training step is performed off-line in order to learn the residual motion models and the dominant motion models needed by the event detection method. Let us note that we have to divide the training set in two sub-sets. The first one is used to learn the motion models required by steps 1 and 2 of the event detection algorithm, while the second one allows us to learn the intruder motion models.

Learning the residual motion models for the two-group selection step. As the first selection step involves the scene motion only, we have to learn residual motion models as specified in subsection 3.1. Because of the large diversity of video contents in the two groups “play” and “no play”, we have to represent each group by several motion models. We apply the ascendant hierarchical classification (AHC) technique, on one hand, to the “play” group, and on the other hand, to the “no play” group of the training set. The overall procedure is defined as follows.

Step 0: A residual motion model is estimated for each video segment belonging to the training set of the considered group. At this early stage, each segment forms a cluster. *Step 1:* The two clusters (either composed of one segment or of several segments) found as the nearest w.r.t the symmetrized Kullback-Leibler distance between their corresponding residual motion models, are merged in the same cluster. The expression of this distance between two residual motion models \mathcal{M}_{res}^1 and \mathcal{M}_{res}^2 is $d(\mathcal{M}_{res}^1, \mathcal{M}_{res}^2) = \frac{1}{2}(d_K(\mathcal{M}_{res}^1, \mathcal{M}_{res}^2) + d_K(\mathcal{M}_{res}^2, \mathcal{M}_{res}^1))$, where $d_K(\mathcal{M}_{res}^1, \mathcal{M}_{res}^2) = d_K(f_{v_{res}}^1, f_{v_{res}}^2) + d_K(f_{\Delta v_{res}}^1, f_{\Delta v_{res}}^2)$. The expression of the Kullback-Leibler distance between the density functions $f_{v_{res}}^1$ with parameters (β_1, σ_1) , and $f_{v_{res}}^2$ with parameters (β_2, σ_2) , of the residual motion measurements is given by:

$$d_K(f_{v_{res}}^1, f_{v_{res}}^2) = \beta_1 \ln \left(\frac{\beta_1}{\beta_2} \right) + (1 - \beta_1) \ln \left(\frac{\sigma_2(1 - \beta_1)}{\sigma_1(1 - \beta_2)} \right) + \frac{1 - \beta_1}{2} \left(\frac{\sigma_1^2}{\sigma_2^2} - 1 \right).$$

The Kullback-Leibler distance between the density functions $f_{\Delta v_{res}}^1$ and $f_{\Delta v_{res}}^2$ of the temporal contrasts can be similarly written. A residual motion model is then estimated for the obtained new cluster. We iterate until the stopping criterion is satisfied. *Stopping criterion:* We stop if the maximum of the symmetrized Kullback-Leibler distances between two clusters is lower than a certain percentage of the maximum of the distances computed at step 0.

At this stage, the load of manually labelling the video segments of the training set is kept low. Indeed, we just need to sort the video segments into the two groups “play” and “no play”. At the end, each group is represented by a (small) set of clusters (depending on the heterogeneity of the video segment contents of the group) and their associated residual motion models, both obtained in an automatic way.

Learning the motion models of the event classes. Camera motion models and residual motion models representing the different event classes to be recognized are required for the second step of our detection method. They are estimated from the same training set as the one used to learn residual motion models involved in the selection step. We first need a manual labelling of the “play” segments of the training set according to the events to detect. For each event class, a camera motion model is estimated from the video segments representing the considered event as explained at the end of subsection 3.2. Similarly, the four parameters of the residual motion models for each event class are estimated using the ML criterion.

Learning of intruder motion models. We have also to determine motion models, from the second subset of the training set, to represent the intruder segments. It is important to consider a different set of video segments than the one used to learn the models involved in the first steps of the detection method. The selection step is applied to the second subset of the training set. The intruder segments are then determined (since we have the ground truth on that training set) and submitted to the classification step of the method. Finally, we get a subset of intruder segments associated to each predefined event j , which allows us to estimate the associated residual motion model previously denoted by $\overline{\mathcal{M}}_{res}^j$.

6 Experimental Results

We have applied the described method on sports videos which involve complex contents while being easily specified. Moreover, events or highlights can be naturally related to motion information in that context. We report here results obtained on athletics and tennis videos.

6.1 Experimental Comparison

First, we have carried out an experimental comparison between our statistical approach and a histogram-based technique. In order to evaluate the probabilistic framework we have designed, we consider the same motion measurements for the histogram technique. Thus, the latter involves three histograms: the histogram of residual motion measurements v_{res} (2), the histogram of their temporal contrasts Δv_{res} , and the 2D histogram of the camera-motion flow vectors (subsection 3.2). Each event j is then represented by three histograms: $H_{v_{res}}^j$, $H_{\Delta v_{res}}^j$ and H_{cam}^j .

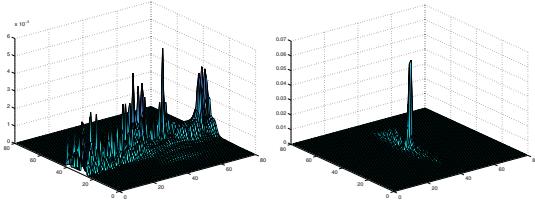


Fig. 2. Athletics video: 2D histograms of the camera-motion flow vectors. Left: for a pole vault shot, right: for a long-shot of track race.



Fig. 3. Athletics video: Detection of relevant events: Top row: ground-truth, middle row: results obtained with the probabilistic motion models, bottom row: results obtained with the histogram-based technique. From dark to light shining: pole vault, replay of pole vault, long-shot of track race and close-up of track-race

To compare two histograms, we consider the Euclidian distance, denoted by d_1 for 1D histograms and by d_2 for 2D histograms. Several distances can be considered to compare two histograms, and this issue has to be carefully addressed. However, the computed motion measurements are all real values and we have a huge number of available computed values. We can thus consider a very fine quantization and the resulting histograms are very close to the real continuous distributions. Moreover, the histogram distance is only used to rank the classes. The Euclidean distance is then a reasonable choice while easy to compute. These histograms are computed (and stored) for each event j from the training set of video samples. Then, we consider the test set and we compute the three histograms $H_{v_{res}}^{s_i}$, $H_{\Delta v_{res}}^{s_i}$ and $H_{cam}^{s_i}$, for each video segment s_i of the test set. The classification step is now formulated as assigning the label l_i of the event which minimizes the sum of the distances between histograms:

$$l_i = \arg \min_{j=1, \dots, J} \left(d_1(H_{v_{res}}^{s_i}, H_{v_{res}}^j) + d_1(H_{\Delta v_{res}}^{s_i}, H_{\Delta v_{res}}^j) + d_2(H_{cam}^{s_i}, H_{cam}^j) \right) \quad (6)$$

In order to focus on the classification performance of the two methods, the test set only involves “play” segments. We have processed a part of an athletics TV program which includes jump events and track race shots. The training set is formed by 12500 images and the test set comprises 7800 images. Some representative images of this video are displayed on Figure 1. We want to recognize four events: Pole vault, Replay of pole vault, Long-shots of track race and Close-up of track race. Consequently, we have to learn four residual motion models and four camera motion models for the method based on the probabilistic motion

modelling. Figure 2 contains the 2D histograms of the camera-motion flow vectors for two classes. In Figure 3, the processed video is represented by a time line exhibiting the duration of the video segments. The “no play” segments have been in fact withdrawn, and the “play” segments have been concatenated to form the time line. A grey level is associated to each event class. The first row corresponds to the ground truth, the second one and the third one contain the results obtained respectively using the probabilistic motion models and using the histogram technique. These results demonstrate that the statistical framework yields quite satisfactory results and outperforms the histogram-based technique.

6.2 Event Detection Method

We have applied our event detection method to a tennis TV program. The first 42 minutes (63000 images) of the video are used as the training set (22 minutes for the learning of the motion models involved in the two first steps and 20 minutes for the learning of intruder models), and the last 15 minutes (18000 images) form the test set.

Selecting video segments. We want to distinguish between “play” segments involving the two tennis players in action and the “no play” segments including views of the audience, referee shots or shots of the players resting, as illustrated in Figure 4. We only exploit the first subset of the training set to learn the residual motion models that we need for the selection step. 205 video segments of the training set represent “play” segments and 95 are “no play” segments. 31 residual motion clusters and their associated models are supplied by the AHC algorithm for the “play” group, and 9 for the “no play” group. The high number of clusters obtained reveals the diversity of dynamic contents in the two groups of the processed video. Quite satisfactory results are obtained, since the precision rate for the play group is 0.88 and the recall rate is 0.89.



Fig. 4. Tennis video: Three image samples extracted from the group of “play” segments and three image samples extracted from the group of “no play” segments.

Table 1. Tennis video: Results of the event detection method based on probabilistic motion models (P: precision, R: recall).

	Rally	Serve	Change of side
P	0.92	0.63	0.85
R	0.89	0.77	0.74

Detecting relevant events. The goal is now to detect the relevant events of the tennis video among the segments selected as “play” segments. For this second step, we introduce the probabilistic camera motion model. The three events we try to detect are the following: Rally, Serve and Change of side. In practice, we consider two sub-classes for the Serve class, which are wide-shot of serve and close-up of serve. Two sub-classes are considered too for the Change-of-side class. As a consequence, five residual motion models and five camera motion models have to be learnt. We have also to determine the residual motion models accounting for the intruder segments for each class. The obtained results are reported in Table 1. Satisfactory results are obtained specially for the rally class. The precision of the serve class is lower than the others. In fact, for the serve class, errors come from the selection step (i.e., some serve segments are wrongly put in the “no play” group, and then, are lost). It appears that a few serve segments are difficult to distinguish from some “no play” segments when using only motion information. However, the proposed statistical framework can easily integrate other information such as color or audio.

7 Conclusion

We have addressed the issue of determining dynamic content concepts from low-level video features with the view to detecting meaningful events in video. We have focused on motion information and designed an original and efficient statistical method. We have introduced new probabilistic motion models representing the scene motion and the camera motion. They can be easily computed from the image sequence and can handle a large variety of dynamic video contents. We have demonstrated that the considered statistical framework outperforms a histogram-based technique. Moreover, it is flexible enough to properly introduce prior on the classes if available, or to incorporate other kinds of video primitives (such as color or audio). The proposed two-step method for event detection is general and does not exploit very specific knowledge (e.g. related to the type of sport) and dedicated solutions. Satisfactory results on sports videos have been reported.

Acknowledgments. This research was supported by “Région Bretagne” (PhD thesis grant) and by the French Ministry of Industry (RNTL Domus Videum project). The authors would like to thank INA, Direction de la Recherche, for providing the videos.

References

1. C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the Integrated Completed Likelihood. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(3):719–725, 2000.
2. A. Divakaran, R. Radhakrishnan, and K.A. Peker. Motion activity-based extraction of key-frame from video shots. *ICIP’02*, Rochester, Sept. 2002.

3. A. Ekin, A.M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Int. Trans. on Image Processing*, 12(7):796–807, July 2003.
4. R. Fablet, P. Bouthemy, and P. Pérez. Non-parametric motion characterization using causal probabilistic models for video indexing and retrieval. *IEEE Trans. on Image Processing*, 11(4):393–407, 2002.
5. J. Li and J.Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on PAMI*, 25(9):1075–1088, Sept. 2003.
6. C-W. Ngo, T-C. Pong, and H-J. Zhang. On clustering and retrieval of video shots through temporal slices analysis. *IEEE Trans. Multimedia*, 4(4):446–458, Dec. 2002.
7. J-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. of Visual Comm. and Image Repr.*, 6(4):348–365, Dec. 1995.
8. Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. *CVPR’2000*, Hilton Head, SC, 2000.
9. N. Vasconcelos and A. Lippman. Statistical models of video structure for content analysis and characterization. *IEEE Trans. on IP*, 9(1):3–19, Jan. 2000.
10. Y. Yacoob and J. Black. Parametrized modeling and recognition of activities. *Sixth IEEE Int. Conf. on Computer Vision*, Bombay, India, 1998.
11. L. Zelnik-Manor and M. Irani. Event-based video analysis. *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, Dec. 2001.

Are Iterations and Curvature Useful for Tensor Voting?[☆]

Sylvain Fischer¹, Pierre Bayerl², Heiko Neumann², Gabriel Cristóbal¹, and Rafael Redondo¹

¹ Instituto de Optica (CSIC), Serrano 121, 28006 Madrid, Spain
{sylvain,gabriel,rafa}@optica.csic.es

² Universität Ulm, Abt. Neuroinformatik, D-89069 Ulm, Germany
{pierre,hneumann}@neuro.informatik.uni-ulm.de

Abstract. Tensor voting is an efficient algorithm for perceptual grouping and feature extraction, particularly for contour extraction. In this paper two studies on tensor voting are presented. First the use of iterations is investigated, and second, a new method for integrating curvature information is evaluated. In opposition to other grouping methods, tensor voting claims the advantage to be non-iterative. Although non-iterative tensor voting methods provide good results in many cases, the algorithm can be iterated to deal with more complex data configurations. The experiments conducted demonstrate that iterations substantially improve the process of feature extraction and help to overcome limitations of the original algorithm. As a further contribution we propose a curvature improvement for tensor voting. On the contrary to the curvature-augmented tensor voting proposed by Tang and Medioni, our method takes advantage of the curvature calculation already performed by the classical tensor voting and evaluates the full curvature, sign and amplitude. Some new curvature-modified voting fields are also proposed. Results show a lower degree of artifacts, smoother curves, a high tolerance to scale parameter changes and also more noise-robustness.

1 Introduction

Medioni and coworkers developed tensor voting as an efficient method for contour extraction and grouping. The method, supported by the Gestalt psychology, is based on tensor representation of image features and non-linear voting, as described in [2]. See also [9] for a comparison with other existing methods. Tensor voting is a non-iterative procedure, in the sense that the original scheme implements only 2 steps of voting, claiming that no more iterations are needed. In opposition, other methods for perceptual grouping [4,3,1] refine the results

* This research is supported in part by the German-Spanish Academic Research Collaboration Program HA 2001-0087 (DAAD, Acciones integradas Hispano-Alemanas 2002/2003), the projects TIC2001-3697-C03-02 from MCYT and IM3 from ISCIII and HGGM grants. S.F. and R.R. are supported by a MECD-FPU and a CSIC-I3P fellowships, respectively.

by iterative feedforward-feedback loops. Therefore, the aim of this study is to investigate how an incremented number of iterations can improve the results of tensor voting. Some basic examples are analyzed and an extraction quality measurement is proposed. The later allows to perform a statistical study on the influence of iterations in a simple case.

A curvature improvement has been proposed by Tang and Medioni [7]. They compute the sign of curvature and use it for modifying the voting fields. We propose a more sophisticated calculation of the curvature information with a low computational cost. Instead of the sign of curvature, the proposed method evaluates the full curvature using part of the calculations previously performed by the tensor voting. We adopt a curvature compatibility approach that was described by Parent and Zucker [6]. A statistical evaluation is presented and the methods are finally tested with more complex data in presence of noise.

Section 2 briefly introduces the tensor voting method. Section 3 presents a study on the usefulness of iterations for tensor voting and the section 4 describes some improvements that can be achieved when both curvature information and iterations are used. Some concluding remarks are drawn in section 5.

2 A Brief Introduction to Tensor Voting

The classical algorithm will not be fully described in detail here and only a brief description is presented in order to stress the new contributions of this paper. For a more in depth study the reader can refer to [2,7]. Also, it is necessary to remark that the present work is only restricted to still 2D images, but it could be extended to N-dimensional features, like volumetric data or motion [8,5].

A local description of the curves at each point of the image can be encoded by a symmetric positive 2x2 tensor. Tensors can be diagonalized, their eigenvalues are denoted λ_1, λ_2 with $\lambda_1 \geq \lambda_2 \geq 0$ and corresponding eigenvectors are denoted by e_1, e_2 . Tensors can be decomposed as follows:

$$T = (\lambda_1 - \lambda_2)e_1e_1^T + \lambda_2I \quad (1)$$

where I is the identity matrix. The first term is called the *stick component*, where e_1 is an evaluation of the tangent to the curve. The *stick saliency* $\lambda_1 - \lambda_2$ gives a confidence measure for the presence of a curve. The second term is called the *ball component*, and its saliency λ_2 gives a confidence measure to have a junction.

The classical tensor voting algorithm performs two voting steps in which each tensor propagates to its neighborhood. Stick tensors propagate mostly in the direction of e_1 . The region of propagation is defined by the stick voting field which decay in function of the distance and curvature (see Eq. 3 and Fig. 2.h). Ball tensors propagate in all directions and decay with the distance. After all tensors are propagated, all contributions are summed up to define new tensors that will be used for the next step. That summation can be considered as an averaging or a “voting”. The first voting step is referred as “sparse vote” because the vote is performed only on points where tensors are not null. The last voting step is called “dense vote” because the vote is accomplished on every point. After

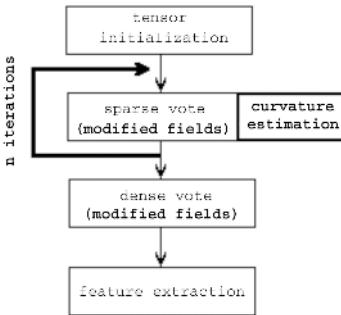


Fig. 1. Classical tensor voting consists of four steps. (1) Tensor initialization, (2) sparse voting, (3) dense voting, and (4) feature extraction. The new contributions are depicted with boldface characters, which describe iterations of the sparse voting process and curvature calculations during the sparse vote stage, modifying the voting fields by incorporating the calculated curvature.

all the voting steps are completed, curves are extracted as local maximum of stick saliency along the normal direction to stick components. Note that thresholds are necessary to eliminate low-saliency local maxima. These thresholds are held constant for each of the following experiments. Fig. 1 summarizes the different steps of the algorithm showing with boldface characters the new contributions proposed: an iterative sparse voting mechanism and a curvature calculation for modifying the voting fields.

3 Iterated Tensor Voting

3.1 Example

Tensor voting is a very efficient technique for grouping data-points that are separated by almost the same distance. A free parameter σ (the scale factor, see Eq. 3) has to be adjusted to the inter-distance between points. If σ is miss-adjusted, performance results strongly decrease: if σ is too small points will not be grouped, if σ is too big the grouping is less selective.

Fig. 2.a shows a simple example with two sets of points: first a three by three array of points separated by 11 pixels vertically and 13 pixels horizontally. Because the vertical distance is smaller, following Gestalt psychology rules, these points have to be grouped vertically. Secondly, a set of three points aligned in diagonal and separated by 42 pixel gaps. Because the gaps are different in both sets of points it is not possible to adjust σ for extracting both structures correctly. As it is shown in Fig. 2.c, if σ is small, i.e. around 5, only the vertical array is grouped. If σ is bigger than 15, only the diagonal line is correctly grouped (Fig. 2.i). Between these values (from $\sigma=7$ to 13, Fig. 2.e and g) none of these sets of points are accurately grouped.

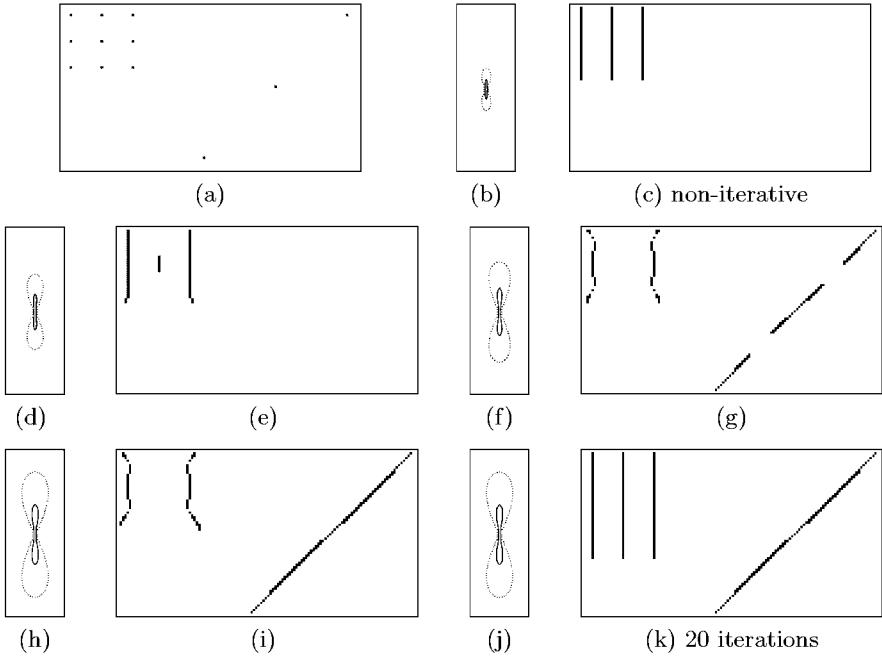


Fig. 2. Example showing the tensor voting results for different values of the scale factor σ and of the number of iterations. **a.** Data points belong to two sets: three points aligned in diagonal and an array which has to be grouped in vertically lines. **b.** Contours of the voting field for $\sigma = 5$ are drawn at 50% (solid line) and 5% (dash-dot line) of the maximum value (see Eq. (3) for voting fields description). **c.** Extraction results with the classical tensor voting algorithm (two voting steps) and $\sigma = 5$: array structure is accurately extracted, but the scale factor is too small to group diagonal points. **d. and f.** Voting fields with $\sigma = 9$ and $\sigma = 12$, respectively. **e. and g.** Contours extracted respectively with $\sigma = 9$ and $\sigma = 12$, by the non-iterative tensor voting. In both cases algorithm fails to find both array and diagonal points structures. The scale factor σ is too big for the array and too small for the diagonal points. **h. and j.** Voting field with $\sigma = 15$. **i.** With non-iterative tensor voting and $\sigma = 15$, diagonal points are correctly grouped, but not array points. **k.** With $\sigma = 15$ and 20 iterations the structure is accurately extracted, both array and diagonal line are correctly grouped.

Iterations are implemented on the sparse voting stage. For n iterations, $n - 1$ sparse votes and one dense vote are required, as shown Fig. 1. An increased number of iterations can refine the results until the correct structure is extracted. Fig. 2.k shows the results with $\sigma = 15$ and 20 iterations. Both array and diagonal line structures are now simultaneously extracted what non-iterative algorithm was not able to do. Note that a normalization stage is applied after each iteration to keep the sum of tensor eigenvalues constant.

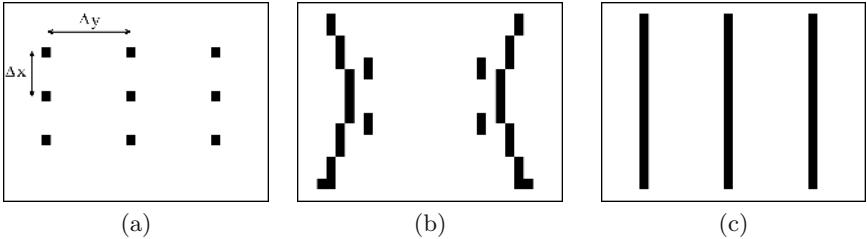


Fig. 3. Systematic evaluation of the influence of iterations using a three by three array of points. **a.** Original array of points separated by Δx , Δy (parameters are the same in all insets: $\Delta x = 4$, $\Delta y = 9$ and $\sigma = 10$). **b.** Contour extraction with the non-iterative tensor voting fails: central points are not grouped, lateral points are grouped but not in strictly vertical lines, moreover there are some artifacts ($Q=0.40$). **c.** The structure is well extracted after 10 iterations of voting: points are grouped in vertical lines ($Q=2.38$).

3.2 Statistics on the Influence of Iterations

A 3x3 array of points, shown in Fig. 3.a, is used to evaluate the effect of iterations on tensor voting. Vertical and horizontal distances between points are denoted Δx and Δy respectively. In the following, Δx will be chosen smaller than Δy . In such case points have to be grouped vertically (on the contrary if $\Delta x > \Delta y$ points would have to be grouped horizontally). Taking into account that points have to be grouped vertically, a measure of how good tensor orientation is can be represented by:

$$Q = -\log_{10} \left[\frac{1}{9} \sum_{i=1,\dots,9} \left(1 - \frac{T_i(1,1)}{S_i} \right) \right] \quad (2)$$

where i indexes the 9 points of the array. T_i is the tensor of the point i , S_i is the sum of eigenvalues of T_i and $T_i(1,1)$ the vertical component of the tensor T_i .

As vertical lines have to be extracted, tensors are correctly oriented if they have a form close to $T_i = S_i \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. In such case $\sum(1 - \frac{T_i(1,1)}{S_i})$ is close to zero, providing a high value for Q . Thus, Q can be considered as an extraction quality measurement for the described experiment. When $Q < 1$ tensors are miss-oriented and extraction can be considered as failed. On the contrary $Q > 2$ indicates tensors are well orientated and the structure is correctly extracted.

3.3 Results

Fig. 4 presents results for different parameters Δx , Δy and n (number of iterations). For all cases the scale factor σ is fixed to 10. Again, please note that in this study we are only considering cases where $\Delta x < \Delta y$ (which should yield vertical grouping following Gestalt rules of proximity and good continuation).

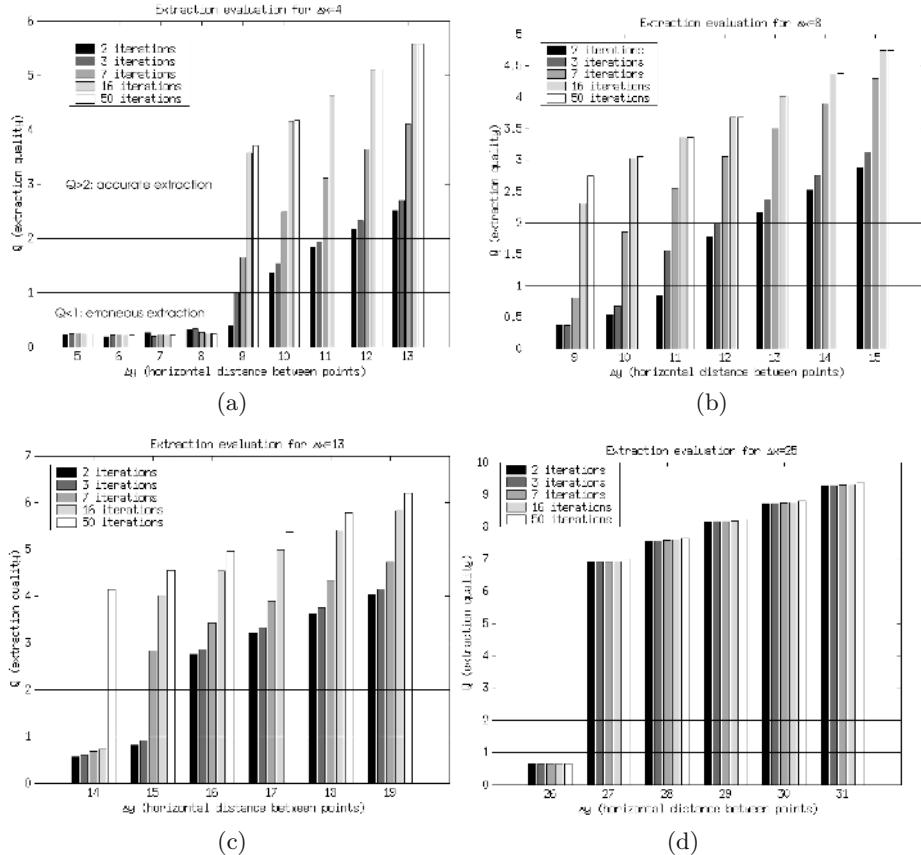


Fig. 4. Extraction quality as a function of array parameters Δx and Δy for the grid example of Fig. 3. The number of iterations n is indicated by different gray shades in the bars (two iterations bar corresponds to the classical algorithm with two voting steps). $\sigma = 10$ is held constant for the entire experiment. Only cases with $\Delta x < \Delta y$ are shown here. **a.** With a fixed $\Delta x = 4$ and $5 \leq \Delta y \leq 13$. If $\Delta y \ll \sigma$, σ is too large in comparison to the features and the extraction fails even if more iterations are deployed. If $9 \leq \Delta y \leq 11$ the structure is extracted using several iterations (results start from failed ($Q < 1$) when using the non-iterative algorithm up to accurate ($Q > 2$) when more iterations are deployed). Only if $\Delta y \geq 12$ the non-iterative algorithm is able to extract the desired information. **b.** $\Delta x = 8$ and $9 \leq \Delta y \leq 15$. **c.** $\Delta x = 13$ and $14 \leq \Delta y \leq 19$. In difficult cases like when $\Delta x \simeq \Delta y$ or $\Delta y \simeq \sigma$ several iterations are needed for extracting accurately the structure. **d.** $\Delta x = 25$ and $26 \leq \Delta y \leq 31$. Although σ is too small in comparison to the features, an accurate extraction is obtained due to the infinite Gaussian extension of the propagation fields.

Extraction is accurate for any number of iterations if σ corresponds to the optimal scale for the investigated stimuli and if there is no competition between vertical and horizontal grouping, that is, if $\sigma \ll \Delta y$ and $\Delta x \ll \Delta y$ (see Fig. 4.a,b,c,d at their rightmost parts).

If $\Delta y \ll \sigma$ it is impossible to extract the structure even if more iterations are deployed (see Fig. 4.a left part), the scale factor is indeed too large to be selective enough.

If $\Delta y \simeq \sigma$ the application of the classical algorithm fails to extract the curves. On the contrary, iterations allow tensor voting obtaining the correct structure as it can be observed in Fig. 4.a center part and Fig. 4.b left part. A similar situation is observed if $\Delta x \simeq \Delta y$ and $\Delta x, \Delta y$ are not much bigger than σ . Iterated tensor voting allow to extract the structure where the classical algorithm fails (see Fig. 4.c left part).

In conclusion, only if the features to be extracted are simple and they do not appear in competition, the non-iterative algorithm would suffice for correctly extracting image features. For more complicated cases, when some competition between orientations is present or when the scale factor σ is not precisely adjusted, more than two iterations are required. Moreover, it has been seen that in almost all cases iterations do not impair the quality of the results and on the contrary they allow to refine the final structures. In all, the use of iterations can help to overcome the limitations of the non-iterative method, improving the feature extraction results.

4 Curvature Improvement

4.1 Method

The proposed curvature improvement introduces a curvature calculation and modified stick voting fields. The curvature is evaluated in each voter point by averaging over all receiver points the curvature calculation ρ already computed in the classical tensor voting. In the classical tensor voting, a voter A votes on a receiver B with an amplitude described by the stick voting field equation:

$$V(A, B) = \exp\left(-\frac{s(A, B)^2 + c \rho(A, B)^2}{\sigma^2}\right) \quad (3)$$

with

$$\rho(A, B) = \frac{2 \sin \theta}{d} \quad (4)$$

where $s(A, B)$ and $\rho(A, B)$ are respectively the length and the curvature of the circular arc which is tangent to $\vec{e}_1(A)$ in point A and goes through point B (see Fig. 5.a). d is the Euclidean distance between both points A and B, θ is the angle between vectors $\vec{e}_1(A)$ and \vec{AB} . σ -the scale factor- and c are constants. Fig. 2.b,d,f,h shows the contours of such voting fields for different values of σ .

The curvature will be evaluated in each voter point A. To permit inflexion points and changes of curvature, the curvature is calculated separately in both half planes P_+ and P_- defined respectively by $P_+ = \{B, (\vec{e}_1(A), \vec{AB}) > 0\}$ and $P_- = \{B, (\vec{e}_1(A), \vec{AB}) < 0\}$. The weighted average over each half plane gives $\gamma_i(A)$ (where $i = +$ or $-$), which is a curvature evaluation at the point A:

$$\gamma_i(A) = \frac{\sum_{B \in P_i} (\lambda_1(B) - \lambda_2(B)) V(A, B) \rho(A, B)}{\sum_{B \in P_i} (\lambda_1(B) - \lambda_2(B)) V(A, B)} \quad (5)$$

where $\lambda_1(B), \lambda_2(B)$ are the eigenvalues of the tensor B . The weighted average is very similar to the “voting” used in tensor voting: the same weighting functions composed by the voting fields V and the stick saliency $\lambda_1 - \lambda_2$ are used.

The γ_i determined at one iteration, can then be used in the next iteration for modifying the stick voting fields. The following equation extends Eq. 3:

$$V(A, B) = \exp\left(-\frac{s(A, B)^2 + c(\rho(A, B) - \gamma_i(A))^2}{\sigma^2}\right) \text{ for any } B \in P_i \quad (6)$$

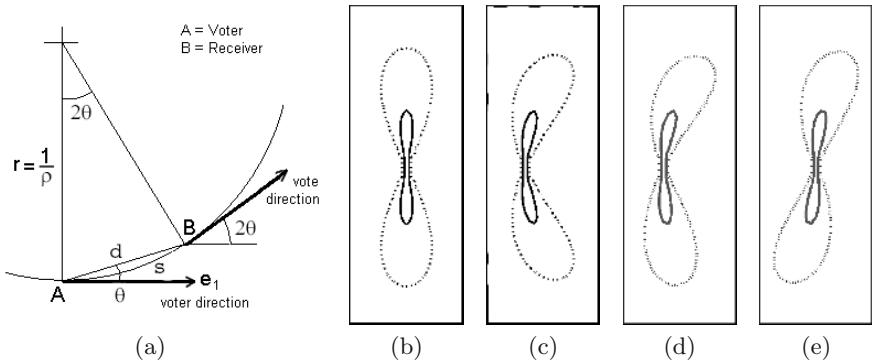


Fig. 5. a. Tensor voting fields are build calculating the distance d , the angle θ , the arc longitude s and the curvature ρ between the voter A oriented by its first eigenvector e_1 and the receiver B . In the curvature improvement the curvature is evaluated in the voter A by averaging ρ over all receivers. **b.** Classical voting field without curvature. Contours are drawn at 50% and 5% of the maximum value, $\sigma = 15$ for all voting fields of the figure. **c.** Symmetric curved voting field with curvatures $\gamma^+ = \gamma^- = .06$. **d.** Curved voting field with different curvatures in both half planes, $\gamma^+ = .09$ and $\gamma^- = .03$. **e.** Curved voting field with inflexion, $\gamma^+ = .06$ and $\gamma^- = -.06$.

Some examples of such curvature-modified voting fields are shown Fig. 5.c,d,e. See in comparison the former contours Fig. 5.b. In points where the ball component has a significant level in comparison to the stick component, curvatures have to be considered as zero because no reliable curvature calculation is possible if curve orientation is itself not reliable. Therefore curved voting fields are employed only where tensor orientation has high confidence (the curved voting fields are only used under the condition $\frac{\lambda_1}{\lambda_2} > 10$).

Remarkably the method follows the “voting” methodology. Curvature is found by averaging. Moreover it uses the same voting fields V as tensor voting. It can then be hoped to conserve the good properties of the tensor voting, like the robustness to noise. The curvature improvement does not entail an important additional computational cost in comparison to the classical method,

while it uses the same kind of operations as the tensor voting and reuses calculations already done, i.e. in the curvature calculation of Eq. 5 all variables λ_1 , λ_2 , V and ρ are already computed by the classical tensor voting.

Note also that an increased number of iterations is necessary to refine the results. The number of iterations can be considered as an additional parameter of the algorithm. A procedure could also be implemented for stopping the iterations when the results do not change much from one iteration to the following one. For all examples presented here a fixed number of iterations is used. 10 iterations have been seen to be sufficient unless data structure presents some special ambiguity.

In the following, the curvature improvement will be compared with the non-iterative tensor voting and iterative tensor voting without curvature improvement. Results need to be compared with Tang and Medioni's method taking into account the sign of curvature [7], although this was out of the scope of the present study.

4.2 Statistical Study

Fig. 6.a shows an image composed by sparse points located on the edges of an ellipse. The distance between points vary between 6 to 12 pixels. This example is used for comparing the three versions of the algorithm. For different values of the scale factor σ , we count the number of points erroneously extracted outside the ellipse contour, tolerating a deviation of two pixels around the ideal ellipse. Results are presented in Fig. 6.b-e.

All versions of the algorithm require a σ value to be higher than a minimum value ($\sigma \geq 7$ in the present case) for extracting the contour of the ellipse. With a smaller value of σ , points are not grouped together. On the other hand, σ needs to be small for avoiding artifacts, i.e. the number of misplaced points increases strongly for tensor voting without curvature information for $\sigma > 10$, and for $\sigma > 34$ if the curvature improvement is considered. Classical tensor voting adequately extracts the contours, although with artifacts, for σ between 7 and 10. Iterations have few influence on the results. In comparison curvature improvement extracts adequately the ellipse over a large range of σ values, i.e. between 7 to 34. Moreover it does not produce any artifacts for σ between 7 and 21 and yields smoother slopes.

4.3 Hand-Written Text Example

Fig. 7 shows another example of contour extraction with the three versions of the algorithm: non-iterative, iterative with 10 iterations and iterative with the curvature improvement (with also 10 iterations). The first image "Cyan" (Fig. 7.a) is composed of sparse points along handwritten characters. The second one (Fig. 7.b) is the same image "Cyan" with 20% of noise (i.e. every fifth data point is noise). Same parameters are used for each method. After tensor voting is applied the contours of the letters are extracted. Results show tensor voting with 10 iterations (Fig. 7.e) reduces the artifacts and closes the curves better

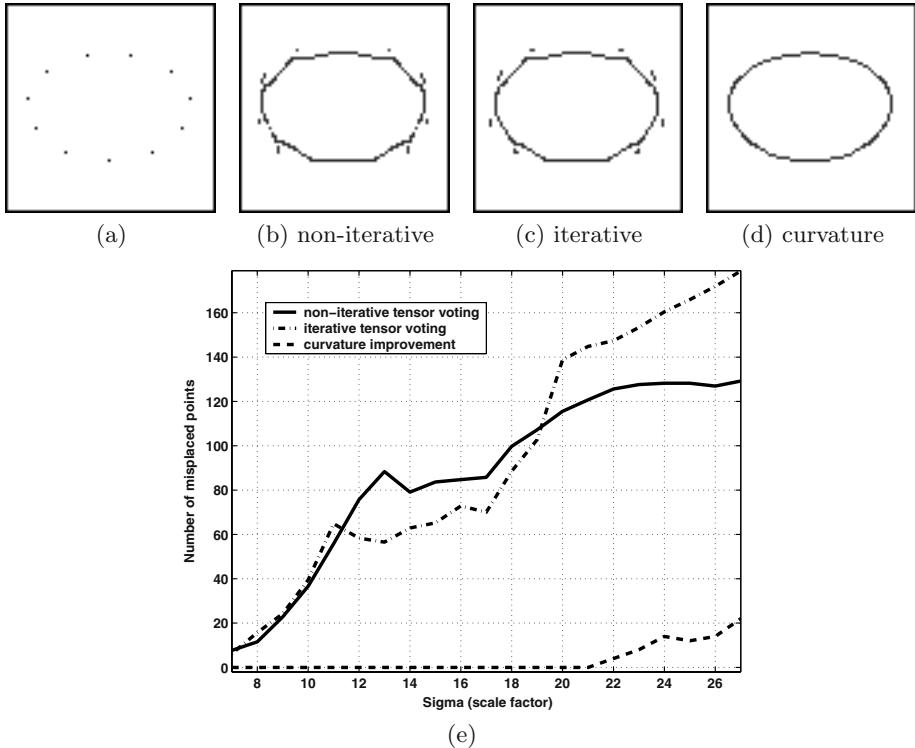


Fig. 6. Comparison between the three methods. **a.** The input image is composed by a sparse set of dots dispersed along the edges of an ellipse. In insets a., b. and c. all parameters are the same and $\sigma = 8$. **b. and c.** Extraction results with, respectively, the non-iterative algorithm and 10 iterations of tensor voting. The ellipse is adequately extracted but artifacts can be observed, moreover slopes are not smooth. Both methods provide similar results. **d.** With the curvature improvement and 10 iterations, the ellipse is extracted without artifacts and with smooth curves. **e.** Results for σ varying between 7 and 27 are presented. The number of points erroneously extracted, that is extracted out of the ellipse are plotted for each method. Tensor voting without curvature information extract the ellipse, although always with artifacts, for σ between 7 and 10. Curvature improvement extracts it without artifacts and tolerates a larger range of σ (from 7 to 21).

than non-iterative tensor voting (Fig. 7.c). With the curvature improvement (Fig. 7.g) extracted contours of the curves have even less artifacts and are much smoother. Comparison of the results with the noisy image (Fig. 7.d,f,h) shows that curvature improvement does not impair the quality but even improves it, e.g. contour continuity is better preserved.

For regions with straight segments and junctions both curvature improvement and iterative tensor voting behaves similarly. Therefore, curvature improvement does not impair the results for such situations. As a consequence the curvature

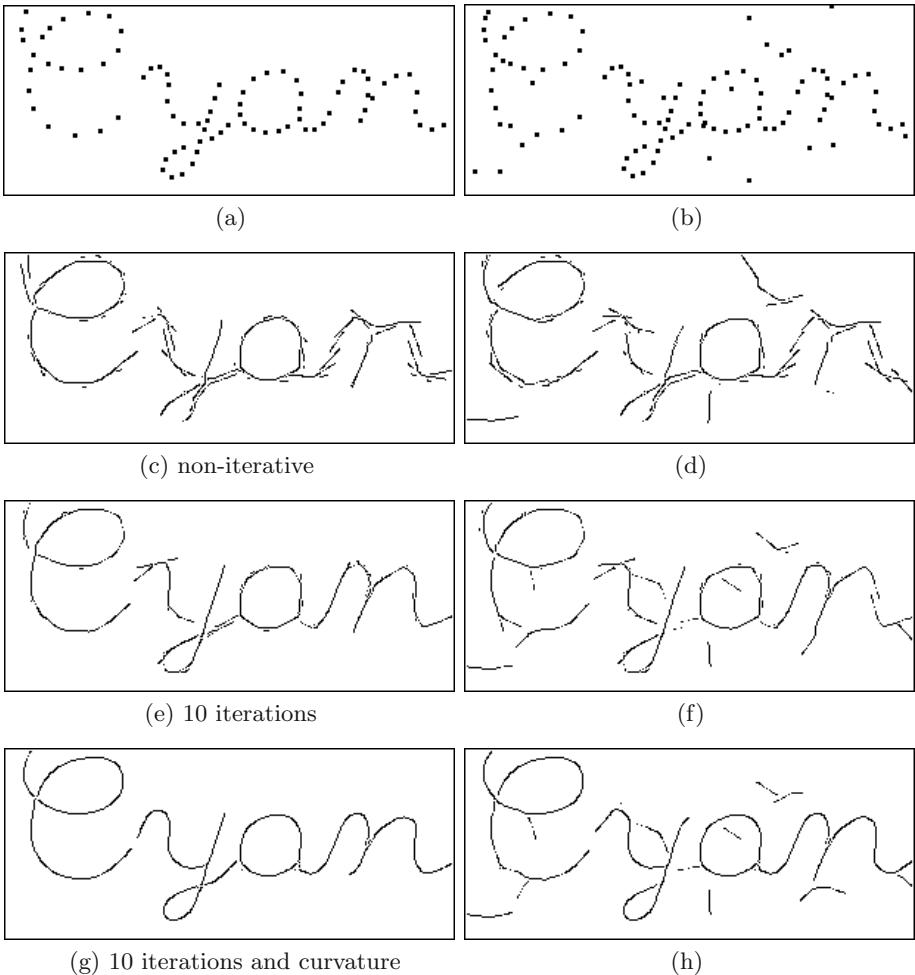


Fig. 7. A hand written example. **a.** The test image “Cyan” is a 128x304 pixel image composed by points dispersed along hand-written letters. For better visualization points are magnified. **b.** The second test image is the same image “Cyan” with 20% noise. Parameters are the same for all experiments ($\sigma = 15$). **c and d.** Extraction results of respectively the image “Cyan” and the noisy image version with non-iterative tensor voting. In both cases the algorithm fails to close the curves and yields high level of artifacts. **e and f.** Extraction results of “Cyan” images with 10 iterations. Curves are better closed and the level of artifacts is lower than with non-iterative tensor voting. **g and h.** Extraction results with the curvature improvement and 10 iterations. The text is accurately extracted, with less artifacts and smoother slopes. Results resist slightly better to noise than without curvature improvement. It is remarkable that the curve continuity of the letters C, Y and N is preserved.

improvement can be used for any kind of images. Remarkably, curvature improvement accurately extracts the structure of the example Fig. 2.a using the same parameters ($\sigma = 15$ and 20 iterations).

5 Conclusion

This paper demonstrated that iterations are useful for tensor voting, particularly for extracting correct contours in difficult situations like feature competition or scale parameter misadjustment. In almost all cases iterations do not impair the quality of the results and on the contrary they allow refining and improving the final structures. The curvature improvement provides better results for curved features as it reduces the level of artifacts and smoothes curves, besides the fact that it also increases the robustness of the method to scale parameter misadjustment and noise.

References

1. Hansen, T., Sepp, W., Neumann, H.: Recurrent long-range interactions in early vision. S. Wermter et al. (Eds.): Emergent neural computational architectures, LNAI 2036 (2001) 127–138
2. Medioni, G., Lee, M.-S., Tang, C.-K.: A computational framework for feature extraction and segmentation. Elsevier Science (mar. 2000)
3. Mingolla, E., Ross, W., Grossberg, S.: A neural network for enhancing boundaries and surfaces in synthetic aperture radar images. *Neural Networks* **12** (1999) 499–511
4. Neumann, H. and Mingolla, E.: Computational neural models of spatial integration in perceptual grouping. T.F. Shipley and P.J. Kellman, editors, *From Fragments to Objects - Segmentation and Grouping in Vision*. Elsevier Science (2001) 353–400
5. Nicolescu, M. and Medioni G., Layered 4D Representation and Voting for Grouping from Motion. *IEEE Trans. P.A.M.I.* **25(4)** (2003) 492–501
6. Parent, P. and Zucker, S.: Trace inference, curvature consistency, and curve detection. *IEEE Trans, P.A.M.I.* **11** (1989) 823–839
7. Tang, C.-K. and Medioni, G.: Curvature-Augmented Tensor Voting for Shape Inference from Noisy 3D Data. *IEEE Trans. P.A.M.I.* **24(6)** (June 2002) 868–864
8. Tang, C.-K., Medioni, G., Lee, M.: N-dimensional tensor voting and application to epipolar geometry estimation. *IEEE Trans P.A.M.I.* **23(8)** (2001) 829–844
9. Williams, L.R. and Thornber, K.K.: A comparison of measures for detecting natural shapes in cluttered backgrounds. *Int. Jour. of Computer Vision* **34(2/3)**(2000)81–96

A Feature-Based Approach for Determining Dense Long Range Correspondences

Josh Wills and Serge Belongie

University of California, San Diego
La Jolla, CA 92093 USA
{josh,sjb}@cs.ucsd.edu
<http://vision.ucsd.edu>

Abstract. Planar motion models can provide gross motion estimation and good segmentation for image pairs with large inter-frame disparity. However, as the disparity becomes larger, the resulting dense correspondences will become increasingly inaccurate for everything but purely planar objects. Flexible motion models, on the other hand, tend to overfit and thus make partitioning difficult. For this reason, to achieve dense optical flow for image sequences with large inter-frame disparity, we propose a two stage process in which a planar model is used to get an approximation for the segmentation and the gross motion, and then a spline is used to refine the fit. We present experimental results for dense optical flow estimation on image pairs with large inter-frame disparity that are beyond the scope of existing approaches.

1 Introduction

Layer-based motion segmentation based on differential optical flow [18,19] can provide good estimation of both the coherent groups in image sequence as well as the associated motion of each group. However, that work is only applicable to scenes where the inter-frame disparity is small. There are two major problems that arise as the disparity increases. The first is that if the disparity exceeds roughly 10-15% of the image size, then even coarse-to-fine optical flow will not be able to find the solution [7]. The second is that with large disparity the planar motion model associated with the layers (e.g. rigid, affine) likely becomes inaccurate for everything but purely planar objects.

In this paper our goal is to determine dense optical flow – by optical flow we are referring to dense correspondences and not the method of differential optical flow estimation – between image pairs with large inter-frame disparity. We propose a two-stage framework based on a planar motion model for capturing the gross motion followed by a regularized spline model for capturing finer scale variations. Our approach is related in spirit to the *deformation* concept in [12], developed for the case of differential motion, which separates overall motion (a finite dimensional group action) from the more general deformation (a diffeomorphism).

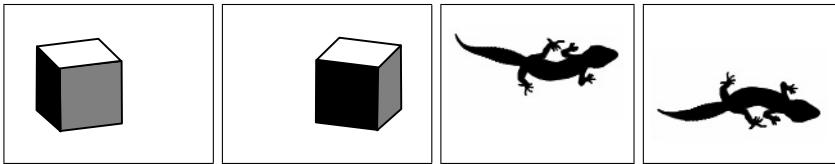


Fig. 1. Non-planarity vs. non-rigidity: The left image pair shows a non-planar object undergoing 3D rigid motion; the right pair shows an approximately planar object undergoing non-rigid motion. Both examples result in residual with respect to a 2D planar fit.

The types of image pairs that we wish to consider are illustrated in Figure 1. These have a significant component of planar motion but exhibit residual with respect to a planar fit because of either the non-planarity of the object (e.g. a cube) or the non-rigidity of the motion (e.g. a lizard). These are scenes for which the motion can be approximately described by a planar layer-based framework, i.e. scenes that have “shallow structure” [10].

It is important to remember that optical flow does not model the 3D motion of objects, but rather the changes in the image that result from this motion. Without the assumption of a rigid object, it is very difficult to estimate the 3D structure and motion of an object from observed change in the image, though there is existing work that attempts to do this [5,17]. For this reason, we choose to do all estimation in the image plane (i.e. we use 2D models), but we show that if the object is assumed to be rigid, the correspondences estimated can be used to recover the dense structure and 3D motion.

This approach extends the capabilities of feature-based scene matching algorithms to include dense optical flow without the limits on allowable motion associated with techniques based on differential optical flow. Previously, feature-based approaches could handle image pairs with large disparity and multiple independently moving objects, while optical flow techniques could provide a dense set of pixel correspondences even for objects with non-rigid motion. However, neither type of approach could handle both simultaneously. Without the assumption of a rigid scene, existing feature-based methods cannot produce dense optical flow from the sparse correspondences, and in the presence of large disparity and multiple independently moving objects, differential optical flow (even coarse-to-fine) can break down. The strength of our approach is that dense optical flow can now be estimated for image pairs with large disparity, more than one independently moving object, and non-planar (including non-rigid) motion.

The structure of the paper is as follows. We will begin in Section 2 with an overview of related work. In Section 3, we detail the components of our approach. We discuss experimental results in Section 4. There is discussion in Section 5 and the paper concludes with Section 6.

2 Related Work

The work related to our approach comes from the areas of motion segmentation, optical flow and feature-based (sparse) matching. Several well known ap-

proaches to motion segmentation are based on dense optical flow estimation [1,3, 8]; in these approaches the optical flow field was assumed to be piecewise smooth to account for discontinuities due to occlusion and object boundaries. Wang & Adelson introduced the idea of decomposing the image sequence into multiple overlapping layers, where each layer represents an affine motion field [18]. However their work was based on differential optical flow, which places strict limits on the amount of motion between two frames.

In [19], Weiss uses regularized radial basis functions (RBFs) to estimate dense optical flow; Weiss' method is based on the assumption that while the motion will not be smooth across the entire image, the motion is smooth within each of the layers. Given the set of spatiotemporal derivatives, he used the EM algorithm to estimate the number of motions, the dense segmentation and the dense optical flow. This work along with other spline-based optical flow methods [13, 14] however, also assumes differential motion and therefore does not apply for the types of sequences that we are considering.

In [16], Torr et al. show that the trifocal tensor can be used to cluster groups of sparse correspondences that move coherently. This work addresses similar types of sequences to those of our work in that it is trying to capture more than simply a planar approximation of motion, but it does not provide dense assignment to motion layers or dense optical flow. The paper states that it is an initialization and that more work is needed to provide a dense segmentation, however the extension of dense stereo assignment to multiple independent motions is certainly non-trivial and there is yet to be a published solution. In addition, this approach is not applicable for objects with non-rigid motion, as the fundamental matrix and trifocal tensor apply only to rigid motion.

Our work builds on the motion segmentation found via planar motion models as in [20], where planar transformations are robustly estimated from point correspondences in a RANSAC framework. A dense assignment of pixels to transformation layers is then estimated using an MRF. We refine the planar estimation produced by [20] using a regularized spline fit. Szeliski & Shum [14] also use a spline basis for motion estimation, however their approach has the same limitations on the allowable motion as other coarse-to-fine methods.

3 Our Approach

Our goal in this paper is to determine the dense optical flow for pairs of images with large inter-frame disparity and in the presence of multiple independent motions. If the scene contains objects undergoing significant 3D motion or deformation, the optical flow cannot be described by any single low dimensional image plane transformation (e.g. an affine transformation or a homography). However, to keep the problem tractable we need a compact representation of these transformations; we propose the use of thin plate splines for this purpose. A single spline is not sufficient for representing multiple independent motions, especially when the motion vectors intersect [19]. Therefore we represent the optical flow between two frames as a set of disjoint splines. By disjoint we mean

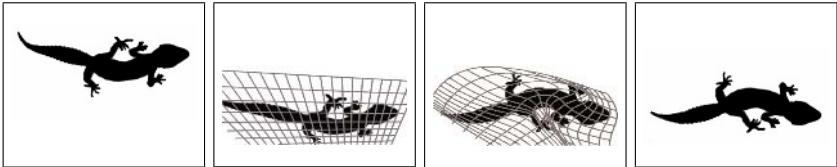


Fig. 2. Determining Long Range Optical Flow. The goal is to provide dense optical flow from the first frame (1), to the second (4). This is done via a planar fit (2) followed by a flexible fit (3).

that the support of the splines are disjoint subsets of the image plane. The task of fitting a mixture of splines naturally decomposes into two subtasks: motion segmentation and spline fitting. Ideally we would like to do both of these tasks simultaneously, however these tasks have conflicting goals. The task of motion segmentation requires us to identify groups of pixels whose motion can be described by a smooth transformation. Smoothness implies that each motion segment has the same gross motion, however except for the rare case in which the entire layer has exactly the same motion everywhere, there will be local variations. Hence the motion segmentation algorithm should be sensitive to inter-layer motion and insensitive to intra-layer variations. On the other hand, fitting a spline to each motion field requires attention to all the local variations. This is an example of different tradeoffs between bias and variance in the two stages of the algorithm. In the first stage we would like to exert a high bias and use models with a high amount of stiffness and insensitivity to local variations, whereas in the second stage we would like to use a more flexible model with a low bias.

The motion segmentation consists of a two stage RANSAC-based robust estimation procedure which operates on a sparse set of correspondences between the two frames. Any planar transformation can be used as the motion model in this stage; we use homographies in this paper. Once the dominant motions have been detected, a dense assignment is performed using a fast graph partitioning algorithm.

The output of the first stage, while sufficient to achieve a good segmentation is not sufficient to recover the optical flow accurately. However it serves two important purposes: firstly it provides an approximate segmentation of the sparse correspondences that allows for coherent groups to be processed separately. This is crucial for the second stage of the algorithm as a flexible model will likely find an unwieldy compromise between distinct moving groups as well as outliers. Secondly, since the assignment is dense, it is possible to find matches for points that were initially mismatched by limiting the correspondence search space to points in the same motion layer. The second stage then bootstraps off of these estimates of motion and layer support to iteratively fit a thin plate spline to account for non-planarity or non-rigidity in the motion. Figure 2 illustrates this process.

We now describe the two stages of the algorithm in detail.

3.1 Detecting Dominant Planar Motion

We begin by finding planar approximations of the motion in the scene as well as a dense assignment of pixels to motion layers. We use the motion segmentation algorithm of [20]. An example of this is shown in Figure 3

Example of Planar Fit and Segmentation. Figure 3 shows an example of the output from the planar fit and segmentation process. In this figure we show the two images, I and I' , and the assignments for each pixel to a motion layer (one of the three detected motion fields). The columns represent the different motion fields and the rows represent the portions of each image that are assigned to a given motion layer. The motions are made explicit in that the pixel support from frame to frame is related exactly by a planar homography. Notice that the portions of the background and the dumpsters that were visible in both frames were segmented correctly, as was the man. The result of the spline fit for this example will be shown in Section 4.



Fig. 3. Notting Hill sequence. (Row 1) Original image pair of size 311×552 , (Row 2) Pixels assigned to warp layers 1-3 in I , (Row 3) Pixels assigned to warp layers 1-3 in I' .

3.2 Refining the Fit with a Flexible Model

The flexible fit is an iterative process using regularized radial basis functions, in this case Thin Plate Spline (TPS). The spline interpolates the correspondences to result in a dense optical flow field. This process is run on a per-motion layer basis.

Feature Extraction and Matching. During the planar motion estimation stage, only a gross estimate of the motion is required so a sparse set of feature points will suffice. In the final fit however, we would like to use as many correspondences as possible to ensure a good fit. In addition, since the correspondence search space is reduced (i.e. matches are only considered between pixels assigned to corresponding motion layers), matching becomes somewhat simpler. For this reason, we use the Canny edge detector to find the set of edge points in each of the frames and estimate correspondences in the same manner as in [20].

Iterated TPS Fitting. Given the approximate planar homography and the set of correspondences between edge pixels, we would like to find the dense set of correspondences. If all of the correspondences were correct, we could jump

straight to a smoothed spline fit to obtain dense (interpolated) correspondences for the whole region. However, we must account for the fact that many of the correspondences are incorrect. As such, the purpose of the iterative matching is essentially to distinguish inliers from outliers, that is, we would like to identify sets of points that exhibit coherence in their correspondences.

One of the assumptions that we make about the scenes we wish to consider is that the motion of the scene can be approximated by a set of planar layers. Therefore a good initial set of inliers are those correspondences that are roughly approximated by the estimated homography. From this set, we use TPS regression with increasingly tighter inlier thresholds to identify the final set of inliers, for which a final fit is used to interpolate the dense optical flow. We now briefly describe this process.

The Thin Plate Spline is the Radial Basis Function (RBF) that minimizes the following bending energy or integral bending norm [4],

$$I_f = \iint_{\mathbb{R}^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dx dy$$

where $f = f(x, y)$ represents the x or y component of the transformation for the pixel at position (x, y) . In our approach we use a regularized version of TPS fitting in which μ controls the tradeoff between data and smoothing in the cost functional

$$H[f] = \sum_i (v_i - f(x_i, y_i))^2 + \mu I_f$$

where v_i represents the target of the transformation and $f(x_i, y_i)$ is the mapped value for the point at location (x_i, y_i) . Since each point gets mapped to a new 2D position, we require two TPS transformations, one for the x -coordinates and another for the y -coordinates. We solve for this transformation as in [9].

We estimate the TPS mapping from the points in the first frame to those in the second where μ_t is the regularization factor for iteration t . The fit is estimated using the set of correspondences that are deemed inliers for the current transformation, where τ_t is the threshold for the t^{th} iteration. After the transformation is estimated, it is applied to the entire edge set and the set of correspondences is again processed for inliers, using the new locations of the points for error computation. This means that some correspondences that were outliers before may be pulled into the set of inliers and vice versa. The iteration continues on this new set of inliers where $\tau_{t+1} \leq \tau_t$ and $\mu_{t+1} \leq \mu_t$. We have found that three iterations of this TPS regression with incrementally decreasing regularization and corresponding outlier thresholds suffices for a large set of real world examples. Additional iterations produced no change in the estimated set of inlier correspondences.

- | |
|---|
| I. Estimate planar motion
1. Find correspondences between I and I'
2. Robustly estimate the motion fields
3. Densely assign pixels to motion layers |
| II. Refine the fit with a flexible model
4. Match edge pixels between I and I'
5. For $t=1:3$
6. Fit all correspondences within τ_t
using TPS regularized by μ_t
7. Apply TPS to set of correspondences
<i>Note:</i> $(\tau_{t+1} \leq \tau_t, \mu_{t+1} \leq \mu_t)$ |

Fig. 4. Algorithm Summary

This simultaneous tightening of the pruning threshold and annealing of the regularization factor aid in differentiating between residual due to localization error or mismatching and residual due to the non-planarity of the object in motion. When the pruning threshold is loose, it is likely that there will be some incorrect correspondences that will pass the threshold. This means that the spline should be stiff enough to avoid the adverse effect of these mismatches. However, as the mapping converges we place higher confidence in the set of correspondences passing the tighter thresholds. This process is similar in spirit to iterative deformable shape matching methods [2,6].

4 Experimental Results

We now illustrate our algorithm, which is summarized in Figure 4, on several pairs of images containing objects undergoing significant, non-planar motion. Since the motion is large, displaying the optical flow as a vector field will result in a very confusing figure. Because of this, we show the quality of the optical flow in other ways, including (1) examining the image and corresponding reconstruction error that result from the application of the estimated transform to the original image (we refer to this transformed image as $\mathcal{T}(I)$), (2) showing intermediate views (as in [11]), or by (3) showing the 3D reconstruction induced by the set of dense correspondences. Examples are presented that exhibit either non-planarity, non-rigidity or a combination of the two. We show that our algorithm is capable of providing optical flow for pairs of images that are beyond the scope of existing algorithms. We performed all of the experiments on grayscale images using the same parameters¹.

4.1 Face Sequence

The first example is shown in Figures 5 and 6. The top row of Figure 5 shows the two input frames, I and I' , in which a man moves his head to the left in

¹ $k = 2, \lambda = .285, \tau_p = 15, \mu_1 = 50, \mu_2 = 20, \mu_3 = 1, \tau_1 = 15, \tau_2 = 10, \tau_3 = 5$. Here, k , λ , and τ_p refer to parameters in [20].

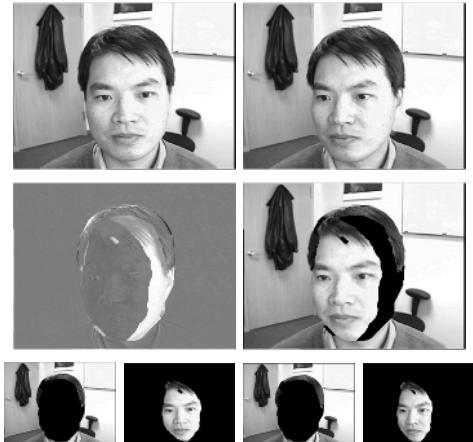


Fig. 5. Face Sequence. (1) The two input images, I and I' of size 240×320 . (2) The difference image is show first where grey regions indicate zero error regions and the reconstruction, $\mathcal{T}(I)$ is second. (3) The initial segmentation found via planar motion.

front of a static scene (the nose moves more than 10% of the image width). The second row shows first the difference image between $\mathcal{T}(I)$ and I' where error values are on the interval $[-1, 1]$ and gray regions indicate areas of zero error. This image is followed by $\mathcal{T}(I)$; this image has two estimated transformations, one for the face and another for the background. Notice that error in the overlap of the faces is very small, which means that according to reconstruction error, the estimated transformation successfully fits the relation between the two frames. This transformation is non-trivial as seen in the change in the nose and lips as well as a shift in gaze seen in the eyes, however all of this is captured by the estimated optical flow. The final row in Figure 5 shows the segmentation and planar approximation from [20], where the planar transformation is made explicit as the regions' pixel supports are related exactly by a planar homography. Dense correspondences allow for the estimation of intermediate views via interpolation as in [11]. Figure 6 shows the two original views of the segment associated with the face as well as a synthesized intermediate view that is realistic in appearance. The second row of this figure shows an estimation of relative depth that comes from the disparity along the rectified horizontal axis. Notice the shape of the nose and lips as well as the relation of the eyes to the nose and forehead. It is important to remember that no information specific to human faces was provided to the algorithm for this optical flow estimation.

4.2 Notting Hill Sequence

The next example shows how the spline can also refine what is already a close approximation via planar models. Figure 7 shows a close up of the planar error image, the reconstruction error and finally the warped grid for the scene that was shown in Figure 3. The planar approximation was not able to capture the 3D nature of the clothing and the non-rigid motion of the head with respect to the torso, however the spline fit captures these things accurately.

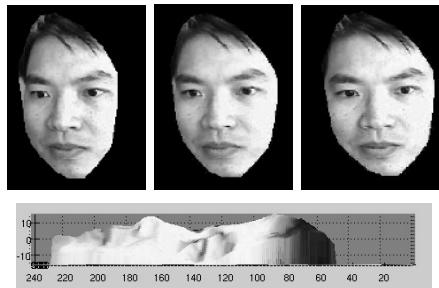


Fig. 6. Face Sequence – Interpolated views.
(1) Original frame I' , synthesized intermediate frame, original frame I , (2) A surface approximation from computed dense correspondences.

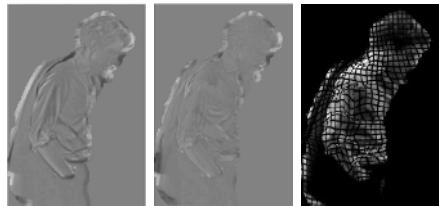


Fig. 7. Notting Hill. Detail of the spline fit for a layer from Figure 3, difference image for the planar fit, difference image for the spline fit, grid transformation.

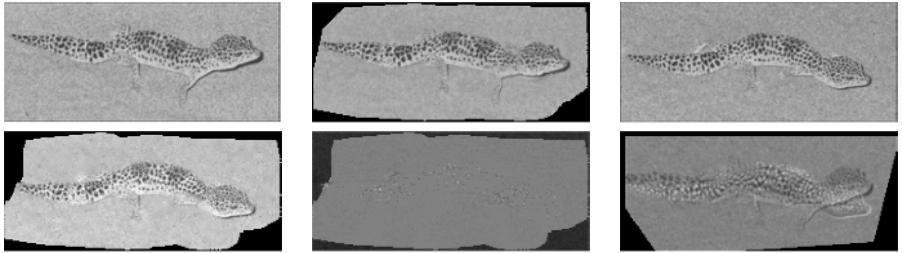


Fig. 8. Gecko Sequence. (1) Original frame I of size 102×236 , synthesized intermediate view, original frame I' . (2) $\mathcal{T}(I)$, Difference image between the above image and I' (gray is zero error), Difference image for the planar fit.

4.3 Gecko Sequence

The second example, shown in Figure 8, displays a combination of a non-planar object (a gecko lizard), undergoing non-rigid motion. While this is a single object sequence, it shows the flexibility of our method to handle complicated motions. In Figure 8(1), the two original frames are shown as well as a synthesized intermediate view (here, intermediate refers to time rather than viewing direction since we are dealing with non-rigid motion). The synthesized image is a reasonable guess at what the scene would look like midway between the two input frames. Figure 8(2) shows $\mathcal{T}(I)$ as well as the reconstruction error for the spline fit ($\mathcal{T}(I) - I'$), and the error incurred with the planar fit. We see in the second row of Figure 8(2) that the tail, back and head of the gecko are aligned very well and those areas have negligible error. When we compare the reconstruction error to the error induced by a planar fit, we see that the motion of the gecko is not well approximated by a rigid plane. Here, there is also some 3D motion present in that the head of the lizard changes in both direction and elevation. This is captured by the estimated optical flow.

4.4 Rubik's Cube

The next example shows a scene with rigid motion of a non-planar object. Figure 9 displays a Rubik's cube and user's manual switching places as the cube rotates in 3D. Below the frames, we see the segmentation that is a result of the planar approximation. It is important to remember that the background in this scene has no distinguishing marks so there is nothing to say that pieces of the background didn't actually move with the objects.

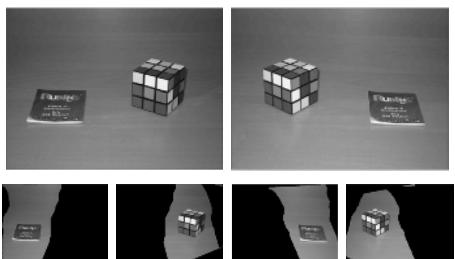


Fig. 9. Rubik's Cube. (1) Original image pair of size 300×400 , (2) assignments of each image to layers 1 and 2.

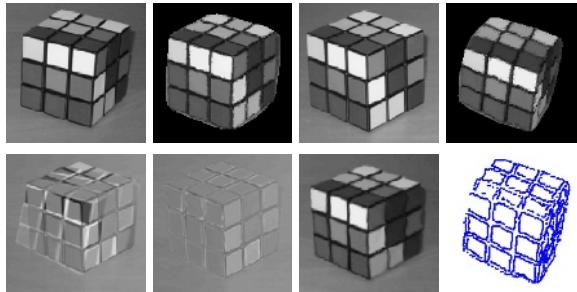


Fig. 10. Rubik’s Cube – Detail. (1) Original frame I , synthesized intermediate frame, original frame I' , A synthesized novel view, (2) difference image for the planar fit, difference image for the spline fit, $\mathcal{T}(I)$, the estimated structure shown for the edge points of I . We used dense 3D structure to produce the novel view.

Figure 10 shows $\mathcal{T}(I)$, the result of the spline fit applied to this same scene. The first row shows a detail of the two original views of the Rubik’s cube as well as a synthesized intermediate view. Notice that the rotation in 3D is accurately captured and demonstrated in this intermediate view. The second row shows the reconstruction errors, first for the planar fit and then for the spline fit, followed by $\mathcal{T}(I)$. Notice how accurate the correspondence is since the spline applied to the first image is almost identical to the second frame.

Correspondences between portions of two frames that are assumed to be projections of rigid objects in motion allow for the recovery of the structure of the object, at least up to a projective transformation. In [15], the authors show a sparse point-set from a novel viewpoint and compare it to a real image from the same viewpoint to show the accuracy of the structure. Figure 10 shows a similar result, however since our correspondences are dense, we can actually render the novel view that validates our structure estimation. The novel viewpoint is well above the observed viewpoints, yet the rendering as well as the displayed structure is fairly accurate. Note that only the set of points that were identified as edges in I are shown; this is not the result of simple edge detection on the rendered view. We use this display convention because the entire point-set is too dense to allow the perception of structure from a printed image. However, the rendered image shows that our estimated structure was very dense. It is important to note that the only assumption that we made about the object is that it is a rigid, piecewise smooth object. To achieve similar results from sparse correspondences would require additional object knowledge, namely that the object in question is a cube and has planar faces. It is also important to point out that this is not a standard stereo pair since the scene contains multiple objects undergoing independent motion.

5 Discussion

Since splines form a family of universal approximators over \mathbb{R}^2 and can represent any 2D transformation to any desired degree of accuracy, it raises the question

as to why one needs to use two different motion models in the two stages of the algorithm. If one were to use the affine transform as the dominant motion model, splines with an infinite or very large degree of regularization can indeed be used in its place. However, in the case where the dominant planar motion is not captured by an affine transform and we need to use a homography, it is not practical to use a spline. This is so because the set of homographies over any connected region of \mathbb{R}^2 are unbounded, and can in principle require a spline with an unbounded number of knots to represent an arbitrary homography. So while a homography can be estimated using a set of four correspondences, the corresponding spline approximation can, in principle, require an arbitrarily large number of control points. This poses a serious problem for robust estimation procedures like RANSAC since the probability of hitting the correct model decreases exponentially with increasing degrees of freedom. Many previous approaches for capturing long range motion are based on the fundamental matrix. However, since the fundamental matrix maps points to lines, translations in a single direction with varying velocity and sign are completely indistinguishable, as pointed out, e.g. by [16]. This type of motion is observed frequently in motion sequences. The trifocal tensor does not have this problem; however, like the fundamental matrix, it is only applicable for scenes with rigid motion and there is not yet a published solution for dense stereo correspondence in the presence of multiple motions.

6 Conclusion

In this paper, we have presented a new method for determining long range optical flow. We have shown that dense optical flow can now be estimated for image pairs with large disparity, multiple independently moving objects, and non-planar (including non-rigid) motion. Our approach is a two-stage framework based on a planar motion model for capturing the gross motion of the group followed by regularized spline fitting for capturing finer scale variations.

Our approach is intentionally generic in that it requires no object knowledge. However, in many cases, information about the types of objects in question could be used. The partitioning and initial estimation of gross motion may benefit from the use of articulated/object models. While a general solution using known models would require a solution to object recognition, incorporating object knowledge and models in specific domains will be the subject of future research.

Acknowledgments. We would like to thank Sameer Agarwal, Charless Fowlkes and Ben Ochoa for helpful discussions. The images in Figures 5 and 6 are used courtesy of Dr. Philip Torr. This work was partially supported under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 and by an NSF IGERT Grant (Vision and Learning in Humans and Machines, #DGE-0333451).

References

1. S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV 95*, pages 777–784, 1995.
2. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
3. M. Black and A. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *T-PAMI*, 18:972–986, 1996.
4. F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.
5. M. Brand. Morphable 3D models from video. In *IEEE Computer Vision and Pattern Recognition*, December 2001.
6. H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 44–51, June 2000.
7. M. Irani and P. Anandan. All about direct methods. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*. Springer-Verlag, 1999.
8. J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998.
9. M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Computational Techniques and Applications (CTAC95)*, Melbourne, Australia, 1995.
10. H. S. Sawhney and A. R. Hanson. Trackability as a cue for potential obstacle identification and 3D description. *International Journal of Computer Vision*, 11(3):237–265, 1993.
11. S. M. Seitz and C. R. Dyer. View morphing. In *SIGGRAPH*, pages 21–30, 1996.
12. S. Soatto and A. J. Yezzi. DEFORMOTION: Deforming motion, shape average and the joint registration and segmentation of images. In *European Conference on Computer Vision*, pages 32–47. Springer, 2002. Copenhagen.
13. R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Conference on Computer Vision Pattern Recognition*, pages 194–201, Seattle, Washington, 1994.
14. R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1210, 1996.
15. C. Tomasi and T. Kanade. Factoring image sequences into shape and motion. In *Proc. IEEE Workshop on Visual Motion*. IEEE, 1991.
16. P. H. S. Torr, A. Zisserman, and D. W. Murray. Motion clustering using the trilinear constraint over three views. In R. Mohr and C. Wu, editors, *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision*, pages 118–125. Springer-Verlag, 1995.
17. L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modelling non-rigid objects with rank constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–500, Kauai, Hawaii, 2001.
18. J. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 361–366, 1993.

19. Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 520–526, 1997.
20. J. Wills, S. Agarwal, and S. Belongie. What went where. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, June 2003.

Combining Geometric- and View-Based Approaches for Articulated Pose Estimation

David Demirdjian

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
`demirdji@ai.mit.edu`

Abstract. In this paper we propose an efficient real-time approach that combines vision-based tracking and a view-based model to estimate the pose of a person. We introduce an appearance model that contains views of a person under various articulated poses. The appearance model is built and updated online. The main contribution consists of modeling, in each frame, the pose changes as a linear transformation of the view change. This linear model allows (i) for predicting the pose in a new image, and (ii) for obtaining a better estimate of the pose corresponding to a key frame. Articulated pose is computed by merging the estimation provided by the tracking-based algorithm and the linear prediction given by the view-based model.

1 Introduction

Speed and robustness are usually the two important features of a vision-based face or person tracking algorithm. Though real-time tracking techniques have been developed and work well in laboratories (compliant users, stable and adapted lightning), they tend to break easily when used in real conditions (users performing fast moves, being occluded or only partially in the field of view of the camera). Tracking algorithms failures usually require a re-initialization, which prevents therefore their use in many applications.

In this paper we address the problem of robustness in tracking algorithms. We propose an efficient online real-time approach that combines vision-based tracking and a view-based model to estimate the pose of an articulated object. We introduce an appearance model that contains views (or key frames) of a person under various articulated poses. The appearance model is built and updated online. The main contribution consists of modeling, in each frame, the pose change as a linear transformation of the view change (optical flow). This linear model allows (i) for predicting the pose in a new image, and (ii) for obtaining a better estimate of the pose that corresponds to a key frame. Articulated pose is computed by merging the estimation provided by the tracking-based algorithm and the linear prediction given by the view-based model.

The following section discusses previous work for tracking and view-based models. Section 3 introduces our view-based model and shows how such a model

is used to predict the articulated pose in a new image. Section 4 describes our standard recursive tracking algorithm. We then present the general framework that combines recursive tracking and view-based model in Section 5. Finally we report experiments with our approach in Section 6 and discuss the general use of our approach in Section 7.

2 Previous Work

Vision-based tracking of articulated objects has been an active and growing research area in the last decade due to its numerous potential applications. Approaches to track articulated models in monocular image sequences have been proposed. Dense optical flow has been used in differential approaches where the gradient in the image is linearly related to the model movement [2,17]. Since monocular motion-based approaches only estimate relative motion from frame to frame, small errors are accumulated over time and cause the pose estimation to be sensitive to *drift*.

Recently, systems for 3-D tracking of hand and face features using stereo has been developed [8,4,9,5,12]. Such approaches usually minimize a fitting function error between a geometric model (limbs modeled as quadrics, cylinders, soft objects, ...) and visual observations (tridimensional scene reconstructions, colors). The minimization is usually performed locally (initialized with the pose estimated at the previous frame) and therefore subject to local minima, causing the tracking to easily fail when, for instance, motions between frames are important. To prevent this pit-fall that is caused by local minima, many researchers investigated stochastic optimization technics such as particle filtering [13,14]. Though promising, these approaches are very time-consuming and cannot yet be implemented for real-time purposes.

In this paper, we propose to tackle the problem of local minima in the minimization of the fitting function error by recovering tracking failures using a view-based model. View-based models have been mainly developed for representing the appearance of a rigid object from different points of view [10]. These appearance models are usually trained on images labeled with sets of landmarks, used for image point matching between frames, and annotated with the corresponding rigid pose. These models are able to capture the shape and appearance variations between people. The main drawback, however, is that the training phase is painstakingly long (requiring manual point matching between hundreds of images) and the pose estimate is very approximate. [3] recently proposed an approach for increasing the pose estimation accuracy in view-based models by using a linear subspace for shape and texture.

Recent work has suggested the combination of traditional tracking algorithms with view-based models. [16] proposes a simple approach that uses a set of pose-annotated views to re-initialize a standard recursive tracking algorithm. However the approach assumes that the annotation is manual and offline. A similar approach is proposed in [11] where an adaptive view-based model is used to reduce the drift of a differential face tracking algorithm. The authors

introduce an interesting linear Gaussian filter that simultaneously estimates the correct pose of a user face and updates the view-based model.

3 View-Based Model

In this paper, we assume that the body model to be articulated. Pose $\boldsymbol{\Pi}$ of a body is defined as the position of the torso and the relative orientation between consecutive limbs. We introduce here a view-based model \mathcal{M} that represents the relationship between visual information and articulated pose $\boldsymbol{\Pi}$.

Our view-based model \mathcal{M} consists of a collection of key frames \mathcal{F} . Each key frame contains information about the visual information (view), the pose associated with the view and a linear transformation that relates the pose change with respect to the view change. Different approaches have been proposed to model image deformation (morphable models, active appearance models,). In this paper, we model image deformations by considering the optical flow around a set of support feature points \mathbf{f}_i . A key frame \mathcal{F} is defined as:

$$\mathcal{F} = \{J, \mathbf{x}, \mathbf{L}, \boldsymbol{\Pi}_0\}$$

where J is the view (intensity image) associated with the key frame. $\mathbf{x} = (\mathbf{f}_1, \dots, \mathbf{f}_N)^\top$ is a vector formed by stacking the location of the feature points \mathbf{f}_i . $\boldsymbol{\Pi}_0$ is the articulated pose associated with the view J . \mathbf{L} is a matrix that represents the local linear transformation between the articulated pose $\boldsymbol{\Pi}$ and the image flow between a new view I and the view J :

$$\boldsymbol{\Pi} = \boldsymbol{\Pi}_0 + \mathbf{L}\mathbf{dx} \quad (1)$$

where $\mathbf{dx} = \mathbf{x}' - \mathbf{x}$ is the image motion between the support points location \mathbf{x}' in the image I and original support points \mathbf{x} in image J .

Modeling the linear transformation between articulated pose and image deformation allows a compact representation of the information contained in similar views. Therefore it enables to span a larger part of the appearance space with fewer key frames. It also provides a better estimate of the articulated pose.

3.1 Pose Prediction

Given a new image I , not necessarily present in the view-based model, an estimation of the corresponding articulated pose $\boldsymbol{\Pi}$ is obtained as follow:

- The key frame \mathcal{F}_k which image J_k is closest to I with respect to image distance $d_{\mathcal{I}}(.,.)$ is selected.
- The image motion $\mathbf{dx}^{(k)}$ of support points $\mathbf{f}^{(k)}$ between images J_k and I is estimated;
- The pose $\boldsymbol{\Pi}$ is predicted as $\boldsymbol{\Pi} = \boldsymbol{\Pi}_0^{(k)} + \mathbf{L}^{(k)}\mathbf{dx}$



Fig. 1. The left image shows the current image. The right image shows the detected key frame of the view-based model, optical flow of the support points (in blue) and the prediction of the articulated body pose from the linear model (in white).

In our current implementation $d_{\mathcal{I}}(I, J_k)$ is defined as the weighted sum of absolute pixel differences between images I and J_k :

$$d_{\mathcal{I}}(I, J_k) = \sum_{i,j} w_{i,j} |I(i,j) - J_k(i,j)|$$

where (i, j) are pixel coordinates and $w_{i,j}$ some *foreground weights* that account for the fact that pixels (i, j) in image I correspond to foreground ($w_{i,j} = 1$) or background ($w_{i,j} = 0$). Weights $w_{i,j}$ are, in this paper, estimated by using a foreground detection algorithm similar to [15]. This algorithm updates online a background model and therefore performs a *robust* foreground detection, allowing our approach to be robust to slowly varying backgrounds

Figure 1 shows an example of detected key frame and linear prediction from the view-based model. The approach we present here consists in building and using such a view-based model to improve the robustness and accuracy of a tracking-based pose estimation algorithm.

4 Model-Based Tracking

This section briefly describes our real-time model-based tracking algorithm previously published in [5]. Our approach uses a force driven technique similar to [4,9] that allows the enforcement of different kind of constraints on the body pose (joint angles, orientation, ...). These constraints can eventually be learnt from examples using a Support Vector Machine [6]. For simplicity, only the force driven technique is described here.

We consider the pose estimation problem as the fitting of a body model pose Π to a set of visual observations. When visual observations come from a stereo or multi-view camera, tridimensional reconstructions $\mathcal{P} = \{M_i\}$ of the points M_i



Fig. 2. Our geometric-based tracking algorithm minimizes the Euclidean distance between an articulated model (left image) and the 3D reconstruction of disparity image (middle image) corresponding to the scene (right image).

in the scene can be estimated. In this case, a fitting error function $E(\boldsymbol{\Pi})$ defined as the distance between reconstructed points \mathcal{P} and the 3D model at pose $\boldsymbol{\Pi}$ is suitable. Such a function can be defined such that:

$$E^2(\boldsymbol{\Pi}) = \sum_{M_i \in \mathcal{P}} d^2(M_i, \mathcal{B}(\boldsymbol{\Pi})) \quad (2)$$

where $\mathcal{B}(\boldsymbol{\Pi})$ is 3D reconstruction of the body model at pose $\boldsymbol{\Pi}$ and $d^2(M_i, \mathcal{B}(\boldsymbol{\Pi}))$ the Euclidean distance between the point M_i and the 3D model $\mathcal{B}(\boldsymbol{\Pi})$.

A direct approach for pose tracking consists in minimizing the fitting error $E(\boldsymbol{\Pi})$ using a recursive scheme: the pose $\boldsymbol{\Pi}_{t-1}$ estimated at the previous frame is used as initialization in a local optimization algorithm that searches for directions $\boldsymbol{\tau}$ around $\boldsymbol{\Pi}_{t-1}$ that minimize the fitting error $E(\boldsymbol{\Pi} + \boldsymbol{\tau})$.

The iterative tracking algorithm consists of 2 steps: (i) an **ICP step** that estimates a set of unconstrained rigid motions δ_k (or forces) to apply to the articulated body to minimize eq.(2) and (ii) an **articulated constraints enforcing step** that finds a set of rigid motions δ_k^* that satisfy articulated constraints while minimizing a Mahalanobis distance w.r.t. rigid motions δ_k . The main steps of this tracking algorithm are recalled below.

ICP step. Given a set of 3D data and a 3D model of a rigid object to register, ICP [1] estimates the motion transformation between the 3D model and the rigid object. The ICP algorithm is applied to each limb \mathcal{L}_k independently, estimating a motion transformation δ_k , and its uncertainty Λ_k .

Articulated constraints enforcing. Motion transformations δ_k correspond to 'directions' that minimize the distance between limbs \mathcal{L}_k and the reconstructed 3D points of the scene. However, altogether δ_k do not satisfy articulated constraints (due to the spherical joints between adjacent limbs).

Let $\Delta = (\delta_1, \dots, \delta_N)^\top$ be the (unconstrained) set of rigid motions and $\Delta^* = (\delta_1^*, \dots, \delta_N^*)^\top$ be a set of rigid motions satisfying articulated constraints. A correct set of motion transformation Δ^* that satisfy the spherical joints constraint can be found by projecting the set of rigid motions δ_k onto the manifold defined by

articulated motions (see [5,6] for details). The projection is linear (hypothesis of small angle rotations) and minimizes the following Mahalanobis distance $\epsilon^2(\Delta^*)$:

$$\begin{aligned}\epsilon^2(\Delta^*) &= \|\Delta^* - \Delta\|_{\Lambda}^2 \\ &= (\Delta^* - \Delta)^T \Lambda^{-1} (\Delta^* - \Delta)\end{aligned}\quad (3)$$

where Λ is the covariance (block-diagonal) matrix $\Lambda = \text{diag}(\Lambda_1, \Lambda_2, \dots)$.

The projection is written $\Delta^* = \mathbf{P}\Delta$, where \mathbf{P} is a projection matrix whose entries are computed only from the covariance matrix Λ and the position of the spherical joints (before motion).

5 Tracking with Key Frames

This section describes how model-based tracking and the view-based model are combined.

At each new frame, articulated poses are estimated independently using the recursive (ICP-based) tracking algorithm and the view-based model. The correct pose is chosen so that it minimizes the fitting error function. Figure 3 illustrates the combined pose estimation algorithm.

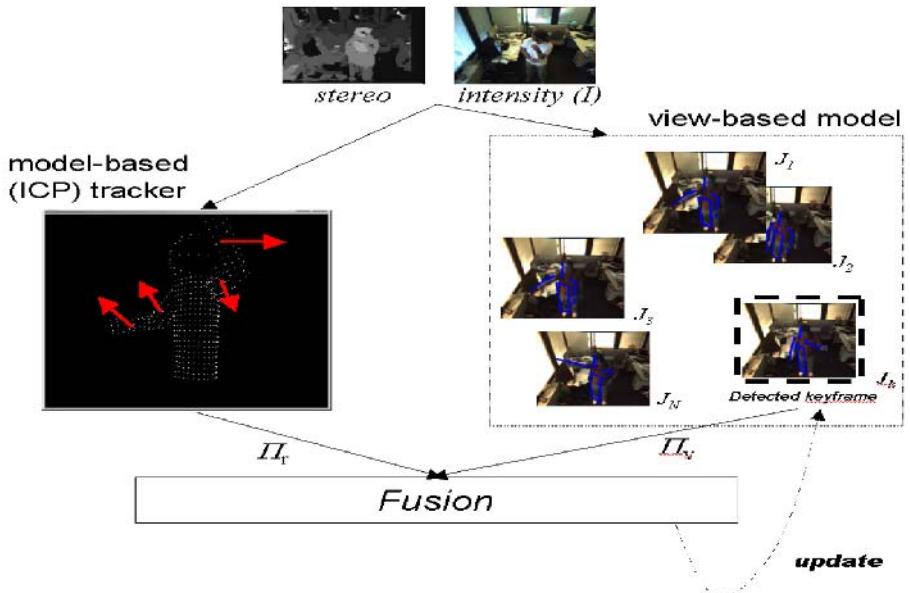


Fig. 3. Combined pose estimation.

Let $\boldsymbol{\Pi}_r$ be the pose estimated by applying the ICP-based tracking algorithm (Section 4) to the pose found at the previous frame. Let $\boldsymbol{\Pi}_v$ be the prediction given by the view-based model (Section 3.1). $\boldsymbol{\Pi}_v$ is found by:

- searching for the key frame $\mathcal{F}_k = \{J_k, \mathbf{x}_k, \mathbf{L}_k, \boldsymbol{\Pi}_{0k}\}$, which view J_k is most similar to the current image I ;
- estimating the optical flow \mathbf{dx} of the support points \mathbf{x}_k between images J_k and I and computing $\boldsymbol{\Pi}_v = \boldsymbol{\Pi}_{0k} + \mathbf{L}_k \mathbf{dx}$.

The fitting error function $E(\boldsymbol{\Pi})$ defined in (2) is evaluated at $\boldsymbol{\Pi}_r$ and $\boldsymbol{\Pi}_v$. The pose corresponding to the smallest value of $E(\boldsymbol{\Pi}_r)$ and $E(\boldsymbol{\Pi}_v)$ is considered as the current pose:

$$\boldsymbol{\Pi} = \arg \min_{\boldsymbol{\Pi}} (E(\boldsymbol{\Pi}_r), E(\boldsymbol{\Pi}_v))$$

The view-based model is built online using images I (observed during the tracking) and pose estimates $\boldsymbol{\Pi}$. The next sections describe how new key frames are added in the view-based model and detail the process for updating existing key frames.

5.1 Key Frames Selection

The maximum number N of key frames in the view-based model \mathcal{M} is obviously limited by the speed¹ and memory² of the CPU. Therefore the choice of key frames to keep in the view-based model is crucial.

Many criteria can be considered to select the key frames (frames for which the tracking is accurate, frames appearing frequently, ...). In this paper, we prefer keeping the key frames which span a maximum of the appearance space. This can be done by selecting key frames that maximizes an intra-class distance $\mathcal{D}(\mathcal{M})$ between key frames.

Let $\mathcal{S}(\mathcal{F}, \mathcal{F}')$ be a distance between key frames \mathcal{F} and \mathcal{F}' . The corresponding intra-class distance $\mathcal{D}(\mathcal{M})$ is defined as:

$$\mathcal{D}(\mathcal{M}) = \sum_{\{\mathcal{F}, \mathcal{F}'\} \subset \mathcal{M}} \mathcal{S}(\mathcal{F}, \mathcal{F}') = \sum_{\mathcal{F}} \sum_{\mathcal{F}' \neq \mathcal{F}} \mathcal{S}(\mathcal{F}, \mathcal{F}')$$

Let \mathcal{F}_k be a key frame from the view-based model \mathcal{M} and \mathcal{F}_{new} be a new key frame. If \mathcal{F}_{new} is such that:

$$\sum_{\mathcal{F} \in \mathcal{M}, \mathcal{F} \neq \mathcal{F}_k} \mathcal{S}(\mathcal{F}_{new}, \mathcal{F}) > \sum_{\mathcal{F} \in \mathcal{M}, \mathcal{F} \neq \mathcal{F}_k} \mathcal{S}(\mathcal{F}_k, \mathcal{F}) \quad (4)$$

then the view-based model \mathcal{M}_{new} obtained by replacing the key frame \mathcal{F}_k by \mathcal{F}_{new} in the view-based model \mathcal{M} satisfies $\mathcal{D}(\mathcal{M}_{new}) > \mathcal{D}(\mathcal{M})$.

In practice, we keep a current estimate of the *weakest* key frame $\mathcal{F}_{min} \in \mathcal{M}$ such that:

$$\mathcal{F}_{min} = \arg \min_{\mathcal{F}} \sum_{\mathcal{F}' \neq \mathcal{F}} \mathcal{S}(\mathcal{F}, \mathcal{F}')$$

When a new frame \mathcal{F}_{new} satisfies (4) with $\mathcal{F}_k = \mathcal{F}_{min}$, then \mathcal{F}_{min} is replaced by \mathcal{F}_{new} , therefore increasing the intra-class distance of \mathcal{M} .

¹ The pose prediction algorithm involves a comparison between the current image I and the images of *all* key frames

² Because of real-time issues, frames cannot be stored on disk

5.2 Key Frame Update

In this section, we show how the parameters \mathbf{x} , \mathbf{L} , $\boldsymbol{\Pi}_0$ of a key frame $\mathcal{F} = \{J, \mathbf{x}, \mathbf{L}, \boldsymbol{\Pi}_0\}$ are estimated. Let J_k (with $1 \leq k \leq N$) be a set of images *similar* to J , and $\boldsymbol{\Pi}_k$ the corresponding articulated pose. Let $d\mathbf{f}_k$ be the motion of a feature point \mathbf{f} between the images J and J_k .

Support points. First, support points \mathbf{x} are estimated as the set of feature points \mathbf{f}_i detected as being part of the articulated object to track. In our current framework, support points \mathbf{x} are chosen so that they correspond to pixels detected as foreground. In practice, we use the foreground weights $w_{i,j}$ introduced in section 3.1. A pixel (i, j) is considered as a support point if its average foreground weight $\bar{w}_{i,j}$ across images J_k is higher than a threshold τ .

Linear model: \mathbf{L} , $\boldsymbol{\Pi}_0$. Let $d\mathbf{x}_k$ be the motion of the support points $\mathbf{x} = (\mathbf{f}_1 \mathbf{f}_2 \dots)^\top$ between the images J and J_k . The matrix \mathbf{L} and vector $\boldsymbol{\Pi}_0$ are constrained by the linear equations (1) corresponding to the observations $(\boldsymbol{\Pi}_k, d\mathbf{x}_k)$.

If the number of images J_k similar to J is too small, there are not enough constraints (1) to estimate \mathbf{L} and $\boldsymbol{\Pi}_0$. In the rest of this section, we assume that there are more constraints than entries in \mathbf{L} and $\boldsymbol{\Pi}_0$.

Solving eqs.(1) directly using a linear least square technique could lead to biased estimates of \mathbf{L} and $\boldsymbol{\Pi}_0$ because (i) the noise in the entries $\boldsymbol{\Pi}_k$ is not uniform and isotropic and (ii) the image motion of some of the support points \mathbf{x} may be mis-estimated due, for instance, to the aperture problem or the presence of similar textures. Therefore we propose a robust scheme to solve for \mathbf{L} and $\boldsymbol{\Pi}_0$ that accounts for the presence of outliers in $d\mathbf{x}_k$.

Eq.(1) can be rewritten:

$$d\mathbf{x}_k = \mathbf{L}^{-1}(\boldsymbol{\Pi}_k - \boldsymbol{\Pi}_0) = \boldsymbol{\Gamma}\boldsymbol{\Pi}_k + \boldsymbol{\mu} \quad (5)$$

with

$$\boldsymbol{\Gamma} = \mathbf{L}^{-1} \quad \boldsymbol{\mu} = -\mathbf{L}^{-1}\boldsymbol{\Pi}_0 \quad (6)$$

Let the matrices $\boldsymbol{\Gamma}_i$ and vectors $\boldsymbol{\mu}_i$ be such that $\boldsymbol{\Gamma} = (\boldsymbol{\Gamma}_1^\top \dots \boldsymbol{\Gamma}_{N_f}^\top)^\top$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_1^\top \dots \boldsymbol{\mu}_{N_f}^\top)^\top$.

With $d\mathbf{x}_k = (\mathbf{f}_1^{(k)}, \mathbf{f}_2^{(k)}, \dots, \mathbf{f}_N^{(k)})^\top$ and considering only the lines of (5) corresponding to the support point motion $d\mathbf{f}_i^{(k)}$, it gives:

$$d\mathbf{f}_i^{(k)} = \boldsymbol{\Gamma}_i \boldsymbol{\Pi}_k + \boldsymbol{\mu}_i = \begin{pmatrix} \boldsymbol{\Gamma}_i^x \\ \boldsymbol{\Gamma}_i^y \end{pmatrix}^\top \boldsymbol{\Pi}_k + \boldsymbol{\mu}_i = \mathbf{P}_k \mathbf{q}_i \quad (7)$$

$$\text{where } \mathbf{P}_k = \begin{pmatrix} \boldsymbol{\Pi}_k^\top & 0 & 1 & 0 \\ 0 & \boldsymbol{\Pi}_k^\top & 0 & 1 \end{pmatrix} \quad \mathbf{q}_i = \begin{pmatrix} \boldsymbol{\Gamma}_i^x \\ \boldsymbol{\Gamma}_i^y \\ \boldsymbol{\mu}_i \end{pmatrix}$$

Vector \mathbf{q}_i is found by solving simultaneously eqs.(7) for all k using a robust optimization technique based on M-estimator [7]. More precisely, we introduce

an influence function $\rho(x, \sigma) = \log(1 + \frac{x^2}{2\sigma^2})$ and minimize the following objective function:

$$\sum_k \rho(||\mathbf{df}_i^{(k)} - \mathbf{P}_k \mathbf{q}_i||, \sigma) \quad (8)$$

The scalar σ corresponds to the expected covariance of the noise in the inliers (in our implementation, $\sigma = 2.0 \text{pix}$). It worth noticing that eq.(8) is actually solved using an iterative weighted linear least-square method (see [7] for details). Once vectors \mathbf{q}_i are estimated, \mathbf{L} and $\boldsymbol{\Pi}_0$ are estimated using (6).

5.3 Summary

The complete tracking algorithm can be summarized as follow:

- **Key frame search.** The key frame $\mathcal{F}_k = \{J_k, \mathbf{x}_k, \mathbf{L}_k, \boldsymbol{\Pi}_{0k}\}$ of the view-based model, which image J_k is the closest to the current image I is estimated;
- **Pose estimation.** Pose $\boldsymbol{\Pi}_v$ is predicted using the linear model (1) and optical flow $d\mathbf{x}$ between image I and J_k . Pose $\boldsymbol{\Pi}_r$ is estimated using the ICP-based algorithm. The pose minimizing the fitting error function (2) is chosen as the correct pose $\boldsymbol{\Pi}$;
- **View-based model update.** The optical flow $d\mathbf{x}$ is added as an additional constraint to update the linear model $(\mathbf{L}_k, \boldsymbol{\Pi}_{0k})$ of key frame \mathcal{F}_k . If image I satisfies criteria (4), then a new key frame \mathcal{F}_{new} is created (with image I and pose $\boldsymbol{\Pi}$).

6 Experiments

We applied the body tracking approach described previously to stereo image sequences captured in our lab. Experiments were done in order to compare the standard recursive (ICP-based) algorithm with our approach (ICP-based combined with a view-based model). The algorithms were run on a Pentium 4 (2GHz). The ICP-based algorithm alone runs at a speed ranging from 8Hz to 12Hz. The ICP-based algorithm combined with a view-based model runs at about 5Hz. In these experiments, the maximum number of key frames in the view-based model is $N = 100$.

In order to learn the view-based model, a training sequence of about 2000 images is used. The training sequence is similar to Figure 4 (same background/subject).

Figure 4 show some comparative results on a testing sequence of more than 1500 images. More exactly, the figure show the corresponding images of the sequence and re-projection of the 3D articulated model for frames 132, 206, 339, 515, 732 and 850. Results show that our approach enables to cope with re-initialization after tracking failure.

Figure 5 shows the average error between the estimation of the 3D model and the 3D scene reconstruction from the stereo camera for the two algorithms.

Additional sequences can be found at: <http://www.ai.mit.edu/~demirdji>.

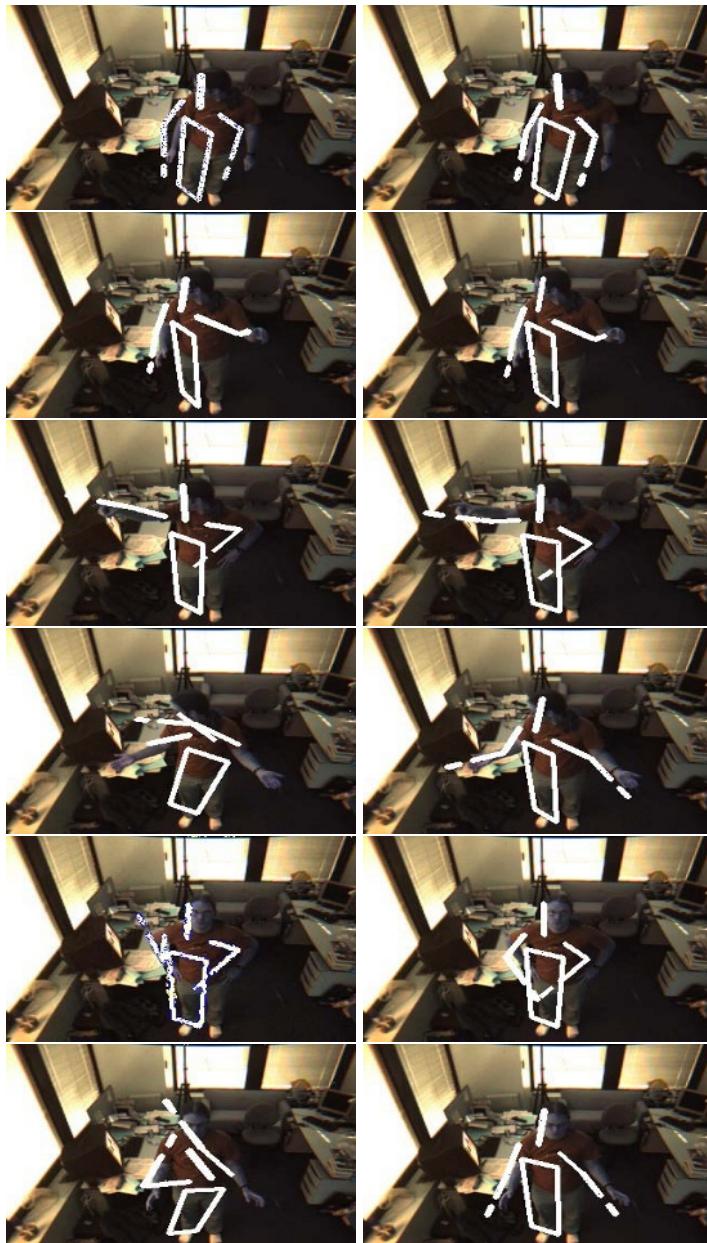


Fig. 4. Comparative results (re-projection of the 3D articulated model) on a sequence of more than 1500 images (lines correspond to frames 132, 206, 339, 515, 732 and 850). The graph shows that, with our approach (ICP + view-based model), the error is always smaller. The left column corresponds to the ICP-based tracking algorithm. The right column corresponds to our algorithm (ICP + view-based model).

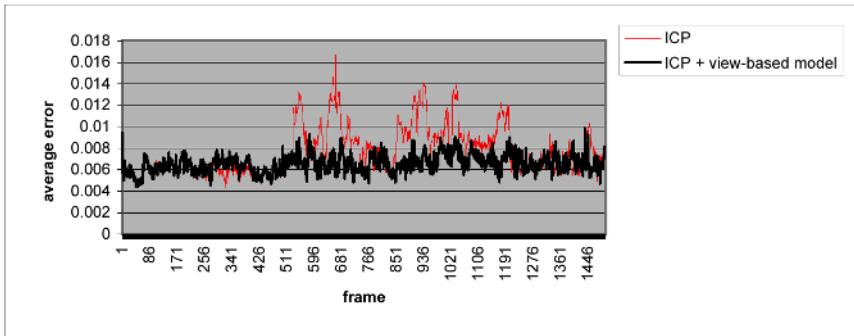


Fig. 5. Average error between the estimation of the 3D articulated model and the 3D scene reconstruction *vs.* number of frames. Peaks in the data (around frames 520, 670, 930, 1100, 1190) corresponding to the ICP algorithm are actually tracking failures.

7 Conclusion

We described an approach for real-time articulated body tracking. The approach combines traditional recursive vision-based tracking and a view-based model to estimate the pose of an articulated object. We introduce an appearance model that contains views (or key frames) of a person under various articulated poses. The appearance model is built and updated online. The main contribution consists in modeling, in each frame, the pose change as a linear transformation of the view change.

The experiments we carried out show that our approach significantly increases the robustness of the tracking by enabling an automatic re-initialization in case of failure of the traditional recursive tracking algorithm. Experiments are being carried out to show the accuracy of the linear predictor of the view-based model. The use of an online background learning algorithm allows our approach to be robust to slowly varying background. However, our approach is not robust to different clothing/person. In future work, we plan to extend our approach by introducing an adaptive appearance model to model the variability of appearance across people/clothes.

References

1. P.J. Besl and N. MacKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
2. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of Computer Vision and Pattern Recognition'98*, 1998.
3. T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *International Conference on Automatic Face and Gesture Recognition*, pages 227–232, Grenoble, France, 2000.

4. Q. Delamarre and O. D. Faugeras. 3D articulated models and multi-view tracking with silhouettes. In *Proceedings of ICCV'99*, pages 716–721, 1999.
5. D. Demirdjian. Enforcing constraints for human body tracking. In *Proceedings of Workshop on Multi-Object Tracking, Madison, Wisconsin, USA*, 2003.
6. D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *Proceedings of the International Conference on Computer Vision, Nice, France*, 2003.
7. F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stehel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, 1986.
8. N. Jojic, M. Turk, and T.S. Huang. Tracking articulated objects in dense disparity maps. In *International Conference on Computer Vision*, pages 123–130, 1999.
9. J.P. Luck, C. Debrunner, W. Hoff, Q. He, and D.E. Small. Development and analysis of a real-time human motion tracking system. In *Workshop on Applications of Computer Vision*, 2002.
10. B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the Identification and Inspection of Humans, SPIE'94*, volume 2257, 1994.
11. L.P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance models. In *Proceedings of Computer Vision and Pattern Recognition*, 2003.
12. R. Plankers and P. Fua. Articulated soft objects for video-based body modeling. In *ICCV*, Vancouver, Canada, July 2001.
13. H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3D human figures using 2d image motion. In *ECCV (2)*, pages 702–718, 2000.
14. C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3D body tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*. IEEE Computer Society Press, Dec 2001.
15. Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.
16. L. Vaccetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3d tracking in real-time. In *Proceedings of Computer Vision and Pattern Recognition*, 2003.
17. M. Yamamoto and K. Yagishita. Scene constraints-aided tracking of human body. In *Proceedings of Computer Vision and Pattern Recognition*, 2000.

Shape Matching and Recognition – Using Generative Models and Informative Features

Zhuowen Tu and Alan L. Yuille

Departments of Statistics,
University of California, Los Angeles, 90095 USA
`{ztu,yuille}@stat.ucla.edu`

Abstract. We present an algorithm for shape matching and recognition based on a generative model for how one shape can be generated by the other. This generative model allows for a class of transformations, such as affine and non-rigid transformations, and induces a similarity measure between shapes. The matching process is formulated in the EM algorithm. To have a fast algorithm and avoid local minima, we show how the EM algorithm can be approximated by using *informative features*, which have two key properties—*invariant* and *representative*. They are also similar to the proposal probabilities used in DDMCMC [13]. The formulation allows us to know when and why approximations can be made and justifies the use of bottom-up features, which are used in a wide range of vision problems. This integrates generative models and feature-based approaches within the EM framework and helps clarifying the relationships between different algorithms for this problem such as shape contexts [3] and softassign [5]. We test the algorithm on a variety of data sets including MPEG7 CE-Shape-1, Kimia silhouettes, and real images of street scenes. We demonstrate very effective performance and compare our results with existing algorithms. Finally, we briefly illustrate how our approach can be generalized to a wider range of problems including object detection.

1 Introduction

Shape matching has been a long standing problem in computer vision and it is fundamental for many tasks such as image compression, image segmentation, object recognition, image retrieval, and motion tracking. A great deal of effort has been made to tackle this problem and numerous matching criteria and algorithms have been proposed. For example, some typical criteria include Fourier analysis, moments analysis, scale space analysis, and the Hausdorff distance. For details of these methods see a recent survey paper [14].

The two methods most related to this paper are shape contexts [3] and softassign [5]. Shape contexts method is a feature-based algorithm which has demonstrated its ability to match certain types of shapes in a variety of applications. The softassign approach [5] formulates shape registration/matching as

free energy minimization problem using the mean field approximation. Recent improvements to these methods include the use of dynamic programming to improve shape contexts[12] and the Bethe-Kikuchi free energy approximation [9] which improves on the mean field theory approximation used in the softassign [5].

Our work builds on shape contexts [3] and softassign [5] to design a fast and effective algorithm for shape matching. Our approach is also influenced by ideas from the Data-Driven Markov Chain Monte Carlo (DDMCMC) paradigm [13] which is a general inference framework. It uses data-driven proposals to activate generative models and thereby guide a Markov Chain to rapid convergence.

First, we formulate the problem as Bayesian inference using generative models allowing for a class of shape transformations, see section (2). In section (3), we relate this to the free energy function for the EM algorithm [8] and, thereby, establish a connection to the free energy function used in softassign [5].

Secondly, we define a set of *informative features*, which observe two key properties: *invariant/semi-invariant* and *representative*, to shape transformations such as scaling, rotation, and certain non-rigid transformations, see sections (4.1,4.2). Shape contexts [3] are examples of informative features.

Thirdly, the generative model and informative features are combined in the EM free energy framework, see section (4.3,4.4). The informative features are used as approximations, similar to the proposals in DDMCMC [13], which guide the algorithm to activate the generative models and achieve rapid convergence. Alternatively, one can think of the informative features as providing approximations to the true probabilities distributions, similar to the mean field and Bethe-Kikuchi approximations used by Rangarajan *et al* [5],[9].

We tested our algorithm on a variety of binary and real images and obtained very good performance, see section (6). The algorithms was extensively tested on binary datasets where its performance could be compared to existing algorithms. But we also give results on real images for recognition and detection.

2 Problem Definition

2.1 Shape Representation

The task of shape matching is to match two arbitrary shapes, X and Y , and to measure the similarity (metric) between them. Following Grenander's pattern theory [6], we can define shape similarity in terms of the transformation F that takes one shape to the other, see Fig. 1. In this paper we allow two types of transformation: (i) a global affine transformation, and (ii) a local small and smooth non-rigid transformation.

We assume that each shape is represented by a set of points which are either *sparse* or *connected* (the choice will depend on the form of the input data).

For the **sparse point representation**, we denote the target and source shape respectively by:

$$X = \{(\mathbf{x}_i) : i = 1, \dots, M\}, \text{ and } Y = \{(\mathbf{y}_a) : a = 1, \dots, N\}.$$

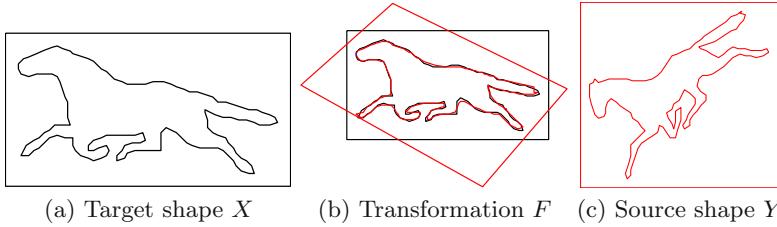


Fig. 1. Illustration of a shape matching case in which a source shape Y is matched with a target shape X through a transformation function F .

This representation will be used if we match a shape to the edge map of an image.

For the **connected point representation**, we denote the target and source shape respectively by:

$$X = \{(\mathbf{x}(s)) : s \in [0, 1]\}, \text{ and } Y = \{(\mathbf{y}(t)) : t \in [0, 1]\},$$

where s and t are normalized arc-length distances. This model is used for matching shapes to silhouettes. (The extension to multiple contours is straightforward.)

2.2 The Probability Models

We assume a shape X is generated by a shape Y by a transformation $F = (A, \mathbf{f})$ where A is an affine transformation, and \mathbf{f} denotes a non-rigid local transformation (in thin-plate-splines (TPS) [4], the two transformations are combined, but we separate them here for clarity). For any point \mathbf{y}_a on Y , let $v_a \in \{0..M\}$ be the correspondence variable to points in X . For example, $v_a = 4$ means that point \mathbf{y}_a on Y corresponds to point \mathbf{x}_4 on X . If $v_a = 0$, then \mathbf{y}_a is unmatched. We define $V = (v_a, a = 1..N)$. The generative model is written as

$$p(X|Y, V, (A, \mathbf{f})) \propto \exp\{-E_D(X, Y, V, (A, \mathbf{f}))\},$$

where

$$E_D(X, Y, V, (A, \mathbf{f})) = \sum_a (1 - \delta(v_a)) \|\mathbf{x}_{v_a} - A\mathbf{y}_a - \mathbf{f}(\mathbf{y}_a)\|^2 / \sigma^2. \quad (1)$$

and $(1 - \delta(v_a))$ is used to discount unmatched points (where $v_a = 0$). There is a prior probability $p(V)$ on the matches which pays a penalty for unmatched points. Therefore,

$$p(X, V|Y, V, (A, \mathbf{f})) \propto \exp\{-E_T(X, Y, V, (A, \mathbf{f}))\},$$

where $E_T(X, Y, V, (A, \mathbf{f})) = E_D(X, Y, V, (A, \mathbf{f})) - \log p(V)$.

The affine transformation A is decomposed [1] as

$$A = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}.$$

where θ is the rotation angle, S_x and S_y denote scaling, and k is shearing. The prior on A is given by $p(A) \propto \exp\{-E_A(A)\}$ where $E_A(A) = E_{rotation}(\theta) + E_{scaling}(S_x, S_y) + E_{shearing}(s)$.

The prior on the non-rigid transformation \mathbf{f} is given by

$$p(\mathbf{f}) \propto \exp\{-E_f(\mathbf{f})\}, \text{ and } E_f(\mathbf{f}) = \lambda \int \sum_{m=0}^{\infty} c_m (D^m \mathbf{f})^2 d\mathbf{y},$$

The $\{c_m\}$ are set to be $\sigma^{2m}/(m)2^m$ (Yuille and Grzywacz [15]). This enforces a probabilistic bias for the transformations to be small (the $c_0 = 1$ term) and smooth (the remaining terms $\{c_i : i \geq 1\}$). It can be shown [15], that \mathbf{f} is of the form $\mathbf{f}(x) = \sum_i \alpha_i G(x - x_i)$ where $G(x)$ is the Green's function of the differential operator. We use the Gaussian kernel for \mathbf{f} in this paper (alternative kernels such as TPS give similar results).

The generative model and the prior probabilities determine a similarity measure:

$$D(X||Y) = -\log p(X|Y) = -\log \int \sum_V p(X, V, (A, \mathbf{f})|Y) dAd\mathbf{f}. \quad (2)$$

Unfortunately evaluating eqn. (2) requires integrating out (A, \mathbf{f}) and summing out V . Both stages are computationally very expensive. Our strategy is to approximate the sum over the V , by using the informative features described in section (4). We then approximate the integral over (A, \mathbf{f}) by the modes of $p(A, \mathbf{f}|X, Y)$ (similar to a saddle point approximation). Therefore we seek to find the $(A, \mathbf{f})^*$ that best represent the distribution:

$$\int \sum_V p(X, V, (A, \mathbf{f})|Y) dAd\mathbf{f} \sim Par(p(X, (A, \mathbf{f})^*|Y)) \quad (3)$$

where *Par* is a Parzen window. Our experiments show that the integral is almost always dominated by $(A, \mathbf{f})^*$. Therefore, we approximate the similarity measure by:

$$D_{Appox}(X||Y) = -\log \sum_V p(X, V, (A, \mathbf{f})^*|Y), \quad (4)$$

where

$$\begin{aligned} (A, \mathbf{f})^* &= \arg \max_{(A, \mathbf{f})} \sum_V p(X, V, (A, \mathbf{f})|Y) \\ &= \arg \min_{(A, \mathbf{f})} -\log \sum_V p(X, V|Y, V, (A, \mathbf{f})) p(A) p(\mathbf{f}). \end{aligned} \quad (5)$$

In rare cases, we will require the sum over several models. For example, three modes $((A, \mathbf{f})^*, (A, \mathbf{f})_1^*, (A, \mathbf{f})_2^*)$ are required when matching two equal lateral triangles, see Fig. (2).

Note that this similarity measure is not symmetric between X and Y . But in practice, we found that it was approximately symmetric unless one shape was

significantly larger than the other (because of how the A scales the measure). To avoid this problem, we can compute $D(X||Y) + D(Y||X)$. The recognition aspect of the algorithm can be naturally extended from the similarity measure for the two shapes.

3 The EM Free Energy

Computing $(A, \mathbf{f})^*$ in equation (5) requires us to sum out the hidden variable V . This fits the framework of the EM algorithm. It can be shown [8] that estimating $(A, \mathbf{f})^*$ in eqn. (5) is equivalent to minimizing the EM free energy function:

$$\begin{aligned} E(\hat{p}, (A, \mathbf{f})) &= -\sum_V \hat{p}(V) \log p(X, V|Y, (A, \mathbf{f})) - \log p(A, \mathbf{f}) + \sum_V \hat{p}(V) \log \hat{p}(V) \\ &= \sum_V \hat{p}(V) E_T(X, Y, V, (A, \mathbf{f})) + E_A(A) + E_f(\mathbf{f}) + \sum_V \hat{p}(V) \log \hat{p}(V). \end{aligned} \quad (6)$$

The EM free energy is minimized when $\hat{p}(V) = p(V|X, Y, A, \mathbf{f})$. The EM algorithm consists of two steps: (I) The E-step minimizes $E(\hat{p}, (A, \mathbf{f}))$ with respect to $\hat{p}(V)$ keeping (A, \mathbf{f}) fixed, (II) The M-step minimizes $E(\hat{p}, (A, \mathbf{f}))$ with respect to (A, \mathbf{f}) with $\hat{p}(V)$ fixed. But an advantage of the EM free energy is that any algorithm which decreases the free energy is guaranteed to converge to, at worst, a local minima [8]. Therefore we do not need to restrict ourselves to the standard E-step and M-step.

Chui and Rangarajan's free energy [5],

$$E(M, f) = \sum_{i=1}^N \sum_{a=1}^N m_{ai} \|x_i - f(v_a)\|^2 + \lambda \|Lf\|^2 + T \sum_{i=1}^N \sum_{a=1}^K m_{ai} \log m_{ai} - \zeta \sum_{i=1}^N \sum_{a=1}^K m_{ai} \quad (7)$$

can be obtained as a *mean field approximation* to the EM free energy. This requires assuming that $\hat{p}(V)$ can be approximated by a factorizable distribution $\prod_a P(v_a)$. The soft-assign variables $m_{ai} \in [0, 1]$ are related to $\hat{p}(V)$ by $m_{ai} = \hat{P}(v_a = i)$. An alternative approximation to the EM free energy can be done by using the Bethe-Kikuchi free energy [9].

Like Rangarajan *et al* [5,9] we will need to approximate $\hat{p}(V)$ in order to make the EM algorithm tractible. Our approximations will be motivated by informative features, see section (4), which will give a link to shape contexts [3] and feature-based algorithms.

4 Implementing the EM Algorithm

In this section, we introduce informative features and describe the implementation of the algorithm.

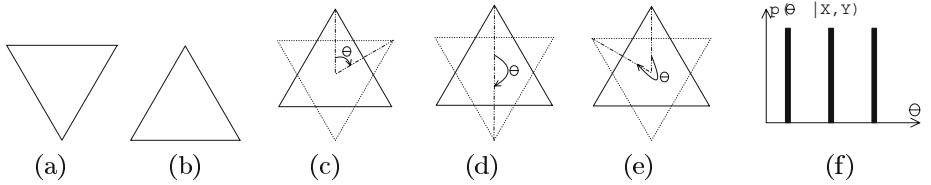


Fig. 2. The distribution $p(\theta|X, Y)$, shown in (f), has three modes for a target shape X , shown in (a), and a source shape Y , shown in (b). (c), (d), and (e) respectively display the three possible values for the θ .

4.1 Computing the Initial State

The EM algorithm is only guaranteed to converge to a local minima of the free energy. Thus, it is critical for the EM algorithm to start with the “right” initial state. Our preliminary experiments in shape matching suggested that the probability distribution for (A, \mathbf{f}) is strongly peaked and the probability mass is concentrated in small areas around $\{(A, \mathbf{f})^*, (A, \mathbf{f})_2^*, (A, \mathbf{f})_3^*, \dots\}$. Hence if we can make good initial estimates of (A, \mathbf{f}) , then EM has a good chance of converging to the global optimum.

The rotation angle θ is usually the most important part of (A, \mathbf{f}) to be estimated. (See Fig. 2 for an example where there are three equally likely choices for θ .) It would be best to get the initial estimate of θ from $p(\theta|X, Y)$, but this requires integrating out variables which is computationally too expensive. Instead, we seek to approximate $p(\theta|X, Y)$ (similar to the Hough Transform [2]) by an *informative feature distribution* $p_{IF}(\theta|X, Y)$:

$$p(\theta|X, Y) \approx p_{IF}(\theta|X, Y) = \sum_i \sum_a q(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) \delta(\theta - \theta(a, i, X, Y)), \quad (8)$$

where $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{y}_a)$ are *informative features* for point \mathbf{x}_i and y_a respectively, $q(\mathbf{x}_i, \mathbf{y}_a)$ is a *similarity measure* between the features, and $\theta(X, Y, a, i)$ is the angle if the i th point on X is matched with a th point on Y .

Next, we describe how to design the informative features $\phi(\mathbf{x}_i)$ and the similarity measures $q(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a))$.

4.2 Designing the Informative Features

The *informative features* are used to make computationally feasible approximations to the true probability distributions. They should observe two key properties to have

$$\int p(\theta|X, Y, (A_{-\theta}, \mathbf{f})) p(A_{-\theta}) p(\mathbf{f}) dA_{-\theta} d\mathbf{f} \approx p(\theta|\phi(X), \phi(Y))$$

(I) They should be “invariant” as possible to the transformations. Ideally $p(\theta|\phi(X), \phi(Y), (A_{-\theta}, f)) = p(\theta|\phi(X), \phi(Y))$.

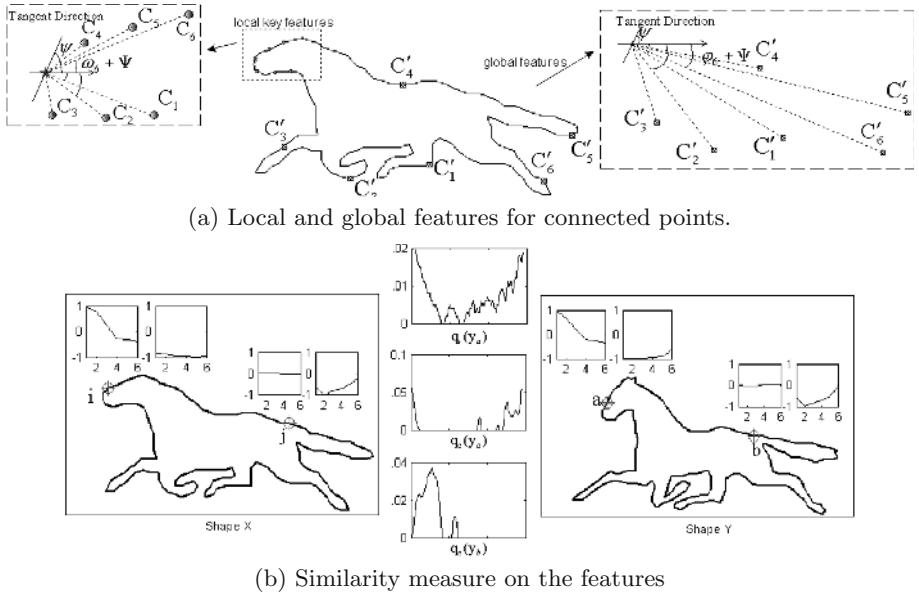


Fig. 3. Features and the similarity measure of the features. a) Illustrates how the local and global features are measured for connected points. In b), the features of two points in shape X and Y are displayed. The top figure in the middle of b). shows similarities between point a in Y w.r.t. all points in X using the shape context feature. The other two figures in the middle of b). are the similarities between points a and b in Y w.r.t. all points in X respectively. As we can see, similarities by features defined in this paper for connected points have lower entropy than those by shape contexts.

(II) They should be “representative”. For example, we would ideally have

$$\int p(\theta|X, Y, (A_{-\theta}, \mathbf{f}))p(A_{-\theta})p(\mathbf{f})dA_{-\theta}d\mathbf{f} = \int p(\theta|\phi(X), \phi(Y), (A_{-\theta}, \mathbf{f}))p(A_{-\theta})p(\mathbf{f})dA_{-\theta}d\mathbf{f}$$

where $A_{-\theta}$ is the components of A except for θ and $\phi(X), \phi(Y)$ are the feature vectors for all points in both images.

The two properties for informative features are also used to approximate distribution of other variables, for example, $p(V|X, Y)$, which requires us to integrate out (A, \mathbf{f}) and can be approximated by $p_{IF}(V|\phi(X), \phi(Y)) = \prod_a q(\phi(\mathbf{x}_{v_a}), \phi(\mathbf{y}_a))$.

In this paper we select the features $\phi(.)$ and measures $q(.,.)$ so as to obtain P_{IF} with *low-entropy*. This is a natural choice because it implies that the features have low matching ambiguity. We can evaluate this *low-entropy* criteria over our dataset for different choices of features and measures, see figure (3).

A better criteria, though harder to implement, is to select the features and measures which maximize the conditional Kullback-Leibler divergence evaluated over the distribution $p(X, Y)$ of problem instances:

$$\sum_{X,Y} p(X, Y) \sum_{(A,\mathbf{f})} p(V, (A, \mathbf{f}) | X, Y) \log \frac{p(V, (A, \mathbf{f}) | X, Y)}{p_{IF}(V, (A, \mathbf{f}) | \phi(X), \phi(Y))}, \quad (9)$$

but full evaluation of this criterion is a task for future work.

We used the low-entropy criterion to devise different sets of features for the two cases, shapes of *connected* points representation and shapes of *sparse* points representation.

Case I: The Connected Point Representation

We use **local** and **global** features illustrated by Fig.3. The local features at a point $\mathbf{x}(s_i)$ with tangent ψ_i are defined as follows. Choose six points on the curve by $(\mathbf{x}(s_i - 3ds), \mathbf{x}(s_i - 2ds), \mathbf{x}(s_i - ds), \mathbf{x}(s_i + ds), \mathbf{x}(s_i + 2ds), \mathbf{x}(s_i + 3ds))$, where ds is a (small) constant. The angles of these positions w.r.t. point \mathbf{x}_i are $\omega_j + \omega_j, j = 1..6$. The local features are $h_l(\mathbf{x}_i) = (\omega_j, j = 1..6)$. The global features are selected in a similar way. We choose six points near $\mathbf{x}(s_i)$, with tangent ψ_i , to be $(\mathbf{x}(s_i - 3\Delta s), \mathbf{x}(s_i - 2\Delta s), \mathbf{x}(s_i - \Delta s), \mathbf{x}(s_i + \Delta s), \mathbf{x}(s_i + 2\Delta s), \mathbf{x}(s_i + 3\Delta s))$, where Δs is a (large) constant, with angles $\psi_i + \varphi_j : j = 1, \dots, 6$. The global features are $h_g(\mathbf{x}_i) = (\varphi_j, j = 1..6)$. Observe that the features $\phi = (h_l, h_g)$ are invariant to rotations in the image plabe and also, to some extent, to local transformations.

In Fig. (3).b., for display purposes we plot sinusoids $(\sin(h_l), \sin(h_g))$ for two points on the X and two points on the Y . Observe the similarity between these features on the corresponding points.

The similarity measure between the two points is defined to be:

$$q_c(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) = 1 - c_1 \left(\sum_{j=1}^6 D_{angle}(\omega_j(x_i) - \omega_j(y_a)) + \sum_{j=1}^6 D_{angle}(\varphi_j(x_i) - \varphi_j(y_a)) \right),$$

where $D_{angle}(\omega_j(x_i) - \omega_j(y_a))$ is the minimal angle from $\omega_j(x_i)$ to $\omega_j(y_a)$, and c_1 is a normalization constant. The second and the third row in the middle of Fig. (3).b. respectively plot the vector $q_c(\mathbf{y}) = [q_c(\phi(\mathbf{x}_i), \phi(\mathbf{y})), i = 1..M]$ as a function of i for points \mathbf{y}_a and \mathbf{y}_b on Y .

Case II: The Sparse Point Representation

In this case, we also use **local** and **global** features. To obtain the local feature for point \mathbf{x}_i , we draw a circle with a (small) radius r and collect all the points that fall into the circle. The relative angles of these points w.r.t. \mathbf{x}_i and \mathbf{x}_i 's tangent angle are computed. The histogram of these angles is then used as the local feature, H_l .

The global feature for the sparse points is computed by shape contexts [3]. We denote it by H_g and the features become $\phi = (H_l, H_g)$.

The feature similarity between two points \mathbf{x}_i and y_a is measured by the χ^2 distance:

$$q_s(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) = 1 - c_2(\chi^2(H_l(\mathbf{x}_i), H_l(\mathbf{y}_a)) + \chi^2(H_g(\mathbf{x}_i), H_g(\mathbf{y}_a))).$$

The first row in the middle of Fig. (3).b. plots the vector

$$q_s(\mathbf{y}_a) = [q_s(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)), i = 1..M]$$

as a function of i for a point \mathbf{y}_a on Y .

The advantage of the sparse point representation is that it is very general and does not require a procedure to group points into contours. But for this very reason, the features and measures have higher entropy than those for the connected point representation. In particular, the global nature of the shape context features [3] means that these features and measures tend to have high entropy, see the Fig. (3)b., particularly shape context features are also of unnecessarily high dimension – consisting of 2D histograms with 60 bins – and better results, in terms of entropy, can be obtained with lower dimensional features.

4.3 The E Step: Approximating $\hat{p}(V)$

We can obtain an approximation $p_{IF}(\theta|X, Y)$ to $p(\theta|X, Y)$, see equation (8), using the informative features and similarity measures described in the previous section. We select each peak in $p_{IF}(\theta|X, Y)$ as an initial condition $\theta_{initial}$ for θ . The same approach is used to estimate the other variables in A and \mathbf{f} from $p(V|X, Y, \theta_{initial})$. We use similar informative features to those described in the previous section except that we replace ψ by $\theta_{initial}$

$$h'_l = (\omega'_j, j = 1..6) = (\alpha_j - \theta_{initial}, j = 1..6),$$

and

$$h'_g = (\varphi'_j, j = 1..6) = (\beta_j - \theta_{initial}, j = 1..6). \quad (10)$$

We also augment the similarity measure by including the scaled relative position of point \mathbf{x}_i to the center of the shape $\bar{\mathbf{x}} = \frac{1}{M} \sum_i \mathbf{x}_i$:

$$\begin{aligned} q'_c(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a)) &= 1 - c'_1 \sum_{j=1}^6 [D_{angle}(\omega'_j(x_i) - \omega'_j(y_a)) + D_{angle}(\varphi'_j(x_i) - \varphi'_j(y_a))] \\ &\quad - c'_2 \|\mathbf{x}_i - \bar{\mathbf{x}}, \mathbf{y}_a - \bar{\mathbf{y}}\|^2. \end{aligned}$$

Thus, we have the following approximation:

$$p(V|X, Y, \theta) \approx p_{IF}(V|X, Y, \theta) = \prod_a p_{if}(v_a|y_a, X, Y, \theta). \quad (11)$$

where

$$p_{if}(v_a = i|y_a, X, Y, \theta) \approx \frac{q'_c(\phi(\mathbf{x}_i), \phi(\mathbf{y}_a))}{\sum_{j=0}^M q'_c(\phi(\mathbf{x}_j), \phi(\mathbf{y}_a))}.$$

After the first iteration, we update the features and feature similarity measure by $q'_c(\phi(\mathbf{x}_i), \phi((A + \mathbf{f})(\mathbf{y}_a)))$ and use them to approximate $p(V|X, Y, (A, \mathbf{f}))$ as in eqn (11).

4.4 The M Step: Estimating A and \mathbf{f}

Once we have an approximation to $\hat{p}(V)$, we then need to estimate (A, \mathbf{f}) according to eqn. (6). We expand $E(\hat{p}, (A, \mathbf{f}))$ as a Taylor series in A, \mathbf{f} keeping the second order terms and then estimate (A, \mathbf{f}) by least squares.

5 Summary of the Algorithm

Our algorithm is performed by an approximation to the EM algorithm and it proceeds as follows:

1. Given a target shape X and a source shape Y , it computes their informative features described in section 4.2 and uses $P_{IF}(\theta|X, Y)$ (equation (8)) to obtain several possible rotation angles $\theta_{initial}$ s.
2. For each rotation angle $\theta_{initial}$, we obtain a new shape Y' by rotating it for $\theta_{initial}$.
3. Update features for shape Y' and estimate $p(V|X, Y, \theta)$ by $P_{IF}(V|XY, \theta)$ as eqn. (11).
4. Estimate (A, \mathbf{f}) from the EM equation by least-squares method.
5. Obtain the new shape Y' by the transformation function $Ay + \mathbf{f}(Y)$. Repeat step 3 for 4 iterations.
6. Compute the similarity measure and keep the best $(A, \mathbf{f})^*$, among all the initial $\theta_{initial}$ s and compute the metric according to eqns. (3) and (2). (We can also combine the results from several starting points to approximate eqn. 2. In practice, we found there is not much difference except for special cases like the equal lateral triangle.)

The algorithm runs at 0.2 seconds for matching X and Y of around 100 points. Note that our method does not need the target shape X and the source shape Y to have the same or nearly the same number of points, which is a key requirement for many matching algorithms.

6 Experiments

We tested our algorithm on a variety of data sets and some results are reported in this section. Fig.4 shows the running example where the source shape Y in (d) is matched with the target shape X . Fig.4.e and .f show the transformation A^* and \mathbf{f}^* estimated.

6.1 MPEG7 Shape Database

We first tested our algorithm on the MPEG7 CE-Shape-1 [7] which consists of 70 types of objects each of which has 20 different silhouette images (i.e. a total of 1400 silhouettes). Since the input images are binarized, we can extract contours and use the connected point representation. Fig.5.a displays 2 images for each type. The task is to do retrieval and the recognition rate is measured by “Bull’s eye” [7]. For every image in the database, we match it with every other image and keep the 40 best matched candidates. For each one of the other 19 of the same type, if it is in the selected 40 best matches, it is considered as a success. Observe that the silhouettes also include mirror transformations which our algorithm can take into account because the informative features are computed based on relative angles. The recognition rates for different algorithms are shown in table 1 [10] which shows that our algorithms outperforms the alternatives. The speed is in the same range as those of shape contexts [3] and curve edit distance [10].

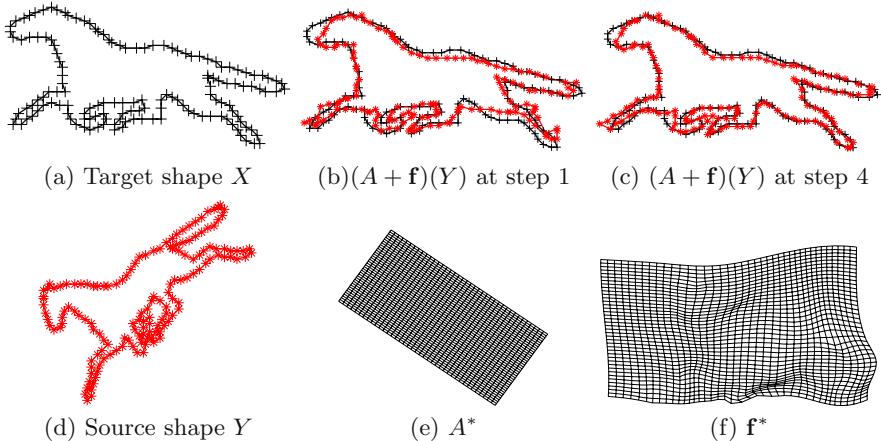


Fig. 4. Shape matching of the running example.

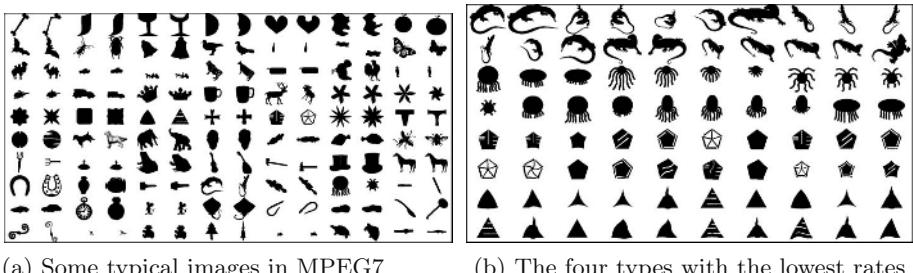


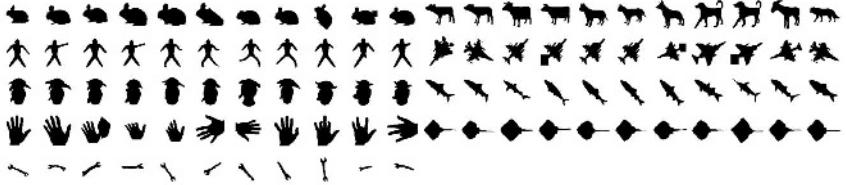
Fig. 5. Matching as image retrieval for the MPEG7 CE-Shape-1.

Table 1. The retrieval rates of different algorithms for the MPEG7 CE-Shape-1. Results by the other algorithms are from Sebastian et al. [10].

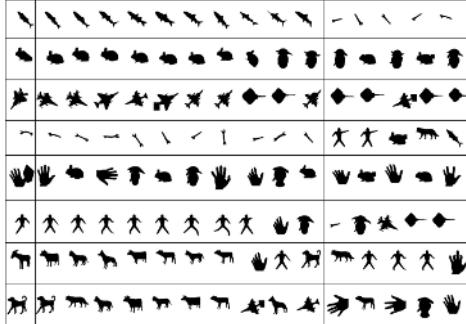
Algorithm	CSS	Visual Parts	Shape Contexts	Curve Edit Distance	Our Method
Recognition Rate	75.44%	76.45%	76.51%[3]	78.17% [10]	80.03%

6.2 The Kimia Data Set

We then tested the identical algorithm (i.e. connected point representation and same algorithm parameters) on the Kimia data set of 99 shapes [11], which are shown in Fig.6.a. For each shape, the 10 best matches are picked since there are 10 other images in the same category. Table 2 shows the numbers of correct matches. Our method performs similarly to Shock Edit [11] for the top 7 matches, but is worse for the top 8 to 10. Shape context performs less well than both algorithms on this task. Fig.6.b. displays the fifteen top matches for some shapes. Our relative failure, compared with Shock Edit, is due to the transformations which occur in the dataste, see the 8-10th examples for each object in figure (6), and which require more sophisticated representations and transformations than those used in this paper.



(a) The 99 silhouette images of the Kimia data set.



(b) Some matching results by our method

550	551	560	567	572	589	593	613	616	678	809	812	828	836	838
350	352	358	600	616	618	646	655	720	770	793	824	860	860	869
739	748	753	756	756	777	788	811	812	836	932	932	933	937	946
300	306	352	379	381	426	452	462	464	488	1036	1057	1092	1127	1132
322	507	572	574	578	589	649	649	704	911	939	942	955	956	957
209	255	268	268	273	276	289	299	334	650	679	697	714	714	
535	556	558	614	628	637	646	652	685	693	702	702	714	720	738
300	600	607	617	622	628	634	637	641	642	643	643	649	650	654

(c) Some matching results by Shock Edit

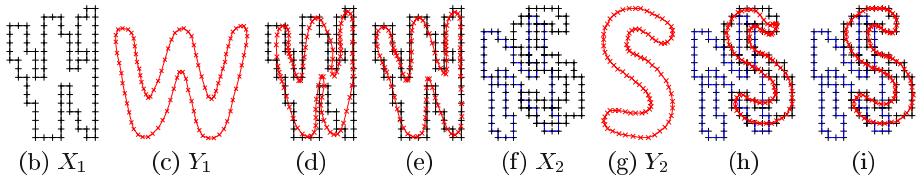
Fig. 6. The Kimia data set of 99 shapes and some matching results.

Table 2. Numbers of matched shapes by different algorithms. Results by the other algorithms are due to Sebastian et al. [11].

Algorithm	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10
Shock Edit	99	99	99	98	98	97	96	95	93	82
Our Method	99	97	99	98	96	96	94	83	75	48
Shape Contexts	97	91	88	85	84	77	75	66	56	37



(a) Some typical text images.

**Fig. 7.** Results on some text images. (e) and (i) display the matching. We purposely put two shapes together and find that the algorithm is robust in this case.

6.3 Text Image Matching

The algorithm was also tested on real images of text in which binarization was performed followed by boundary extraction. Some examples are shown in Fig.7. Similar results can be obtained by matching the model to edges in the image. Further tests on this dataset are ongoing.

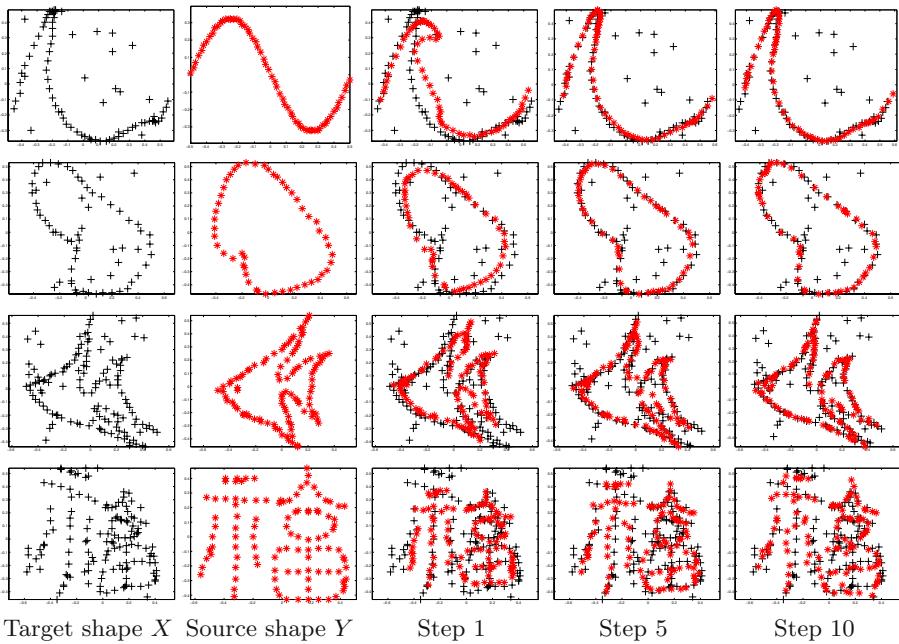


Fig. 8. Some results on Chui and Rangarajan data set.

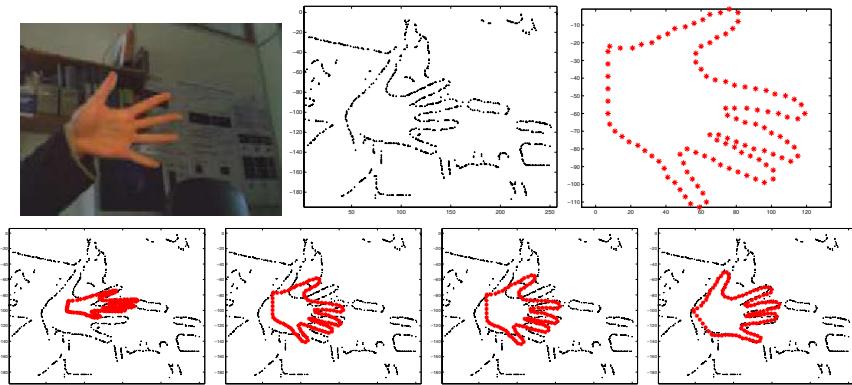


Fig. 9. Result on a hand image.

6.4 Chui and Rangarajan

To test our algorithm as a shape registration method, we also tried the data set used by Chui and Rangarajan [5]. We used the sparse point representation in this case. The algorithm runs for 10 steps and some results are shown in Fig.8. The quality of our results are similar to those reported in [5]. But our algorithm runs an estimated 20 times faster.

6.5 A Detection Task

Our algorithm can also be used for object detection where, unlike recognition, we do not know where the object is in the image. To illustrate this, we tested our algorithm on a hand image used in [14]. Edge points were extracted to act as the target shape and the source image was a hand represented by sparse points. The result is shown in Fig.9.

7 Discussion

This paper introduced a criterion for shape similarity and an algorithm for computing it. Our approach helps show relations between softassign [5] and shape contexts [3]. We formulated shape similarity by a generative model and used a modified variant of the EM algorithm for inference. A key element is the use of informative features to guide the algorithm to rapid and correct solutions. We illustrated our approach on datasets of binary and real images, and gave comparison to other methods. Our algorithm runs at speeds which are comparable to alternatives and is faster than others by orders of magnitude.

Our work is currently limited by the types of representations we used and the transformations we allow. For example, it would give poor results for shape composed of parts that can deform independently (e.g. human figures). For such objects, we would need representations based on symmetry axes such as skeletons [10] and parts [16]. Our current research is to extend our method to deal with such objects and to enable the algorithm to use input features other than edge maps and binary segmentations.

References

1. S. Abbasi and F. Mokhtarian, “Robustness of Shape Similarity Retrieval under Affine”, *Proc. of Challenge of Image Retrieval*, 1999
2. D. H. Ballard, “Generalizing the Hough Transform to Detect Arbitrary Shapes”, *Pattern Recognition*, vol. 13, no. 2, 1981.
3. S. Belongie, J. Malik, and J. Puzicha, “Shape Matching and Object Recognition Using Shape Contexts”, *IEEE Trans. on PAMI*, vol. 24, no. 24, 2002.
4. F. L. Bookstein, “Principal Warps: Thin-Plate Splines and the Decomposition of Deformations”, *IEEE Trans. on PAMI*, vol. 11, no. 6, 1989.
5. H. Chui and A. Rangarajan, “A New Point Matching Algorithm for Non-rigid Registration”, *Computer Vision and Image Understanding*, March, 2003.
6. U. Grenander, “General Pattern Theory: A Mathematical Study of Regular Structures”, *Oxford*, 1994.
7. L. J. Latechi, R. Lakamper, and U. Eckhardt, “Shape Descriptors for Non-rigid Shapes with a Single Closed Contour”, *Proc. of CVPR*, 2000.
8. R. Neal and G. E. Hinton, “A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants”, *Learning in Graphical Models*, 1998.
9. A. Rangarajan, J.M. Coughlan, A.L. Yuille, “A Bayesian Network for Relational Shape Matching”, *Proc. of ICCV*, Nice, France, 2003.

10. T. B. Sebastian, P. N. Klein, and B. B. Kimia, “On Aligning Curves”, *IEEE Trans. on PAMI*, vol. 25, no. 1, 2003.
11. T. B. Sebastian, P. N. Klein, and B. B. Kimia, “Recognition of Shapes by Editing their Shock Graphs”, *accepted by IEEE Trans. on PAMI*, 2003.
12. A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla, “Shape Context and Chamfer Matching in Cluttered Scenes”, *CVPR*, 2003.
13. Z. Tu, X. Chen, A. Yuille, and S.C. Zhu, “Image Parsing: Unifying Segmentation, Detection and Recognition”, *Proc. of ICCV*, Nice, France, 2003.
14. R. C. Veltkamp and M. Hagedoorn, “State of the Art in Shape Matching”, Technical Report UU-CS-1999-27, Utrecht, 1999.
15. A. L. Yuille and N. M. Grzywacz, “A Computational Theory for the Perception of Coherent Visual Motion”, *Nature*, vo. 333, no. 6168, 1988.
16. S. C. Zhu and A. L. Yuille, “FORMS: A Flexible Object Recognition and Modeling System”, *IJCV*, vol.20, no.3, 1996.

Generalized Histogram: Empirical Optimization of Low Dimensional Features for Image Matching

Shin'ichi Satoh

National Institute of Informatics, Tokyo 101-8430, Japan,
satoh@nii.ac.jp,
<http://research.nii.ac.jp/~satoh/>

Abstract. We propose *Generalized Histogram* as low-dimensional representation of an image for efficient and precise image matching. Multiplicity detection of videos in broadcast video archives is getting important for many video-based applications including commercial film identification, unsupervised video parsing and structuring, and robust highlight shot detection. This inherently requires efficient and precise image matching among extremely huge number of images. Histogram-based image similarity search and matching is known to be effective, and its enhancement techniques such as adaptive binning, subregion histogram, and adaptive weighting have been studied. We show that these techniques can be represented as linear conversion of high-dimensional primitive histograms and can be integrated into generalized histograms. A linear learning method to obtain generalized histograms from sample sets is presented with a sample expansion technique to circumvent the overfitting problem due to high-dimensionality and insufficient sample size. The generalized histogram takes advantage of these techniques, and achieves more than 90% precision and recall with 16-D generalized histogram compared to the ground truth computed by normalized cross correlation. The practical importance of the work is revealed by successful matching performance with 20,000 frame images obtained from actual broadcast videos.

1 Introduction

Recent advance in broadband networks and digital television broadcasting enables huge-scale image and video archives in the WWW space and broadcast video streams. In these huge-scale image and video archives, multiplicity detection is getting important. For example, Cheung et al. [1] proposed a multiplicity detection method for video segments in the WWW to detect possible unlawful copy or tampering of videos. On the other hand, some researchers notice that multiplicity detection is especially useful for broadcast video archives [2,3]. In this paper, by multiplicity detection for video archives, we assume that pairs of distinctively similar (or almost identical) video segments can be extracted from videos; i.e., they are originated from the same video materials, but with different transmission conditions or different post production effects, e.g., video captions, scale, shift, clip, etc. Figure 1 shows typical examples of multiplicity in video archives. They include (a) commercial films, (b) opening/ending of programs, (c) opening/ending of some types of scenes, (d) video shots of an important event, etc. Detection of (a) is particularly important for a company that needs to check if its commercial films are

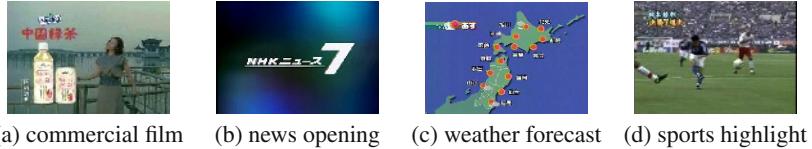


Fig. 1. Examples of identical video segments

properly broadcasted. Video segments (b) and (c) themselves are not very important, however, since they represent “punctuation” of video streams, they are useful for video parsing and structuring; i.e., identification of particular programs or locating breaks between news stories. Multiplicity detection may realize unsupervised video parsing (especially for news videos) without any a priori knowledge on visual features of the punctuation shots [4]. Highlight shots of important events such as Olympic games or world cup soccer games including (d) tend to be repeated; for instance, highlight shots in on-the-spot broadcasting of sports may repeatedly used in several news or sports news as “play of the day.” Thus multiplicity detection may help to detect highlight shots.

Despite its importance, since identical video segment pairs are essentially very rare, multiplicity detection could become meaningless unless it is applied to huge-scale video archives. Therefore precise matching between images is required; otherwise, desired identical video segment pairs may be buried under large number of mistakenly detected pairs. While at the same time, the method should allow slight modification to images, so that it is required to ignore small video captions, slight clipping, etc. In order to satisfy the contradictory requirements, pixelwise comparison is needed, however, is very time consuming. In order to apply to huge-scale video archives, severe speedup should be addressed. Compact (low-dimensional) representation of images is necessary for efficient image matching, while preserving precise matching performance.

In this paper, we propose generalized histogram as low-dimensional representation of an image which realizes precise yet at the same time efficient image matching. The generalized histogram is histogram-based low-dimensional representation; thus it achieves very efficient matching process. Several techniques have been studied for enhanced histogram representation, including adaptive binning, subregion histogram, and adaptive weighting. We show that the generalized histogram takes advantage of these techniques at the same time. Successful results are shown by detecting multiplicity in actual video footages taken from television broadcast to reveal effectiveness of the generalized histograms.

2 Multiplicity Detection in Large Collection of Images

As described, multiplicity detection in video archives should be applied to huge-scale video archives. Since video archives obviously contain huge amount of frame images, the method should discover identical image pairs among huge number of images. Strong correlation between consecutive frames in a video allows most frames to be skipped in a certain period. However, in order not to miss many identical video segments with dynamic motion, the period should be short enough (ten frames or so). Consequently, discovery of identical image pairs between large image sets is an inevitable key technology.

Assume that two sets of images $IS^1 = \{I_i^1\}$ and $IS^2 = \{I_j^2\}$ are given. We use normalized cross correlation (NCC) for the similarity check here using a threshold $-1 \ll \theta_h < 1$. The resultant identical image pair set $IIPS$ is thus:

$$IIPS = \{(I^1, I^2) \mid I^1 \in IS^1, I^2 \in IS^2, NCC(I^1, I^2) > \theta_h\}$$

$$NCC(I^1, I^2) = \frac{\sum_{x,y} (I^1 - \bar{I}^1)(I^2 - \bar{I}^2)}{\{\sum_{x,y} (I^1 - \bar{I}^1)^2 \sum_{x,y} (I^2 - \bar{I}^2)^2\}^{1/2}}$$

where, for notational convenience, I^1 and I^2 represent intensity value of each image at the location (x, y) , and \bar{I} is its mean. For simplicity, we assume that all images in collections are monochrome, but extension to color images can easily be achieved. It obviously is intractable to calculate NCC for every combination of images. Our implementation takes about 16ms on an ordinary PC to compute NCC between two 352×240 images. For example, to extract $IIPS$ from two sets of 100,000 images, it will take $16\text{ms} \times 100,000 \times 100,000 \simeq$ five years. Since even one hour-long video (30fps) contains about 100,000 frames, intensive speedup is required.

In order to address efficient search of database which requires costly similarity (or distance) calculation, a “dirty filtering” technique is sometimes used [5]. In this, each element in the database is first converted into a low-dimensional (ten to twenty dimensions) vector by some function f . Benefit of dimensionality reduction is two fold: Firstly, metric calculation becomes much lighter, because the calculation is basically proportional to the number of dimension (352×240 versus $10 \sim 20$). Secondly, data of less than 100 dimensions are very suitable to the high-dimensional indexing techniques using tree-type structures and/or hash [6], which further accelerate the search speed. Then approximation of $IIPS$ is obtained as follows:

$$II\tilde{PS} = \{(I^1, I^2) \mid I^1 \in IS^1, I^2 \in IS^2, d(f(I^1), f(I^2)) < \theta_d\}$$

where $d(\cdot)$ represents the distance between vectors, $II\tilde{PS}$ is the approximation of $IIPS$, and θ_d is a threshold value for the distance. Finally, each element of $II\tilde{PS}$ is “cleansed” by the original metric (in our case, NCC). “The lower-bound condition,” i.e., $IIPS \subseteq II\tilde{PS}$, ensures the dirty filtering will not cause false dismissals. However, due to discrepancy between the original metric and the metric in the converted low-dimensional space, this condition tends to be difficult to hold. If we somehow achieve $IIPS \subseteq II\tilde{PS}$, $II\tilde{PS}$ may become unexpectedly large, thus cleansing may take impractically long. Instead we allow $IIPS \not\subseteq II\tilde{PS}$, but we keep $IIPS - II\tilde{PS}$ (false dismissals) as small as possible. While, since the size of $II\tilde{PS}$ reflects the cleansing cost, i.e., the number of NCC calculation, this is also preferably small.

To evaluate these, precision and recall are suitable: $\text{precision} = |IIPS \cap II\tilde{PS}| / |II\tilde{PS}|$, $\text{recall} = |IIPS \cap II\tilde{PS}| / |IIPS|$. In order to make $II\tilde{PS}$ small, precision should be larger and close to one. On the other hand, to make $IIPS - II\tilde{PS}$ small, since $1 - \text{recall} = |IIPS - II\tilde{PS}| / |IIPS|$, recall should also be larger and close to one. In evaluation, we set another threshold θ_l , $\theta_l < \theta_h$, to employ rejection range; image pairs having NCC values between θ_l and θ_h are excluded and will not be used for the evaluation.

3 Generalized Histogram

3.1 Histogram Composition

Histograms are sometimes used as low-dimensional representation of images, especially for image similarity matching [9,7] and multiplicity detection [1,8]. Assume that an image I is composed of M pixels, and its intensity value at the location (x, y) is $v = v(x, y)$. Then the image can be regarded as the scatter of M pixels in the three dimensional space (x, y, v) . To compute a histogram of the image, we first need to tessellate the (x, y, v) space into N non-overlapping regions $R_i, i = 1, \dots, N$. Then an N -bin histogram ($H = [h_1 \ h_2 \ \dots \ h_N]^T$ in vector notation) can be computed by counting pixels which fall into each region: $h_i = |\{(x, y, v(x, y)) \mid (x, y, v(x, y)) \text{ in } R_i\}|$. For example, if we divide the range of intensity value $[v_l, v_h]$ into N regions at the points: $v_i = v_l + \frac{|v_h - v_l|}{N}i$ where $i = 0, \dots, N$, by defining the tessellation as $R_i = \{(x, y, v) \mid v_{i-1} \leq v < v_i\}$, we will obtain N -bin global intensity histograms. Subregion histograms can be obtained by the tessellation $R_{i,j,k} = \{(x, y, v) \mid x_{i-1} \leq x < x_i, y_{j-1} \leq y < y_j, v_{k-1} \leq v < v_k\}$ where x_i, y_j , and v_k are dividing points of the range of x, y , and v respectively. Resultant histogram $h_{i,j,k}$ should then be reordered into a linear list by a particular ordering such as the lexicographic order.

Obviously, choice of tessellation of the (x, y, v) space affects the performance of image similarity evaluation and matching. For subregion division, researchers employ regular division such as 1×1 (global histograms), 2×2 , 3×3 , etc., as well as irregular division such as using one at the center and four subregions in the peripheral region [10]. For tessellation of the range of intensity, or tessellation of the color space in most cases, adaptive binning techniques have been studied [11,12]. In the adaptive binning, tessellation is determined according to the actual distribution of pixels, and thus the tessellation would become fine in dense region, while rough in coarse region. If tessellation is independently determined for each image, resultant histograms better reflect the actual distribution of pixels. We call this dynamic adaptive binning. However, since bins of histograms of the different images do not necessarily correspond, special metrics such as the Earth Mover's Distance [11] or weighted correlation [12] should be used, which are computationally costly than the simple Euclidean distance. Thus the dynamic adaptive binning would not fit to the purpose of dirty filtering. Instead, we can determine tessellation based on the distribution of pixels for all images in image collections. This makes the tessellation unique for all the images, therefore ordinary metrics including the Euclidean distance can be used. We call this static adaptive binning. We will consider static adaptive binning instead of dynamic adaptive binning in this paper.

After histograms are obtained from images, we then need to evaluate distances between histograms. In evaluating distance between two histograms $H^1 = [h_1^1 \ \dots \ h_N^1]^T$ and $H^2 = [h_1^2 \ \dots \ h_N^2]^T$, the Minkowski distance is sometimes used: $d(H^1, H^2) = (\sum_i ||h_i^1 - h_i^2||^p)^{\frac{1}{p}}$. Especially, the Manhattan distance ($p = 1$) and the Euclidean distance ($p = 2$) are frequently used. As a variant of this, the weighted Minkowski distance is also used: $d(H^1, H^2) = (\sum_i w_i ||h_i^1 - h_i^2||^p)^{\frac{1}{p}}$ where w_i are weight coefficients. An adaptive weighting technique optimizes coefficients adaptively, mainly according to users' preference by relevance feedback in image retrieval systems [13]. Other metrics such as

quadratic metric [14] and Mahalanobis distance are also used. We mainly concentrate on the Euclidean distance in this paper.

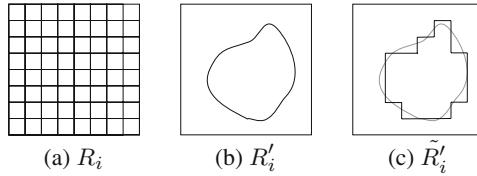


Fig. 2. Tessellation Approximation by Primitive Histogram

3.2 Histogram Conversion

If we have fine-grain tessellation, any tessellation can be approximated by aggregating the fine-grain tessellation. Based on this idea, we will show that histograms using adaptive binning and various subregion can be realized by linear conversion of fine-grain histograms. Assume that regions $R_i, i = 1, \dots, N$ compose fine-grain tessellation, i.e., the size of each R_i is small enough, and a histogram $H = [h_1 \dots h_N]^T$ is calculated by the tessellation. The tessellation could be regular, i.e., $8 \times 8 \times 16$ for the range of x, y , and v respectively, which generates 1024-bin histograms. We call the fine-grain histograms as primitive histograms. Then any tessellation $R'_i, i = 1, \dots, N'$ can be approximated by R_i as follows:

$$\tilde{R}'_i = \bigcup_j \{R_j \mid \forall k |R'_i \cap R_j| \geq |R'_k \cap R_j|\}$$

where \tilde{R}'_i is an approximation of R'_i and $|\cdot|$ represents the size of the region. Intuitively, R'_i is approximated by the union of R_j which *belong* to R'_i (see Fig. 2). Thus a histogram $H' = [h'_1 \dots h'_{N'}]^T$ based on the tessellation \tilde{R}'_i is calculated using N' by N matrix A as follows:

$$H' = AH$$

$$(A)_{i,j} = \begin{cases} 1 & \text{if } \tilde{R}'_i \cap R_j \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

We call the matrix A as aggregation matrix. By this conversion, any variation of binning and subregion can be realized as a simple linear conversion at the cost of approximation error. The error can become arbitrarily small by using finer tessellation for primitive histogram calculation.

Weighted distance between histograms can also be realized by linear conversion of histograms. The weighted Euclidean distance between histograms $H^1 = [h_1^1 \dots h_N^1]^T$ and $H^2 = [h_1^2 \dots h_N^2]^T$ can be calculated as:

$$d^2(H^1, H^2) = \sum_i w_i^2 ||h_i^1 - h_i^2||^2$$

where w_i are weight coefficients. This can be achieved by simply calculating the Euclidean distance of weighted histograms: $H' = [w_1 h_1 \dots w_N h_N]^T$. Thus weighted distance can also be realized by linear histogram conversion:

$$\begin{aligned} H' &= WH \\ W &= \text{diag}(w_1, w_2, \dots, w_N). \end{aligned}$$

Obviously these two conversions have the same linear form. Thus they can be combined easily, e.g., $H' = AWH$. We now allow the conversion matrix to be a general matrix G , i.e., $H' = GH$, which converts N -bin histogram H to N' -bin histogram H' . The converted histogram can represent not only various binning and subregion histogram plus weighted histogram, but also much general adaptable histogram. We call this histogram as generalized histogram, and G as generalized histogram generator matrix, or generator matrix in short. By properly designing the generator matrix, generalized histograms can take advantage of adaptive binning, adaptive subregion, adaptive weighting, and possibly much flexible adaptability.

3.3 Linear Learning of Generalized Histogram Generator Matrix

Learning generator matrix from training samples would be an ideal method because it could adapt to actual distribution of images. As training sets, we assume that positive samples S^+ and negative samples S^- are given:

$$\begin{aligned} S^+ &= \{(I^1, I^2) \mid \text{NCC}(I^1, I^2) > \theta_h\} \\ S^- &= \{(I^1, I^2) \mid \text{NCC}(I^1, I^2) < \theta_l\}. \end{aligned}$$

Let $\mathcal{H}(I)$ be the primitive histogram of the image I . The Euclidean distance between generalized histograms of images I^1 and I^2 is:

$$d^2(I^1, I^2) = \|G\mathcal{H}(I^1) - G\mathcal{H}(I^2)\|^2 = \|G[\mathcal{H}(I^1) - \mathcal{H}(I^2)]\|^2.$$

Ideal generator matrix G should preferably make distances between image pairs of positive samples smaller, while at the same time distances between image pairs of negative samples larger. We can achieve this by a similar technique to well-known Multiple Discriminant Analysis [15] (MDA for short). Let covariance matrices of histogram differences of image pairs of positive and negative samples, C^+ and C^- resp., be:

$$C^{+/-} = \sum_{(I^1, I^2) \in S^{+/-}} (\mathcal{H}(I^1) - \mathcal{H}(I^2))(\mathcal{H}(I^1) - \mathcal{H}(I^2))^T.$$

Since GCG^T represents a scatter matrix of vectors $G[\mathcal{H}(I^1) - \mathcal{H}(I^2)]$, an ideal generator matrix G minimizes the following criterion:

$$J(G) = \frac{\det(GC^+G^T)}{\det(GC^-G^T)}$$

because $\det(GCG^T)$ is proportional to the square of the hyperellipsoidal scattering volume [15]. $J(G)$ can be minimized by solving the following generalized eigenvalue problem: $C^-\Phi = C^+\Phi\Lambda$ where $\Phi = [\phi_1 \phi_2 \dots \phi_N]$ is eigenvector matrix and Λ is diagonal matrix of eigenvalues. Thus N' by N optimal generalized histogram generator matrix G is obtained by N' eigenvectors corresponding to N' largest eigenvalues, i.e., $G = [\phi_1 \phi_2 \dots \phi_{N'}]^T$.

3.4 Learnability of Histogram Adaptation

It is still unclear whether the presented algorithm produces similar effect to histogram adaptation techniques. We will discuss on this here.

Assume that the algorithm determines generator matrix $G = [\phi_1 \dots \phi_{N'}]^T$ which converts primitive histogram $H = [h_1 \dots h_N]^T$ into generalized histogram $H' = [h'_1 \dots h'_{N'}]^T$, i.e., $H' = GH$. The algorithm determines the plane spanned by $\phi_1, \dots, \phi_{N'}$, and generalized histogram is the projection of primitive histogram onto this plane. If there is large variance in the h_i direction in positive samples S^+ , i.e., variance of $h_i^1 - h_i^2$ is large where $(I^1, I^2) \in S^+$ and $\mathcal{H}(I^1) = [\dots h_i^1 \dots]$, $\mathcal{H}(I^2) = [\dots h_i^2 \dots]$, especially if this is larger than that in the h_j direction, then ideal adaptive weighting should provide larger weight to h_j direction than to h_i . This is because h_j is stationary in the positive samples and thus useful for discrimination, but h_i is unstable and should be suppressed. MDA has the similar effect since it determines projection direction so that to suppress large variance in the original scatter of positive samples. Thus the proposed algorithm has similar effect to adaptive weighting.

In ideal adaptive binning, two bins are preferably merged (in most cases neighboring bins) if they are strongly correlated in training samples. When we think of subregion histograms, two blocks should be merged into one block if these two have correlated intensity between image pairs in samples. Two blocks have correlated intensity when corresponding bins between the blocks are correlated. Thus adaptive binning, as well as adaptive subregion, may be realized by merging correlated elements. Assume that the direction h_p and h_q are correlated. Then MDA will produce projection direction which is close to parallel to the plane spanned by h_p and h_q , resulting one direction in generalized histogram is made close to $w_p h_p + w_q h_q$, while the other independent direction of the form $w'_p h_p + w'_q h_q$ is suppressed. Weights w_p and w_q will be determined according to the distribution in these directions, similarly to linear regression analysis. This obviously includes the case $w_p = w_q$, which has the same effect as aggregation matrix. The case when more than three directions are correlated can also be treated similarly.

3.5 Sample Expansion

In order to obtain effective generalized histograms, finer tessellation for primitive histograms is necessary, and thus high-dimensional vectors should be handled as primitive histograms. This makes estimation of generator matrix harder, because the estimation requires optimization of low-dimensional projection in very high-dimensional space. Generally, since the insufficient number of samples for learning in high-dimensional space may cause performance deterioration due to overfitting, learning in high-dimensional space requires more training samples (the preferable number of samples is exponential to the number of dimension [16]). However, it could be difficult to prepare the sufficient number of samples; especially for positive samples, by the nature of rareness of identical image pairs.

In order to circumvent the problem, we expand the number of samples by virtually adding perturbation to intensity value of each pixel. We call this technique sample expansion. Assume that we have a pair of images I^1 and I^2 from sample set (either positive or negative), and corresponding primitive histograms $H^1 = [h_1^1 h_2^1 \dots h_N^1]$ and

$H^2 = [h_1^2 \ h_2^2 \ \dots \ h_N^2]$. For simplicity, we think of intensity tessellation only. Covariance matrix of primitive histogram differences of image pairs can be obtained by summing the following elementary covariance matrix for all image pairs:

$$E = (H^1 - H^2)(H^1 - H^2)^T.$$

Basic idea of the sample expansion is to derive expectation of E with probabilistically adding perturbation to intensity values of pixels, and thus to derive expectation of covariance matrix of primitive histogram differences. We assume that only one pixel of either I^1 or I^2 will suffer from perturbation with probability P . If we virtually add perturbation to a pixel such that to make the pixel belong to the neighbor of its original bin, and if the pixel originally belongs to h_i , the value of h_i will decrease and the value of its neighbor bin (h_{i-1} or h_{i+1} , depending on the sign of the perturbation) will increase. Thus the primitive histogram becomes

$$\begin{aligned} H_{i,+} &= [h_1 \ \dots \ h_{i-1} \ h_{i-1} \ h_{i+1}+1 \ \dots \ h_N]^T \\ H_{i,-} &= [h_1 \ \dots \ h_{i-1}+1 \ h_{i-1} \ h_{i+1} \ \dots \ h_N]^T \end{aligned}$$

depending on the sign of the perturbation. The probability that the case $H_{i,+}$ occurs is assumed to be equal to the probability of $H_{i,-}$, and is the probability that a pixel in I falls in h_i (we set this to p_i) times $\frac{1}{2}P$. If we assume the distribution of pixel intensity is identical for all images, the probability that a pixel in any image falls in R_i is p_i . If we let $p(v)$ be pdf of intensity value, p_i is derived as follows:

$$p_i = \int_{R_i} p(v) dx dy dv. \quad (1)$$

In the case of $H_{i,+}$ or $H_{i,-}$ for I^1 , i.e., $H_{i,\pm}^1$, the elementary covariance matrix is:

$$\begin{aligned} E &= E|H_{i,\pm}^1 = (H_{i,\pm}^1 - H^2)(H_{i,\pm}^1 - H^2)^T \\ (E)_{i,i} &= (\Delta h_i)^2 \mp 2\Delta h_i + 1 \\ (E)_{i,i+1} &= (E)_{i+1,i} = \Delta h_i \Delta h_{i+1} \pm \Delta h_i \mp \Delta h_{i+1} - 1 \\ (E)_{i+1,i+1} &= (\Delta h_{i+1})^2 \pm 2\Delta h_{i+1} + 1 \\ (E)_{i,j} &= (E)_{j,i} = \Delta h_i \Delta h_j \mp \Delta h_j \\ (E)_{i+1,j} &= (E)_{j,i+1} = \Delta h_{i-1} \Delta h_j \pm \Delta h_j \\ (E)_{j,k} &= \Delta h_j \Delta h_k \end{aligned}$$

where $\Delta h_i = h_i^1 - h_i^2$, and $j, k \neq i, i+1$. We can then integrate the equations above to derive the expectation of the elementary covariance matrix:

$$\begin{aligned} E\{E\} &= \sum_i \sum_{s \in \{+,-\}} \frac{E|H_{i,s}^1 + E|H_{i,s}^2}{2} P p_i \\ (\Delta E)_{i,j} &\stackrel{\text{def}}{=} E\{(E)_{i,j}\} - \Delta h_i \Delta h_j \\ (\Delta E)_{1,1} &= P(\frac{1}{2}p_1 + \frac{1}{2}p_2) \end{aligned}$$

$$\begin{aligned}
(\Delta E)_{i,i} &= P\left(\frac{1}{2}p_{i-1} + p_i + \frac{1}{2}p_{i+1}\right) \quad (i \neq 1, N) \\
(\Delta E)_{N,N} &= P\left(\frac{1}{2}p_{N-1} + \frac{1}{2}p_N\right) \\
(\Delta E)_{i-1,i} &= (\Delta E)_{i,i-1} = -P\left(\frac{1}{2}p_{i-1} + \frac{1}{2}p_i\right) \\
(\Delta E)_{i,j} &= 0 \quad (otherwise)
\end{aligned}$$

(note that terms which have signs affected by the sign of perturbation are canceled), where $E\{\cdot\}$ provides expectation. We can then derive expectation of covariance matrices as follows:

$$\begin{aligned}
E\{C^+\} &= C^+ + |S^+| \Delta E \\
E\{C^-\} &= C^- + |S^-| \Delta E.
\end{aligned}$$

This modification still keeps the covariance matrices positive semi-definite. In the above derivation, we think of intensity tessellation only. However, extension to joint tessellation of intensity and location is quite straightforward. Resultant expectation of covariance matrices is then used for calculation of generalized histogram generator matrix.

4 Experiments

We apply the proposed method to images obtained from broadcast video archives. We select four video footages, one hour long each, broadcasted in the same time slot, at the same channel, but on the different days. 10,000 randomly chosen images are then extracted from them to compose four image sets, namely, A, B, C, and D. Since they are taken from the same time slot, they include the same video programs (actually the slot includes news and documentary), and thus each image set is expected to have similar distribution in the image space, or at least they include some shots, identical each other, such as opening and ending shots. Two sets (A and B) are used as training samples, and the other two (C and D) are used for test. From the training set, the positive sample set is composed of pairs of images from two image sets having larger NCC value than θ_h . The negative sample set is composed of 10,000 randomly chosen pairs of images having smaller NCC than θ_l . From the test set, the positive sample set is generated similarly, but the negative sample set is composed of all pairs having smaller NCC than θ_l . We use $\theta_h = 0.9$ and $\theta_l = 0.8$ in our experiments. The resultant positive and negative test sets are used as the ground truth for precision-recall evaluation. The size of positive sample sets is 2124 and 3278, while the size of negative sample sets is 10,000 and approximately $10,000 \times 10,000$ for the training and test sets respectively.

Primitive histograms are then calculated for images in the sets. Since we use NCC as the image matching criteria, intensity values of pixels v are first normalized by the mean m and the standard deviation σ into normalized intensity, i.e., $v' = (v - m)/\sigma$. To compute primitive histograms, we use regular tessellation $8 \times 8 \times 16$ in the (x, y, v') space. For the normalized intensity, we regularly divide the range $[-2, 2]$ since this range covers the most part of the distribution (95%) for the Gaussian distribution. We call the

histograms normalized intensity histograms. In this case, since normalized intensity values yield normalized Gaussian, Eq. (1) particularly becomes:

$$p_i = \int_{R_i} \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} dx dy dv. \quad (2)$$

We use this for p_i in calculating generalized histograms.

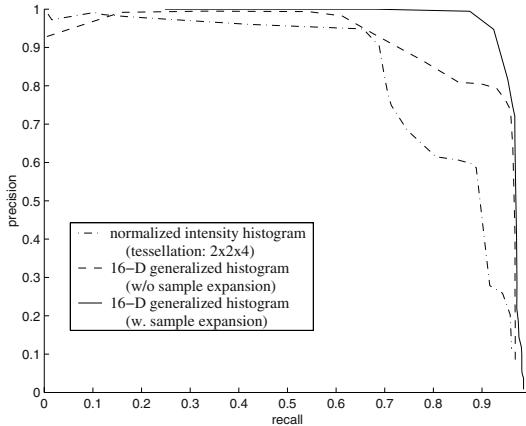


Fig. 3. Comparison of matching performance

We thus employ $8 \times 8 \times 16 = 1024$ -bin normalized intensity histograms as primitive histograms. By using training sets, a generator matrix is generated. Then 16-D generalized histograms are generated from test sets, and the image matching performance is evaluated in terms of precision-recall. For comparison, we generated 16-D normalized intensity histograms with regular tessellation $2 \times 2 \times 4$ in the (x, y, v') space, and the filtering performance is evaluated using the Euclidean distance. We have investigated several combinations of regular tessellation and evaluated 8- to 64-D normalized intensity histograms, to find that the tessellation used here performs the best [17]. Figure 3 shows the resultant precision-recall graphs comparing the matching performance of the normalized intensity histograms, the generalized histograms without sample expansion, and the generalized histograms with sample expansion. This clearly depicts that the generalized histograms obviously outperform ordinary histograms. Moreover, the sample expansion technique drastically boosts the performance. The final performance exceeds 90% recall and 90% precision at the same time, or even at 95% recall more than 80% precision is achieved, which is very satisfactory. In the evaluation, we changed the perturbation probability P from 0.0001 to 0.01 and found that there was no effect on the matching performance, and thus confirmed that P is very insensitive to the performance. To inspect the dependence of the performance to training set, we conduct another experiment by swapping training and test sets, i.e., image sets C and D for training and A and B for test. The performance is almost the same as Fig. 3, and thus the proposed method is thought to be relatively insensitive to the change of training set.

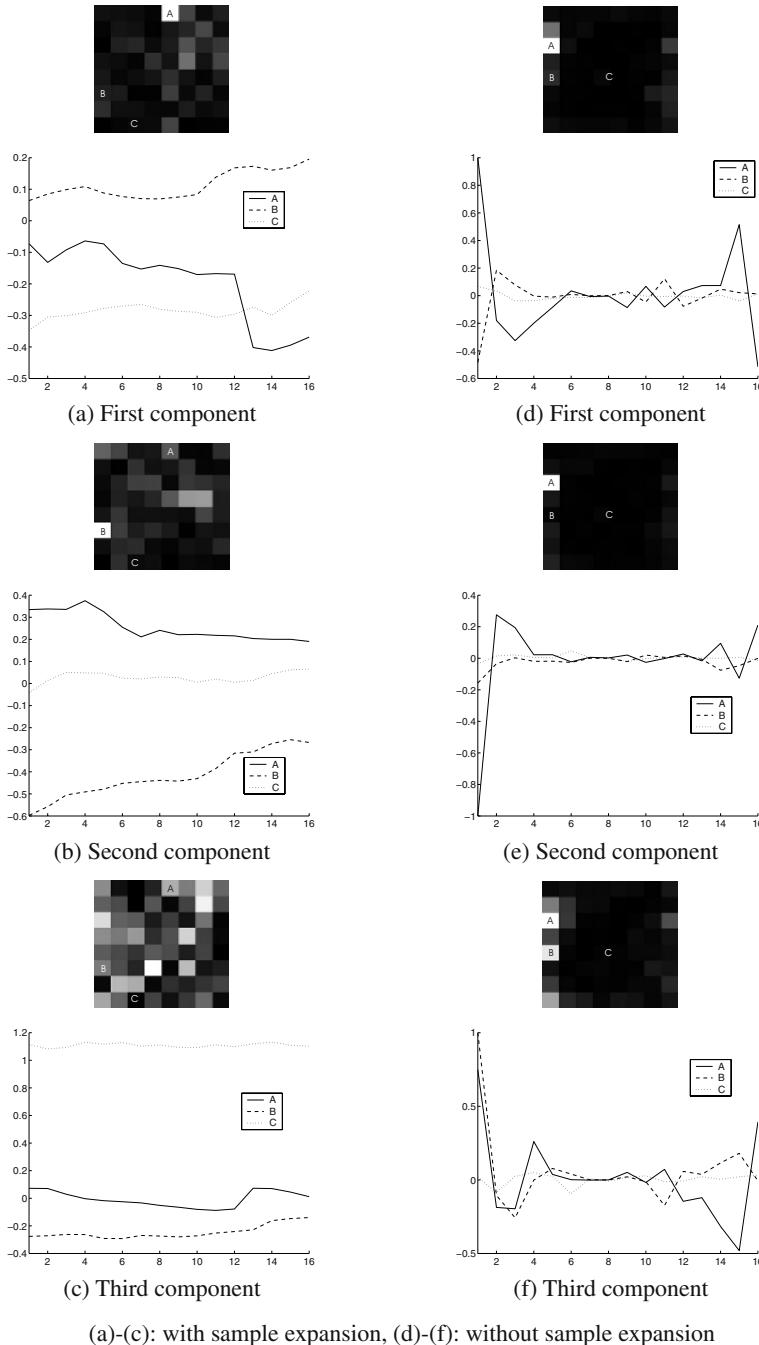
**Fig. 4.** Identical Images in Test Sets

We then review the detected identical images. Figure 4 (a)-(c) are example images successfully recognized as identical image pairs. In particular, (a) is an opening image of news, (b) is a weather forecast CG image of news, and (c) is an opening image of documentary. Since video segments identical to them appear everyday, similar images to them are included both in training and test sets as positive sample pairs. These video segments can be regarded as “punctuation,” so that the results can effectively be used for video parsing and structuring. On the other hand, Fig. 4 (d) shows an image, a pair identical to which is included in the positive samples of the test set, and thus is expected to be detected, but the method fails to detect. Actually the image represents an explanatory video segment, and is shared by news topics, explaining about the related topics broadcasted on the different days. The major reason of the failure is that image pairs identical to this image are included in neither positive nor negative samples of training sets. Statistically, such video segments are negligible, and thus the generalized histogram is statistically successful, but unfortunately may fail to detect extremely rare but interesting multiplicity. On the other hand, punctuation segment detection, as well as commercial film detection can effectively be achieved by the proposed method, because the distribution in the image space of possible multiplicity images is known beforehand.

Then we analyse generator matrix G to inspect the mechanism of the generalized histogram. $g_{ij} = (G)_{i,j}$ corresponds to the weight coefficient for the j -th element of primitive histograms h_j , and $[g_{i1} g_{i2} \dots g_{iN}]$ can be regarded as a vector having the same length to the primitive histograms. In calculating the primitive histograms, we use regular tessellation ($8 \times 8 \times 16$) in the (x, y, v') space to obtain 3-D histograms $h_{i,j,k}$, then convert them into a vector by a particular ordering. We thus inversely apply the ordering to $[g_{i1} g_{i2} \dots g_{iN}]$ to obtain a 3-D tensor $(G)_{p,q,r}^i = g_{pqr}^i = g_{ij}$ where p, q , and r are reversely mapped index of j . In particular, let (p, q) correspond to the location of a region in the (x, y) space, and r correspond to the index of the range of normalized intensity. We then define contribution of the block at (p, q) in the i -th component as the variance of coefficients g_{pqr}^i :

$$\Gamma_{p,q}^i = E \left\{ (g_{pqr}^i - E \{ g_{pqr}^i | p, q, i \})^2 | p, q, i \right\}.$$

$\Gamma_{p,q}^i$ can be visualized as an image in the (p, q) space. In addition, g_{pqr}^i can be regarded as weight coefficients for intensity histograms when we fix p, q , and i . They can be plotted as weight coefficients at the block (p, q) in the i -th component. We then visualize the first, second, and third components of the generator matrices obtained with and without the sample expansion in Fig. 5. For each component, contribution of each block is shown as brightness of the corresponding block; a brighter block represents more contribution. In addition, weight coefficients of blocks, in particular, for three blocks indicated as A, B, and C, are shown in the graphs. With sample expansion (Fig. 5 (a)-(c)), the block A

**Fig. 5.** Visualization of Generator Matrix

has the highest contribution for the first component and the corresponding graph shows relatively high variation. On the other hand, C has very low contribution in the three components, and the corresponding graphs are relatively unchanging. Without sample expansion (Fig. 5 (d)-(f)), it is shown that all three components concentrate on the same block, namely, A, possibly due to overfitting. Weight coefficient graphs are all jagged shapes, disregarding high correlations between neighboring bins. On the other hand, in Fig. 5 (a)-(c) weight coefficient graphs are stable. Thus the effect of the sample expansion is visually shown.

As for processing time, it takes about 4 minutes to train generalized histogram generator matrix, and 12 minutes to convert 10,000 frames in a MPEG-1 file into generalized histograms (including MPEG decoding). Given converted generalized histograms, it takes 4.4 seconds for dirty filtering in matching two sets of 10,000 frames, and 8 minutes for cleansing by NCC when precision is about 90%. We use tree-type data structure SR-tree [18] for range search instead of linear scan to accelerate dirty filtering. The experiments are conducted on an ordinary PC (Pentium 2GHz). The proposed method thus achieves tractable computation time for identical video segment detection.

5 Conclusions

We propose generalized histogram as low-dimensional representation of an image for efficient and precise image matching. Among techniques to enhance the matching performance for histogram-based matching, adaptive binning, subregion histogram, and adaptive weighting are inspected, and we show that these techniques can effectively be realized in the form of linear conversion of high-dimensional primitive histograms. Linear learning algorithm to derive generalized histograms is introduced to take advantage of these enhancement techniques. A sample expansion technique is also introduced to circumvent the overfitting problem due to high-dimensionality and insufficient sample size. The effectiveness of the generalized histogram and sample expansion is revealed with experiments in detecting multiplicity in sets of randomly chosen 20,000 images taken from television broadcast. The matching using generalized histograms achieves almost the same performance compared to the precise matching using NCC, but may miss rare multiplicity which do not appear in training sets. This point should be investigated for future work. Incorporation of high-dimensional indexing techniques as well as extension to identical shot discovery should also be addressed. We also think that the generalized histogram may also be useful for learning framework for appearance-base vision tasks.

References

1. Cheung, S.-S., Zakhor, A.: Video similarity detection with video signature clustering. In: Proc. of International Conference on Image Processing. (2001) 649–652
2. Hampapur, A., Bolle, R.M.: Comparison of distance measures for video copy detection. In: Proc. of ICME. (2001)
3. Jaimes, A., Chang, S., Loui, A.C.: Duplicate detection in consumer photography and news video. In: Proc. of ACM Multimedia. (2002) 423–424

4. Satoh, S.: News video analysis based on identical shot detection. In: Proc. of International Conference on Multimedia and Expo. (2002)
5. Zaniolo, C., Ceri, S., Faloutsos, C., Snodgrass, R.T., Subrahmanian, V.S., Zicari, R.: Advanced Database Systems. Morgan Kaufmann (1997)
6. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys* **33** (2001) 322–373
7. Flickner, M., et al.: Query by image and video content: The QBIC system. *IEEE Computer* (1995) 23–32
8. Vinod, V.V., Murase, H.: Focused color intersection with efficient searching for object extraction. *Pattern Recognition* **30** (1997) 1787–1797
9. Swain, M., Ballard, D.: Color indexing. *International Journal on Computer Vision* **7** (1991) 11–32
10. Sclaroff, S., Taycher, L., Cascia, M.L.: ImageRover: A content-based image browser for the world wide web. In: Proc. of Workshop on Content-Based Access of Image and Video Libraries. (1997) 2–9
11. Puzicha, J., Buhmann, J.M., Rubner, Y., Tomasi, C.: Empirical evaluation of dissimilarity measures for color and texture. In: Proc. of International Conference on Computer Vision. (1999) 1165–1173
12. Leow, W.K., Li, R.: Adaptive binning and dissimilarity measure for image retrieval and classification. In: Proc. of Computer Vision and Pattern Recognition. Volume II. (2001) 234–239
13. Rui, Y., Huang, T.S., Ortega, H., Mehrotra, S.: Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. on Circuits and Systems for Video Technology* **8** (1998) 644–655
14. Hafner, J., Sawhney, H., Equitz, W., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **17** (1995) 729–736
15. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Inc. (2001)
16. Fukunaga, K.: Introduction to Statistical Pattern Recognition (2nd ed.). Academic Press (1990)
17. Yamagishi, F., Satoh, S., Hamada, T., Sakauchi, M.: Identical video segment detection for large-scale broadcast video archives. In: Proc. of International Workshop on Content-Based Multimedia Indexing (CBMI'03). (2003) 135–142
18. Katayama, N., Satoh, S.: The SR-tree: An index structure for high-dimensional nearest neighbor queries. In: Proc. ACM SIGMOD. (1997) 369–380

Recognizing Objects in Range Data Using Regional Point Descriptors

Andrea Frome¹, Daniel Huber², Ravi Kolluri¹,
Thomas Bülow^{1*}, and Jitendra Malik¹

¹ University of California Berkeley, Berkeley CA 94530, USA,
`{afrome,rkolluri,malik}@cs.berkeley.edu`
`thomas.buelow@philips.com`

² Carnegie Mellon University, Pittsburgh PA 15213, USA
`dhuber@cs.cmu.edu`

Abstract. Recognition of three dimensional (3D) objects in noisy and cluttered scenes is a challenging problem in 3D computer vision. One approach that has been successful in past research is the regional shape descriptor. In this paper, we introduce two new regional shape descriptors: 3D shape contexts and harmonic shape contexts. We evaluate the performance of these descriptors on the task of recognizing vehicles in range scans of scenes using a database of 56 cars. We compare the two novel descriptors to an existing descriptor, the spin image, showing that the shape context based descriptors have a higher recognition rate on noisy scenes and that 3D shape contexts outperform the others on cluttered scenes.

1 Introduction

Recognition of three dimensional (3D) objects in noisy and cluttered scenes is a challenging problem in 3D computer vision. Given a 3D point cloud produced by a range scanner observing a 3D scene (Fig. 1), the goal is to identify objects in the scene (in this case, vehicles) by comparing them to a set of candidate objects. This problem is challenging for several reasons. First, in range scans, much of the target object is obscured due to self-occlusion or is occluded by other objects. Nearby objects can also act as background clutter, which can interfere with the recognition process. Second, many classes of objects, for example the vehicles in our experiments, are very similar in shape and size. Third, range scanners have limited spatial resolution; the surface is only sampled at discrete points, and fine detail in the objects is usually lost or blurred. Finally, high-speed range scanners (e.g., flash ladars) introduce significant noise in the range measurement, making it nearly impossible to manually identify objects.

Object recognition in such a setting is interesting in its own right, but would also be useful in applications such as scan registration [9][6] and robot localization. The ability to recognize objects in 2 1/2-D images such as range scans

* Current affiliation is with Philips Research Laboratories, Roentgenstrasse 24-26, 22335 Hamburg

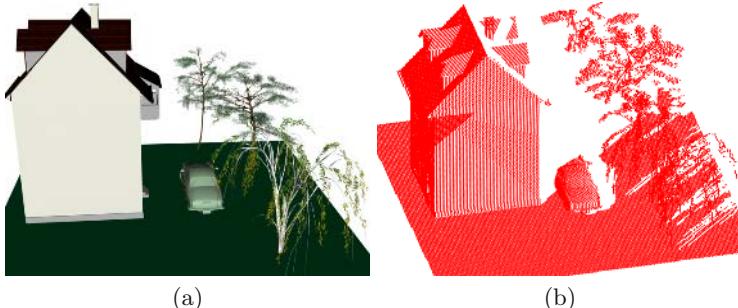


Fig. 1. (a) An example of a cluttered scene containing trees, a house, the ground, and a vehicle to be recognized. (b) A point cloud generated from a scan simulation of the scene. Notice that the range shadow of the building occludes the front half of the vehicle.

may also prove valuable in recognizing objects in 2D images when some depth information can be inferred from cues such as shading or motion.

Many approaches to 3D object recognition have been put forth, including generalized cylinders [3], superquadrics [7], geons [23], medial axis representations [1], skeletons [4], shape distributions [19], and spherical harmonic representations of global shape [8]. Many of these methods require that the target be segmented from the background, which makes them difficult to apply to real-world 3D scenes. Furthermore, many global methods have difficulty leveraging subtle shape variations, especially with large parts of the shape missing from the scene. At the other end of the spectrum, purely local descriptors, such as surface curvature, are well-known for being unstable when faced with noisy data. Regional point descriptors lie midway between the global and local approaches, giving them the advantages of both. This is the approach that we follow in this paper.

Methods which use regional point descriptors have proven successful in the context of image-based recognition [17][15][2] as well as 3D recognition and surface matching [22][13][5][21]. A regional point descriptor characterizes some property of the scene in a local support region surrounding a basis point. In our case, the descriptors characterize regional surface shape. Ideally, a descriptor should be invariant to transformations of the target object (e.g., rotation and translation in 3D) and robust to noise and clutter. The descriptor for a basis point located on the target object in the scene will, therefore, be similar to the descriptor for the corresponding point on a model of the target object. These model descriptors can be stored in a pre-computed database and accessed using fast nearest-neighbor search methods such as locality-sensitive hashing [11]. The limited support region of descriptors makes them robust to significant levels of occlusion. Reliable recognition is made possible by combining the results from multiple basis points distributed across the scene.

In this paper we make the following contributions: (1) we develop the 3D generalization of the 2D shape context descriptor, (2) we introduce the harmonic shape context descriptor, (3) we systematically compare the performance of the 3D shape context, harmonic shape context, and spin images in recognizing sim-

ilar objects in scenes with noise or clutter. We also briefly examine the trade-off of applying hashing techniques to speed search over a large set of objects.

The organization of the paper is as follows: in section 2, we introduce the 3D shape context and harmonic shape context descriptors and review the spin image descriptor. Section 3 describes the representative descriptor method for aggregating distances between point descriptors to give an overall matching score between a query scene and model. Our data set is introduced in section 4, and our experiments and results are presented in section 5. We finish in section 6 with a brief analysis of a method for speeding our matching process.

2 Descriptors

In this section, we provide the details of the new 3D shape context and harmonic shape context descriptors and review the existing spin-image descriptor. All three descriptors take as input a point cloud \mathcal{P} and a basis point p , and capture the regional shape of the scene at p using the distribution of points in a support region surrounding p . The support region is discretized into bins, and a histogram is formed by counting the number of points falling within each bin. For the 3D shape contexts and spin-images, this histogram is used directly as the descriptor, while with harmonic shape contexts, an additional transformation is applied.

When designing such a 3D descriptor, the first two decisions to be made are (1) what is the shape of the support region and (2) how to map the bins in 3D space to positions in the histogram vector. All three methods address the second issue by aligning the support region’s “up” or north pole direction with an estimate of the surface normal at the basis point, which leaves a degree of freedom along the azimuth. Their differences arise from the shape of their support region and how they remove this degree of freedom.

2.1 3D Shape Contexts

The 3D shape context is the straightforward extension of 2D shape contexts, introduced by Belongie et al. [2], to three dimensions. The support region for a 3D shape context is a sphere centered on the basis point p and its north pole oriented with the surface normal estimate \mathcal{N} for p (Fig. 2). The support region is divided into bins by equally spaced boundaries in the azimuth and elevation dimensions and logarithmically spaced boundaries along the radial dimension. We denote the $J + 1$ radial divisions by $R = \{R_0 \dots R_J\}$, the $K + 1$ elevation divisions by $\Theta = \{\Theta_0 \dots \Theta_K\}$, and the $L + 1$ azimuth divisions by $\Phi = \{\Phi_0 \dots \Phi_L\}$. Each bin corresponds to one element in the $J \times K \times L$ feature vector. The first radius division R_0 is the minimum radius r_{\min} , and R_J is the maximum radius r_{\max} . The radius boundaries are calculated as

$$R_j = \exp \left\{ \ln(r_{\min}) + \frac{j}{J} \ln \left(\frac{r_{\max}}{r_{\min}} \right) \right\}. \quad (1)$$

Sampling logarithmically makes the descriptor more robust to distortions in shape with distance from the basis point. Bins closer to the center are smaller in all three spherical dimensions, so we use a minimum radius ($r_{\min} > 0$) to avoid being overly sensitive to small differences in shape very close to the center. The Θ and Φ divisions are evenly spaced along the 180° and 360° elevation and azimuth ranges.

$\text{Bin}(j, k, l)$ accumulates a weighted count $w(p_i)$ for each point p_i whose spherical coordinates relative to p fall within the radius interval $[R_j, R_{j+1}]$, azimuth interval $[\Phi_k, \Phi_{k+1})$ and elevation interval $[\Theta_l, \Theta_{l+1})$. The contribution to the bin count for point p_i is given by

$$w(p_i) = \frac{1}{\rho_i \sqrt[3]{V(j, k, l)}} \quad (2)$$

where $V(j, k, l)$ is the volume of the bin and ρ_i is the local point density around the bin. Normalizing by the bin volume compensates for the large variation in bin sizes with radius and elevation. We found empirically that using the cube root of the volume retains significant discriminative power while leaving the descriptor robust to noise which causes points to cross over bin boundaries. The local point density ρ_i is estimated as the count of points in a sphere of radius δ around p_i . This normalization accounts for variations in sampling density due to the angle of the surface or distance to the scanner.

We have a degree of freedom in the azimuth direction that we must remove in order to compare shape contexts calculated in different coordinate systems. To account for this, we choose some direction to be Φ_0 in an initial shape context, and then rotate the shape context about its north pole into L positions, such that each Φ_l division is located at the original 0° position in one of the rotations. For descriptor data sets derived from our reference scans, L rotations for each basis point are included, whereas in the query data sets, we include only one position per basis point.

2.2 Harmonic Shape Contexts

To compute harmonic shape contexts, we begin with the histogram described above for 3D shape contexts, but we use the bin values as samples to calculate a spherical harmonic transformation for the shells and discard the original histogram. The descriptor is a vector of the amplitudes of the transformation, which are rotationally invariant in the azimuth direction, thus removing the degree of freedom.

Any real function $f(\theta, \phi)$ can be expressed as a sum of complex spherical harmonic basis functions Y_l^m .

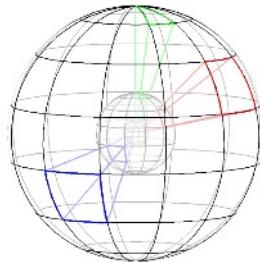


Fig. 2. Visualization of the histogram bins of the 3D shape context.

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{m=l} A_l^m Y_l^m(\theta, \phi) \quad (3)$$

A key property of this harmonic transformation is that a rotation in the azimuthal direction results in a phase shift in the frequency domain, and hence amplitudes of the harmonic coefficients $\|A_l^m\|$ are invariant to rotations in the azimuth direction. We translate a 3D shape context into a harmonic shape context by defining a function $f_j(\theta, \phi)$ based on the bins of the 3D shape context in a single spherical shell $R_j \leq R < R_{j+1}$ as:

$$f_j(\theta, \phi) = SC(j, k, l), \theta_k < \theta \leq \theta_{k+1}, \phi_l < \phi \leq \phi_{l+1}. \quad (4)$$

As in [14], we choose a bandwidth b and store only b lowest-frequency components of the harmonic representation in our descriptor, which is given by $HSC(l, m, k) = \|A_{l,k}^m\|$, $l, m = 0 \dots b, r = 0 \dots K$. For any real function, $\|A_l^m\| = \|A_l^{-m}\|$, so we drop the coefficients A_l^m for $m < 0$. The dimensionality of the resulting harmonic shape context is $K \cdot b(b+1)/2$. Note that the number of azimuth and elevation divisions do not affect the dimensionality of the descriptor.

Harmonic shape contexts are related to the rotation-invariant shape descriptors $SH(f)$ described in [14]. One difference between those and the harmonic shape contexts is that one $SH(f)$ descriptor is used to describe the global shape of a single object. Also, the shape descriptor $SH(f)$ is a vector of length b whose components are the energies of the function f in the b lowest frequencies: $SH_l(f) = \|\sum_{m=-l}^l A_l^m Y_l^m\|$. In contrast, harmonic shape contexts retain the amplitudes of the individual frequency components, and, as a result, are more descriptive.

2.3 Spin Images

We compared the performance of both of these shape context-based descriptors to spin images [13]. Spin-images are well-known 3D shape descriptors that have proven useful for object recognition [13], classification [20], and modeling [10]. Although spin-images were originally defined for surfaces, the adaptation to point clouds is straightforward. The support region of a spin image at a basis point p is a cylinder of radius r_{\max} and height h centered on p with its axis aligned with the surface normal at p . The support region is divided linearly into J segments radially and K segments vertically, forming a set of $J \times K$ rings. The spin-image for a basis point p is computed by counting the points that fall within each ring, forming a 2D histogram. As with the other descriptors, the contribution of each point q_i is weighted by the inverse of that point's density estimate (ρ_i); however, the bins are not weighted by volume. Summing within each ring eliminates the degree of freedom along the azimuth, making spin-images rotationally invariant. We treat a spin-image as a $J \times K$ dimensional feature vector.

3 Using Point Descriptors for Recognition

To compare two descriptors of the same type to one another, we use some measure of distance between the feature vectors: ℓ_2 distance for 3D shape contexts and spin images, and the inverse of the normalized correlation for harmonic shape contexts. Given a query scene \mathcal{S}_q and a set of reference descriptors calculated from scans of known models, we would like to choose the known model which is most similar to an object in \mathcal{S}_q . After we calculate descriptors from \mathcal{S}_q and distances between the query descriptors and reference descriptors, we face the problem of how to aggregate these distances to make a choice as to which model is the best match to \mathcal{S}_q .

A straightforward way of doing this would be to have every descriptor from \mathcal{S}_q vote for the model that gave the closest descriptor, and choose the model with the most votes as the best match. The problem is that in placing a hard vote, we discard the relative distances between descriptors which provide information about the quality of the matches. To remedy this, we use the representative shape context method introduced in Mori et al. [18], which we refer to as the *representative descriptor method*, since we also apply it to spin images.

3.1 Representative Descriptor Method

We precompute M descriptors at points p_1, \dots, p_M for each reference scan \mathcal{S}_i , and compute at query time K descriptors at points q_1, \dots, q_K from the query scene \mathcal{S}_q , where $K \ll M$. We call these K points *representative descriptors* (RDs). For each of the query points q_k and each reference scan \mathcal{S}_i , we find the descriptor p_m computed from \mathcal{S}_i that has the smallest ℓ_2 distance to q_k . We then sum the distances found for each q_k , and call this the *representative descriptor cost* of matching \mathcal{S}_q to \mathcal{S}_i :

$$\text{cost}(\mathcal{S}_q, \mathcal{S}_i) = \sum_{k \in \{1, \dots, K\}} \min_{m \in \{1, \dots, M\}} \text{dist}(q_k, p_m) \quad (5)$$

The best match is the reference model \mathcal{S} that minimizes this cost.

Scoring matches solely on the representative descriptor costs can be thought of as a lower bound on an ideal cost measure that takes geometric constraints between points into account. We show empirically that recognition performance using just these costs is remarkably good even without a more sophisticated analysis of the matches.

One could select the center points for the representative descriptors using some criteria, for example by picking out points near which the 3D structure is interesting. For purposes of this paper, we sidestep that question altogether and choose our basis points randomly. To be sure that we are representing the performance of the algorithm, we performed each representative descriptor experiment 100 times with different random subsets of basis points. For each run we get a recognition rate that is the percentage of the 56 query scenes that we correctly identified using the above method. The mean recognition rate is the recognition rate averaged across runs.

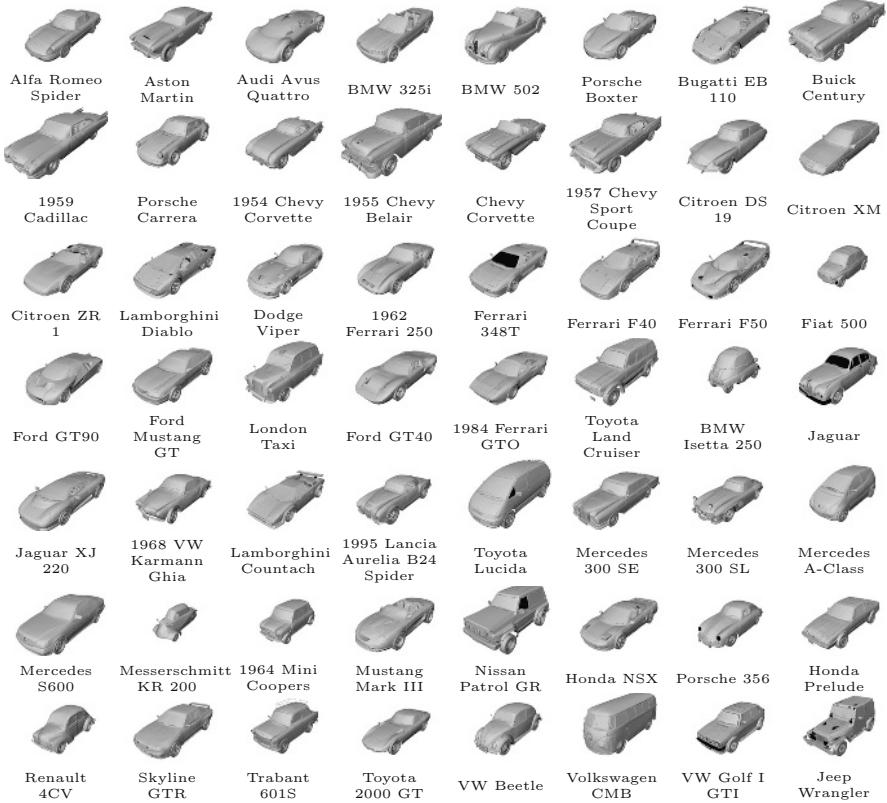


Fig. 3. The 56 car models used in our experiments shown to scale.

4 Data Set

We tested our descriptors on a set of 56 3D models of passenger vehicles taken from the De Espina 3D model library [12] and rescaled to their actual sizes (Fig. 3).¹ The point clouds used in our experiments were generated using a laser sensor simulator that emulates a non-commercial airborne range scanner system. We have shown in separate experiments that these descriptors work well for real data, but for these experiments, our goal was to compare the performance of the descriptors in controlled circumstances.

We generated two types of point clouds: a set of model or “reference” scans, and several sets of scene or “query” scans. For each vehicle, we generated four reference scans with the sensor positioned at 90° azimuth intervals ($\phi = 45^\circ$, 135° , 225° , and 315°), a 45° declination angle, and a range of 500 m from the

¹ The Princeton Shape Benchmark, a data set with 1,814 3D models, was recently released. We didn’t learn of the data set in time to use it in this paper, but we will be using it in future experiments. It can be found online at <http://shape.cs.princeton.edu/benchmark/>.

target. The resulting point clouds contained an average of 1,990 target points spaced approximately 6 cm apart. The query scans were generated in a similar manner, except that the declination was 30° and the azimuth was at least 15° different from the nearest reference scan. Depending on the experiment, either clutter and occlusion or noise was added. Clutter and occlusion were generated by placing the model in a test scene consisting of a building, overhanging trees, and a ground plane (Fig. 1(a)). The point clouds for these scenes contained an average of 60,650 points. Noisy scans were modeled by adding Gaussian noise ($\mathcal{N}(0, \sigma)$) along the line of sight of each point.

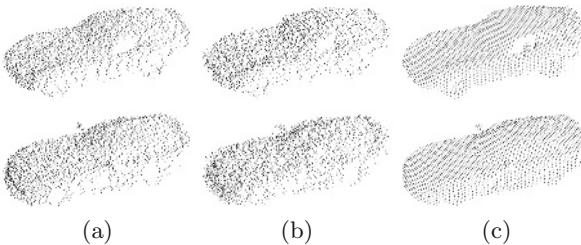


Fig. 4. The top row shows scans from the 1962 Ferrari 250 model, and the bottom scans are from the Dodge Viper. The scans in column (a) are the query scans at 30° elevation and 15° azimuth with $\sigma = 5$ cm noise, and those in (b) are from the same angle but with $\sigma = 10$ cm noise. With 10 cm noise, it is difficult to differentiate the vehicles by looking at the 2D images of the point clouds. Column (c) shows the reference scans closest in viewing direction to the query scans (45° azimuth and 45° elevation). In the 5 cm and 10 cm noise experiments, we first chose 300 candidate basis points and sampled RDs from those.

Basis points for the descriptors in the reference point clouds were selected using a method that ensures approximately uniform sampling over the model's visible surface. Each point cloud was divided into 0.2-meter voxels and one point was selected at random from each occupied voxel, giving an average of 373 descriptors per point cloud (1,494 descriptors per model). Basis points in the query point clouds were chosen using the same method, except that the set was further reduced by selecting a random subset of N basis points ($N=300$ for the clutter-free queries and $N=2000$ for the clutter queries) from which representative descriptors were chosen. For a given experiment, the same subset of basis points were used in generating the three types of descriptors. After noise and clutter were added, normals for the basis points were computed using a method which preserves discontinuities in the shape and that accounts for noise along the viewing direction [16]. The algorithm uses points within a cube-shaped window around the basis point for the estimation, where the size of the window can be chosen based on the expected noise level.

5 Experiments

The parameters for the descriptors (Table 1) were chosen based on extensive experimentation on other sets of 3D models not used in these experiments (Table 1). However, some parameters (specifically K and r_{\min}) were fine-tuned using descriptors in 20 randomly selected models from our 56 vehicle database. The basis points used for training were independent from those used in testing. The relative scale of the support regions was chosen to make the volume encompassed comparable across descriptors.

Table 1. Parameters used in the experiments for shape contexts (SC), harmonic shape contexts (HSC), and spin images (SI). All distances are in meters

	SC	HSC	SI
max radius (r_{\max})	2.5	2.5	2.5
min radius (r_{\min})	0.1	0.1	-
height (h)	-	-	2.5
radial divisions (J)	15	15	15
elev./ht. divisions (K)	11	11	15
azimuth divisions (L)	12	12	-
bandwidth (b)	-	16	-
dimensions	1980	2040	225
density radius (δ)	0.2	0.2	0.2

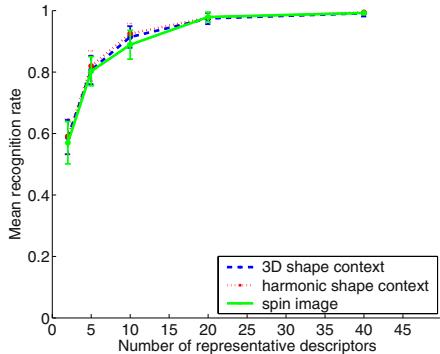


Fig. 5. Results for the 5cm noise experiment. All three methods performed roughly equally. From 300 basis points sampled evenly from the surface, we chose varying numbers of RDs, and recorded the mean recognition rate. The error bars show one standard deviation.

5.1 Scenes with 5 cm Noise

In this set of experiments, our query data was a set of 56 scans, each containing one of the car models. We added Gaussian noise to the query scans along the scan viewing direction with a standard deviation of 5 cm (Fig. 4). The window for computing normals was a cube 55 cm on a side. Fig. 5 shows the mean recognition rate versus number of RDs. All of the descriptors perform roughly equally, achieving close to 100% average recognition with 40 RDs.

5.2 Scenes with 10 cm Noise

We performed two experiments with the standard deviation increased to 10 cm (see Fig. 4). In the first experiment, our window size for computing normals was the same as in the 5 cm experiments. The results in Fig. 6 show a significant decrease in performance by all three descriptors, especially spin images. To test how much the normals contributed to the decrease in recognition, we performed

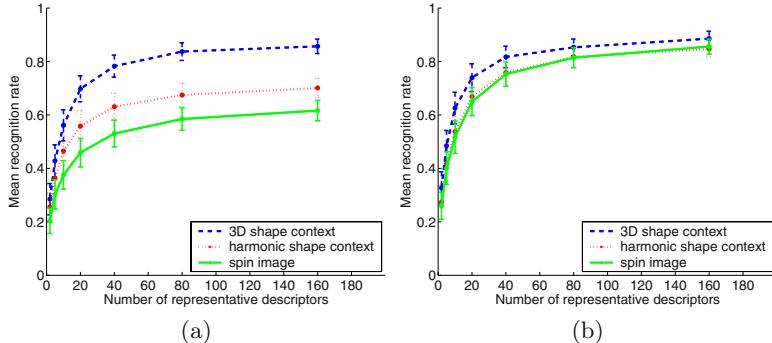


Fig. 6. Results for 10 cm noise experiments. In experiment (a) we used a window for the normals that was a cube 55 cm on a side, whereas in (b) the size was increased to a cube 105 cm on a side. The error bars show one standard deviation from the mean. From this experiment, we see that shape contexts degrade less as we add noise and in particular are less sensitive to the quality of the normals than spin images. All three methods would benefit from tuning their parameters to the higher noise case, but this would entail a recalculation of the reference set. In general, a method that is more robust to changes in query conditions is preferable.

a second experiment with a normal estimation window size of 105 cm, giving us normals more robust to noise. The spin images showed the most improvement, indicating their performance is more sensitive to the quality of the normals.

5.3 Cluttered Scenes

To test the ability of the descriptors to handle a query scene containing substantial clutter, we created scenes by placing each of the vehicle models in the clutter scene shown in Fig. 1(a). We generated scans of each scene from a 30° declination and two different azimuth angles ($\phi = 150^\circ$ and $\phi = 300^\circ$), which we will call views #1 and #2 (Fig. 7). We assume that the approximate location of the target model is given in the form of a box-shaped volume of interest (VOI). The VOI could be determined automatically by a generic object saliency algorithm, but for the controlled experiments in this paper, we manually specified the VOI to be a $2 \text{ m} \times 4 \text{ m} \times 6 \text{ m}$ volume that contains the vehicle as well as some clutter, including the ground plane (Fig. 7(b)). Basis points for the descriptors were chosen from within this VOI, but for a given basis point, all the scene points within the descriptor's support region were used, including those outside of the VOI.

We ran separate experiments for views 1 and 2, using 80 RDs for each run. When calculating the representative descriptor cost for a given scene-model pair, we included in the sum in equation (5) only the 40 smallest distances between RDs and the reference descriptors for a given model. This acts as a form of outlier rejection, filtering out many of the basis points not located on the vehicle. We chose 40 because approximately half of the basis points in each VOI fell on a vehicle. The results are shown in Fig. 8.

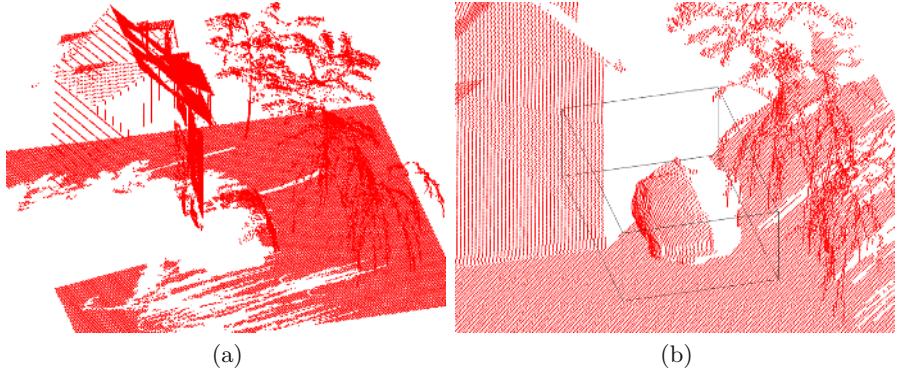


Fig. 7. The cluttered scene with the Karmann Ghia. Picture (a) is the scan from view 2, and (b) is a close-up of the VOI in view 1. For the fully-rendered scene and the full scan from view 1, refer to Fig. 1. The scanner in view 1 was located on the other side of the building from the car, causing the hood of the car to be mostly occluded. In view 2, the scanner was on the other side of the trees, so the branches occlude large parts of the vehicle. There were about 100 basis points in the VOI in each query scene, and from those we randomly chose 80 representative descriptors for each run.

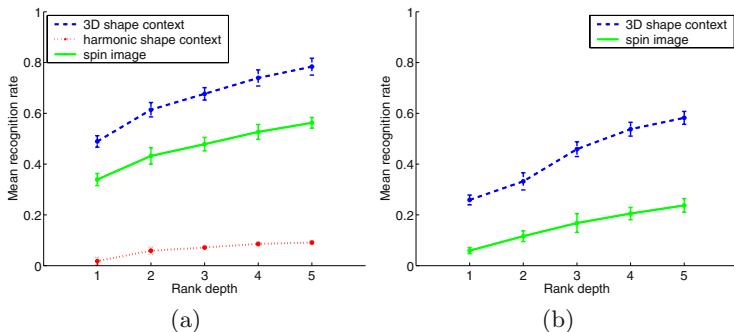


Fig. 8. Cluttered scene results. In both, we included in the cost the 40 smallest distances out of those calculated for 80 RDs. The graphs show recognition rate versus rank depth with error bars one standard deviation from the mean. We calculated the recognition rate based on the k best choices, where k is our *rank depth* (as opposed to considering only the best choice for each query scene). We computed the mean recognition rate as described before, but counted a match to a query scene as “correct” if the correct model was within the top k matches. Graph (a) shows the results for view #1 and (b) for view #2. Using the 3D shape context we identifying on average 78% of the 56 models correctly using the top 5 choices for each scene, but only 49% of the models if we look at only the top choice for each. Spin images did not perform as well; considering the top 5 matches, spin images achieved a mean recognition rate of 56% and only 34% if only the top choice is considered. Harmonic shape contexts do particularly bad, achieving recognition slightly above chance. They chose the largest vehicles as matches to almost all the queries.

The shape context performance is impressive given that this is a result of doing naive point-to-point matching without taking geometric constraints into account. Points on the ground plane were routinely confused for some of the car models which geometric constraints could rule out. A benefit of the 3D shape context over the other two descriptors is that a point-to-point match gives a candidate orientation of the model in the scene which can be used to verify other point matches.

6 Speeding Search with Locality-Sensitive Hashing

In this section, we briefly explore the cost of using 3D shape contexts and discuss a way to bring the amount of computation required for a 3D shape context query closer to what is used for spin images while maintaining accuracy.

In the spin image and harmonic shape context experiments, we are comparing each of our representative descriptors to 83,640 reference descriptors. We must compare to the 12 rotations when using 3D shape contexts, giving a total of 1,003,680. Our system implementation takes 7.4 seconds on a 2.2 GHz processor to perform the comparison of one 3D shape context to the reference set.

Fast search techniques such as locality-sensitive hashing (LSH) [11] can reduce the search space by orders of magnitude, making it more practical to search over the 3D shape context rotations, though there is a tradeoff between speed and accuracy of the nearest-neighbor result. The method divides the high-dimensional feature space where the descriptors lie into hypercubes, divided by a set of k randomly-chosen axis-parallel hyperplanes. These define a hash function where points that lie in the same hypercube hash to the same value. The greater the number of planes, the more likely that two neighbors will have different hash values. The probability that two nearby points are separated is reduced by independently choosing l different sets of hyperplanes, thus defining l different hash functions. Given a query vector, the result is the set of hashed vectors which share one of their l hash values with the query vector.

In Figure 9, we show LSH results on the 10cm noise dataset with the 105 cm window size using 160 RDs (exact nearest neighbor results are shown in Figure 6(b)). We chose this data set because it was the most challenging of the noise tests where spin images performed well (using an easier test such as the 5 cm noise experiment provides a greater reduction in the number of comparisons). In calculating the RD costs, the distance from a query point

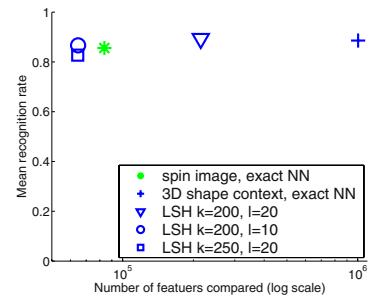


Fig. 9. Results for LSH experiments with 3D shape contexts on the 10cm noise query dataset using the 105 cm window size. Shown are results using 160 RDs where we included the 80 smallest distances in the RD sum. The exact nearest neighbor results for spin images and 3D shape contexts are shown for comparison.

to a given model for which there were no hash matches was set to a value larger than any of the other distances. In this way, we penalized for a failure to match any hashed descriptors. To remove outliers caused by unlucky hash divisions, we included in the sum in equation (5) only the 80 smallest distances between RDs and the returned reference descriptors. Note that performing LSH using 3D shape contexts with $k = 200$ hash divisions and $l = 10$ hash functions requires fewer descriptor comparisons than an exact nearest neighbor search using spin images, and provides slightly better accuracy.

Acknowledgements. We would like to thank Bogdan Matei at Sarnoff Corporation for use of his normal calculation code and technical support. Thanks also to Anuj Kapuria and Raghu Donamukkala at Carnegie Mellon University, who helped write the spin image code used for our experiments. This work was supported in part by the DARPA E3D program (F33615-02-C-1265) and NSF ITR grant IIS-00-85864.

References

1. E. Bardinet, S. F. Vidal, Arroyo S. D., Malandain G., and N. P. de la Blanca Capilla. Structural object matching. Technical Report DECSAI-000303, University of Granada, Dept. of Computer Science and AI, Granada, Spain, February 2000.
2. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
3. T. O. Binford. Visual perception by computer. Presented at IEEE Conference on Systems and Control, Miami, FL, 1971.
4. Bloomenthal and C. Lim. Skeletal methods of shape manipulation. In *International Conference on Shape Modeling and Applications*, pages 44–47, 1999.
5. Chin Seng Chua and Ray Jarvis. Point signatures: a new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, Oct 1997.
6. D.Zhang and M.Herbert. Experimental analysis of harmonic shape images. In *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling*, pages 191–200, October 1999.
7. Solina F. and Bajcsy R. Recovery of parametric models from range images: The case for superquadrics with global deformations. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, February 1990.
8. Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22:83–105, January 2003.
9. G.Roth. Registering two overlapping range images. In *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling*, pages 191–200, October 1999.
10. Daniel F. Huber and Martial Hebert. Fully automatic registration of multiple 3D data sets. *Img. and Vis. Comp.*, 21(7):637–650, July 2003.

11. P. Indyk and R. Motwani. Approximate nearest neighbor - towards removing the curse of dimensionality. In *Proceedings of the 30th Symposium on Theory of Computing*, 1998.
12. De Espona Infografica. *De Espona 3D Models Encyclopedia*.
<http://www.deespona.com/3dencyclopedia/menu.html>.
13. Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.
14. Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164. Eurographics Association, 2003.
15. D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1000–1015, Sep 1999.
16. Bogdan Matei and Peter Meer. A general method for errors-in-variables problems in computer vision. In *CVPR*, volume 2, June 2000.
17. K. Mikolajczk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, volume II, pages 257–263, Jun 2003.
18. G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *CVPR*, volume 1, pages 723–730, 2001.
19. R. Osada, T. Funkhouser, B. Chayelle, and D. Dobkin. Matching 3d models with shape distributions. In *Shape Modeling International*, May 2001.
20. Salvador Ruiz-Correa, Linda Shapiro, and Marina Miela. A new paradigm for recognizing 3d object shapes from range data. In *ICCV*, Oct 2003.
21. Fridtjof Stein and Gerard Medioni. Structural indexing: efficient 3D object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):125–45, Feb 1992.
22. Y. Sun and M.A. Abidi. Surface matching by 3d point’s fingerprint. In *ICCV*, pages 263–9, Jul 2001.
23. K. Wu and Levine M. Recovering parametrics geons from multiview range data. In *CVPR*, June 1994.

Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces

Ye Duan¹, Liu Yang², Hong Qin², and Dimitris Samaras²

¹ Department of Computer Science

University of Missouri at Columbia, Columbia, MO 65203, USA

duanye@missouri.edu,

<http://www.cs.missouri.edu/~duan>

² Center for Visual Computing, Department of Computer Science

State University of New York at Stony Brook, NY 11794, USA

{yliu, qin, samaras}@cs.sunysb.edu

<http://www.cs.sunysb.edu/~{yliu,qin,samaras}>

Abstract. In this paper, we propose a new PDE-based methodology for deformable surfaces that is capable of automatically evolving its shape to capture the geometric boundary of the data and simultaneously discover its underlying topological structure. Our model can handle multiple types of data (such as volumetric data, 3D point clouds and 2D image data), using a common mathematical framework. The deformation behavior of the model is governed by partial differential equations (e.g. the weighted minimal surface flow). Unlike the level-set approach, our model always has an explicit representation of geometry and topology. The regularity of the model and the stability of the numerical integration process are ensured by a powerful Laplacian tangential smoothing operator. By allowing local adaptive refinement of the mesh, the model can accurately represent sharp features. We have applied our model for shape reconstruction from volumetric data, unorganized 3D point clouds and multiple view images. The versatility and robustness of our model allow its application to the challenging problem of multiple view reconstruction. Our approach is unique in its combination of simultaneous use of a high number of arbitrary camera views with an explicit mesh that is intuitive and easy-to-interact-with. Our model-based approach automatically selects the best views for reconstruction, allows for visibility checking and progressive refinement of the model as more images become available. The results of our extensive experiments on synthetic and real data demonstrate robustness, high reconstruction accuracy and visual quality.

1 Introduction

During the past decade, PDE-driven surface evolution has become very popular in the computer vision community for shape recovery and object detection. Most of the existing work is based on the Eulerian approach, i.e., the geometry and topology of the shape is implicitly defined as the level-set solution of time-varying

implicit functions over the entire 3D space [24], which can be computationally very expensive. In this paper, we propose a new PDE-based deformable model that, in contrast, takes the Lagrangian approach, i.e., the geometry and topology of the deformable surface are always explicitly represented throughout the simulation process. The elegance of our approach lies in the fact that we can use the same PDE-based model for different types of data. The only thing that is data-dependant is the control function, which describes the interaction with the data. This is an important property that will allow easy application of our methodology to other types of data, such as points, surfels, images and to incorporate other visual cues such as shading and optical flow.

Starting with [17], deformable models have achieved great success in the areas of computer vision and pattern recognition. In general, deformable models can be divided into two categories: explicit models and implicit models. Explicit models include parametric representations [27] and discrete representations [28]. Implicit models [24,5,14] handle topology changes, based on the modeling of propagating fronts, which are the level set of some scalar function. The desirable shape must be explicitly evaluated using marching-cube-like techniques [23] in an additional post-processing stage. The narrow band algorithm [14] can reduce the computational cost related to the higher dimension. Recently, topologically adaptive explicit models have been proposed [26,22], reviewed in detail in [25]. The aforementioned deformable models were proposed mainly for the purpose of shape reconstruction from volumetric data and for medical image segmentation. For shape reconstruction from point clouds, existing work is mostly on static methods. They are either explicit methods [11,1,8], implicit methods [15] or based on radial basis functions [4,10].

Compared with level-set based methods, our new model is simpler, more intuitive, and makes it easier to incorporate user-control during the deformation process. To ensure the regularity of the model and the stability of the numerical integration process, powerful Laplacian tangential smoothing, along with commonly used mesh optimization techniques, is employed throughout the geometric deformation and topological variation process. The new model can either grow from the inside or shrink from the outside, and it can automatically split to multiple objects whenever necessary during the deformation process. More importantly, our model supports level-of-details control through global subdivision and local/adaptive subdivision. Based on our experiments, the new model can generate a good, high-quality polygonal mesh that can capture underlying topological structure simultaneously from various datasets such as volumetric data, 3D unorganized point clouds and multiple view images. The explicit representation of the model enables us to check for visibility and camera pose directly.

Automatic reconstruction of 3D objects and environments from photographic images is important for many applications that integrate virtual and real data. [29] Many approaches have been used to solve the problem, e.g. by matching features [34] or textures [12]. In traditional stereo methods, many partial models must be computed with respect to a set of base viewpoints, and the surface patches must be fused into a single consistent model [2] by Iterative Closest Points(ICP) [30], but a parameterized model is still needed for final dense sur-

face reconstruction, and there is no explicit handling of occlusion. Meshes and/or systems of particles [13] can be deformed according to constraints derived from images, but may end up clustering in areas of high curvature, and often fail with complicated topology. Recently, voxel-based approaches have been widely used to represent 3D shape [12,32,21,3,7] based on 3D scene space instead of image space. Marching-cube-like techniques [23] are still necessary to get the parameterized surface. The space carving method [21] recovers a family of increasingly tighter supersets of the true scene. Once a voxel is carved away, it cannot be recovered, and any errors propagate into further erroneous carvings. This is partially alleviated by probabilistic space carving [8]. The level set method [12,9,16] is based upon variational analysis of the objects in the scene and their images while handling topology changes automatically. To overcome the complexity of implicit representation of objects in the level set approach, [18,19] operate on a surface represented by a depth function, at the cost of being limited to surface patches. A multi-resolution method using space carving and level sets methods [33], starts with coarse settings, refined when necessary.

However these methods do not take advantage of the existence of a simple explicit geometric representation. We show that high quality results can be achieved with the use of a simple intuitive mesh, easy to interact with and can both incorporate all the available image information and also allow for progressive refinement as more images become available. With increased computer performance, our reconstruction method will soon achieve interactive run times. One can envision user controlling the quality of the reconstruction during image capture, being able to capture the most necessary remaining views to complete the reconstruction [31]. Our PDE-based deformable model is described in sec. 2 and applied to volumetric data in sec. 3.1, unorganized point clouds in sec. 3.2 and multi-view images in sec. 3.3. Experimental results on synthetic and real data are presented in sec. 4.

2 PDE-Based Deformable Surface

The deformation behavior of our new deformable surface is governed by an evolutionary system of nonlinear initial-value partial differential equations (PDE) with the following general form:

$$\frac{\partial S(p)}{\partial t} = F(t, k, k', f \dots) U(p, t) \quad (1)$$

where F is speed function, t is the time parameter, k and k' are the surface curvature and its derivative at the point p , and f is the external force. $S(p, 0) = S_0(p)$ is the initial surface. U is the unit direction vector and often it represents the surface normal vector. Eq. 1 can be either directly provided by the user, or more generally, obtained as a gradient descent flow by the Euler-Lagrange equation of some underlying energy functionals based on the calculus of variations.

2.1 Model Refinement

Once an initial shape of the object is recovered, the model can be further refined several times to improve the fitting accuracy. In this paper, we have implemented two kinds of model refinement: global refinement and local/adaptive refinement. Global refinement is conducted by Loop's subdivision scheme [6].

Adaptive refinement is guided by fitting accuracy as measured by the variance of the distance from the triangle to the boundary of the object [36]. If the variance of the distance samples for a given triangle is bigger than a user defined threshold, then this triangle will be refined. The variance of a discrete set of distances is computed in the standard way: $V_T[d] = E[d^2] - E[d]^2$, where E denotes the mean of its argument. To calculate the variance of the distance samples for a given triangle, we temporarily quadrisect the triangle T into four smaller triangles and for each smaller triangle, calculate the distance at its barycentric center. At each level of adaptive refinement, all the triangles with fitting accuracy below the user-specified threshold will be quadrisected. The deformation of the model will resume only among those newly refined regions. In Fig. 4, we show different levels of refinements.

2.2 Mesh Regularity

To ensure that the numerical simulation of the deformation process proceeds smoothly, we must maintain mesh regularity so that the mesh's nodes have a good distribution, a proper node density, and a good aspect ratio of the triangles. This is achieved by the incorporation of a tangential Laplacian operator, and three mesh operations: edge split, edge collapse, and edge swap.

The tangential Laplacian operator is used to maintain good node distribution. The Laplacian operator, in its simplest form, moves repeatedly each mesh vertex by a displacement equal to a positive scale factor times the average of the neighboring vertices.

When edge lengths fall outside a predefined range, edge splitting and edge collapsing are used to keep an appropriate node density. Edge swapping is used to ensure a good aspect ratio of the triangles. This can be achieved by forcing the average valence to be as close to 6 as possible [20]. An edge is swapped if and only if the quantity $\sum_{p \in A} (valence(p) - 6)^2$ is minimized after the swapping.

2.3 Topology Modification

In order to recover a shape of arbitrary, unknown topology, the model must be able to modify its topology properly whenever necessary. In general, there are two kinds of topology operations: (1) Topology Merging, and (2) Topology Splitting.

Topology Merging. We propose a novel method called "lazy merging" to handle topology merging. The basic idea is that whenever two non-neighboring vertices are too close to each other, they will be deactivated. Topology merging

will happen only after the deformation of the model stops and all the vertices become non-active. There are three steps in the topology merging operation: (1) *Collision Detection*, (2) *Merging-vertices Clustering*, and (3) *Multiple-contours Stitching*. **Collision Detection:** Collision detection is done hierarchically in two different levels: coarser-level and finer-level. Coarser-level collision detection is mainly for the purpose of collision exclusion. For each active vertex V , we will calculate its distance to all other non-neighboring active vertices. Vertices whose distance to the current vertex V is small enough so that no collision might happen between, will be passed to the finer-level collision detection. For each face f with three corner points (u, v, w) that is adjacent to one of the vertices being passed into the finer level of collision detection, we will calculate the distance between a number of sample points $\alpha u + \beta v + \gamma w$ of the face f with barycentric coordinates $\alpha + \beta + \gamma = 1$ and the current vertex V . If at least one of these distances is smaller than the collision threshold, the two corresponding vertices will be marked as merging vertices and will be deactivated. **Merging-Vertices Clustering:** After all the merging vertices have been deactivated, we need to divide them into several connected clusters. We randomly pick any merging vertex and find all of its connected merging vertices by a breadth-first search. We continue recursively, until all the merging vertices belong to appropriate merging vertex clusters. Then for each cluster of merging vertices, all the interior vertices will be removed and the remaining vertices will be put into a linked list. This is based on the following observation: *when two regions are merging together, only the boundary regions will remain, all the interior regions will be burned out (i.e. removed).*

Multiple-Contours Stitching: After the merging vertex linked lists have been created, we stitch them together in three separate steps:

1. For each vertex in the linked lists, find its closest vertex in other linked lists.
2. Based on the proximity information obtained from the previous step, find a pair of vertices A and B such that they are adjacent to each other in the linked list L (Fig. 1 (a)), and their closest merging vertices A' and B' are also adjacent to each other in the corresponding linked list L' , in addition, the closest merging vertices of A' and B' are A and B , respectively. Starting from this pair of vertices A and B , iteratively go through the linked lists and if possible, connect each pair of adjacent vertices in one linked list to a corresponding vertex in another linked list and create a new triangle.
3. If there are more than two linked lists to be stitched together, then after stitching all the corresponding vertices, there may be some in-between gaps that need to be filled in. For example, in (Fig. 1 (a)), there is a gap between the linked lists L , L' and L'' that consists of vertices B , C , C'' , C' and B' . We filled in the gap by creating a new vertex E at the center and connecting the new vertex E with all the other vertices in the loop (Fig. 1 (b)).

Topology Splitting. Topology splitting occurs when a part of the surface tends to shrink to a single point. In this scenario, the surface has to split up into two parts precisely at that location. We use a method similar to [22]. In particular, a split-operation is triggered if there exists three neighboring vertices which are

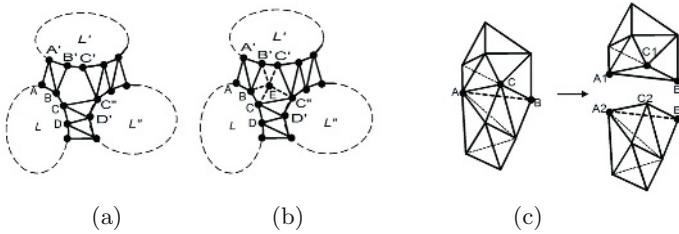


Fig. 1. (a)(b) Multiple contours stitching. (a) New triangles are created between corresponding vertices, and a gap is created by vertices B, C, C'', C', B' . (b) The gap is filled in by creating a new vertex E in the center and connecting it with all the other vertices in the gap. (c) Topology split by splitting the virtual face ABC into two faces whose orientations are opposite to each other.

interconnected to each other, but the face formed by these three vertices does not belong to the model (i.e., a virtual face), and if the length of any of the three edges of the virtual face is smaller than the minimum edge length threshold and thus needs to be collapsed. For example, in Fig. 1(c), face ABC represents a virtual face that needs to be split. We divide the surface exactly at this location by cutting it into two open sub-surfaces. Then we close the two split-in-two surfaces using two faces $A_1B_1C_1$ and $A_2C_2B_2$ whose orientations are opposite to each other. Finally, we reorganize the neighborhood.

3 Surface Reconstruction

We have applied our PDE-based deformable surface to shape reconstruction from volumetric data, unorganized 3D point clouds and multi-view 2D images. The PDE we used is the general weighted minimal surface flow [5]:

$$\frac{\partial S}{\partial t} = (g(v + H) - \nabla g \cdot \mathbf{N})\mathbf{N}, \quad S(0) = S_0 \quad (2)$$

where $S = S(t)$ is the 3D deformable surface, t is the time parameter, and S_0 is the initial shape of the surface. Note that, H is the mean curvature of the surface, and \mathbf{N} is the unit normal of the surface. v is a constant velocity that will enable the convex initial shape to capture non-convex, arbitrary complicated shapes. It is also useful to avoid allowing the model to get stuck into local minima during the evolution process. g is a monotonic non-increasing, non-negative function that is used for interaction of the model with the datasets, and will stop the deformation of the model when it reaches the boundary of the object. In essence, Eq. 2 controls how each point in the deformable surface should move in order to minimize the weighted surface area. Hence, the detected object is given by the steady-state solution of the equation: $S_t = 0$, i.e. when the velocity F is zero. In order to apply the model to different types of data, we simply need to provide the definition of g that is appropriate for the dataset. For example, in 2D multi-view based reconstruction, direct checking for visibility and camera pose can be easily incorporated into the appropriate control function.

3.1 Volumetric Images

For volumetric data sets, the stopping function g is defined as:

$$g(S) = \frac{1}{1 + |\nabla(G_\sigma * I(S))|^2} \quad (3)$$

where I is the volumetric density function, and $G_\sigma * I$ is the smoothed density function by convoluting with a Gaussian filter with variance σ .

3.2 Point Clouds

For surface reconstruction from 3D unorganized point clouds, we use a simplified version of Eq. 2 with part $\nabla g \cdot \mathbf{N}$ removed. The stopping function g is:

$$g(p) = \begin{cases} 1, & \text{if } D(p) < T_D \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $D(p)$ is the distance between current position p and its closest data points, T_D is the threshold distance that is decided by the sampling rate of the point clouds datasets. In order to efficiently find the closest data points of any given position p , we preprocess the point clouds by putting them into a uniform regular grid and connecting all the points inside one grid cell by a linked list. The above distance threshold T_D will stop the movement of the model before it arrives at the "real" boundary of the object. To reduce the distance from the model to the boundary of the object, after the model stops its deformation, we will project each vertex point to the local tangent plane of its k -nearest neighbors.

The local tangent plane can be estimated using principle component analysis (PCA) [15]: For any point p , its local tangent plane can be represented by a center point c and a unit normal vector n . The center point c is the centroid of the one neighborhood of point p , which is denoted as $Nbhd(p)$. The normal vector n is computed by doing eigenanalysis of the covariance matrix C of $Nbhd(p)$, which is a symmetric 3×3 positive semi-definite matrix:

$$C = \sum_{p_i \in Nbhd(p)} (p_i - c) \otimes (p_i - c) \quad (5)$$

Here, \otimes denotes the outer product vector operator, and the normal vector n is the eigenvector associated with the smallest eigenvalue of the covariance matrix C . In our experiments, k is set to five.

3.3 2D Multiple Views

The new model is capable of recovering shape not only from 3D data but also from 2D images. In the case of 2D photographic images, we use a photo consistency as the weight function g in Eq. 2. Here we use photo consistency criterion similar to the one in [21]. We only calculate the photo consistency w.r.t. each

of the model vertices. For numerical robustness, the consistency calculation is performed by projecting a small patch around each vertex, to the image planes. It is assumed that the patch is reasonably small w.r.t. the distance between the object and the camera, while it is large enough to capture local features over images. A reasonable planar approximation for Lambertian surfaces is to take a patch on the tangent plane to the vertex. Again, we can use PCA to estimate the normal of the tangent plane, which is actually the first order approximation for the current model surface. In order to find the correct correspondence of a point i_1 in correlation window A to a point i_2 in correlation window B (A and B are projections of surface patch P onto different images), we back-project i_1 to a point $p_1 \in P$ and then reproject p_1 to i_2 in correlation window B . Then we can calculate the photo consistency within patch projections in different views as discussed in [21]:

$$\begin{aligned} g = \sigma^2 &= \sigma_R^2 + \sigma_G^2 + \sigma_B^2, \quad \sigma_R^2 = \frac{1}{N-1} \sum_i^N R_i^2 - (\frac{1}{N} \sum_i^N R_i)^2 \\ \sigma_G^2 &= \frac{1}{N-1} \sum_i^N G_i^2 - (\frac{1}{N} \sum_i^N G_i)^2, \quad \sigma_B^2 = \frac{1}{N-1} \sum_i^N B_i^2 - (\frac{1}{N} \sum_i^N B_i)^2 \end{aligned} \quad (6)$$

where N is the number of selected views. We only select the best- N views for our reconstruction, ignoring the most degenerate views. For a particular camera, using the OpenGL depth buffer, we first check visibility based on the current model. The visibility info can be calculated for every iteration with the complexity $O(mn)$, where m is the number of total cameras, and n is the number of surface vertices.

Among the visible points, we take those with the largest projection area to the image plane. The projection area gives both distance and pose information.

$$\frac{\delta O}{\delta I} = \frac{\cos \alpha}{\cos \theta} \left(\frac{Z}{f} \right)^2 \quad (7)$$

Using the geometric properties of solid angles [35], we derive our camera weight $w_{i,j}$ from Eq. 7

$$w_{i,j} = \frac{\cos \theta}{(\cos \alpha)^3} \left(\frac{f}{D} \right)^2, \quad w_{i,j} = 0 \text{ when not visible} \quad (8)$$

In Fig. 2, we show how to evaluate camera position and orientation.

Using Eq. 2, we can set the speed function for each vertex. We can start from a coarse mesh (like a cube), and subdivide it after it is close to the surface

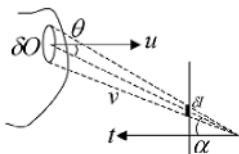


Fig. 2. Selecting the best- N views by camera positions

of the object. More specifically, we can use adaptive mesh refinement to locally subdivide areas with large curvature, where most details exist. The regularization term helps to keep the surface smooth and to make it converge to the final result. Since we have the explicit representation of the object surface, we can always incorporate new images taken from new camera positions, and incrementally enhance the quality of the final reconstruction. The complete algorithm is:

1. Preprocessing: For each camera position check visibility using the depth buffer value of each vertex.
2. For each vertex point on the model:
 - a) For each camera, check camera visibility and pose.
 - b) Select the best N-views.
 - c) Calculate the photo consistency g at the vertex position (first order patch approximation).
 - d) Calculate the gradient $\nabla(g)$ at the vertex position.
 - e) Get the vertex speed from the PDE of Eq. 2, and the new vertex position in the next iteration.
3. Laplacian regularization.
4. Mesh adaptive refinement if necessary.

4 Experimental Results

Results from 3D Data: In this section, we will show experimental results on both real and synthetic datasets. In all the following figures, grey regions represent parts of the model that are still active and deforming, black regions represent deactivated parts of the model that have already reached the boundary of the object. In order to illustrate the good mesh quality generated by our new model, we will show at least one wireframe view of the model in all the following figures. The input volumetric dataset of Fig. 3 is obtained from CT scanning of a phantom of the vertebra. The data size is $128 \times 120 \times 52$ voxels. Fig. 4 and Fig. 5 illustrate the surface reconstruction process from 3D unorganized point clouds. The input dataset of Fig. 4 is the mannequin head with 7440 data points. The original dataset has a big hole in the bottom of the head. We have manually filled in the hole since our model currently can only handle closed shapes. The input dataset of Fig. 5 is obtained by sampling a subdivision surface with 6140 data points.

Results from 2D Multiple Views: Fig. 6(b) is the reconstruction from a synthesized torus, demonstrating topology changes while recovering shape. The original dimensions of the synthesized torus are: $206 \times 205 \times 57$. Compared to ground truth, we get min error of 0.166, max error of 14.231, mean error of 2.544 and RMS of 1.087. Fig. 6(d) is the reconstruction of a mug from 16 real images that proves the ability of recovering the underlying topology from 2-D images. The use of local adaptive refinement on the top allows the model to get the detail of the small grip on the lid of the mug. The multi-resolution procedure has two levels of detail in Fig. 7. Fig. 7(b) is the result after adaptive refinement.

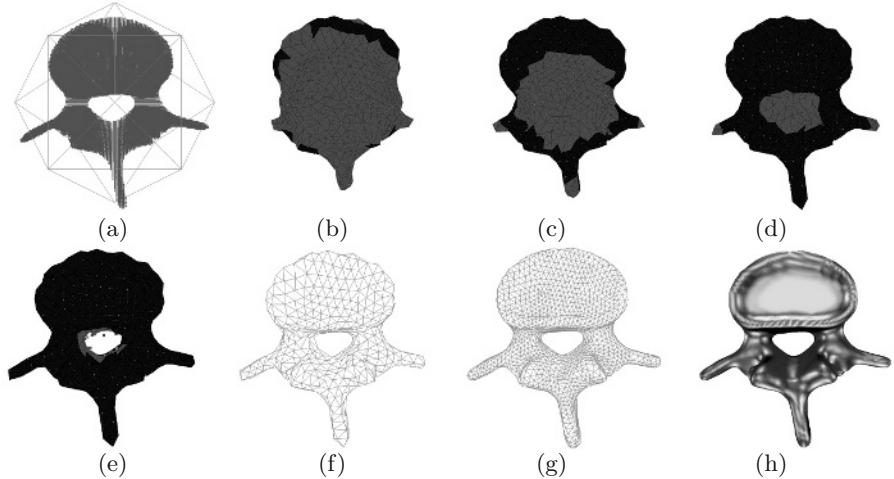


Fig. 3. Segmentation of a CT volumetric dataset of a human vertebra. (a) Initial model and input data; (b)-(e) model evolving; (f) mesh model; (g) optimized mesh model; (h) shaded model.

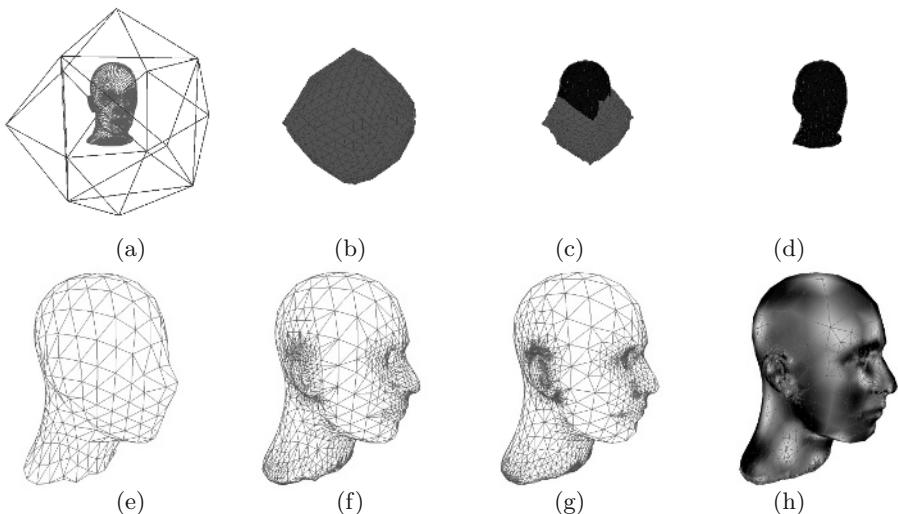


Fig. 4. Reconstruction from point-clouds data of mannequin head. (a) model initialization; (b) and (c) during deformation; (d)(e) final shape of the model; (f)(g) are two different levels of adaptive refinement; (h) is shaded result

It increases the resolution while avoiding global subdivision. Fig. 8(a) shows the positions of 16 real images of a Buddha figure. The 2 images with question marks (one of which is shown in 8(b)) were not used in the reconstruction. Fig. 8(c) is the final texture mapped mesh rendered from a similar viewpoint as the image in 8(b). Fig. 9 shows an example of incremental reconstruction. Fig. 9(a) is the

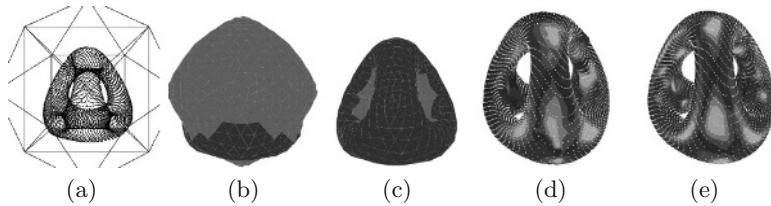


Fig. 5. (a) model initialization; (b) (c) during deformation, dark area stands for non-active, while grey is active; (d) final shape; (e) one level of refinement

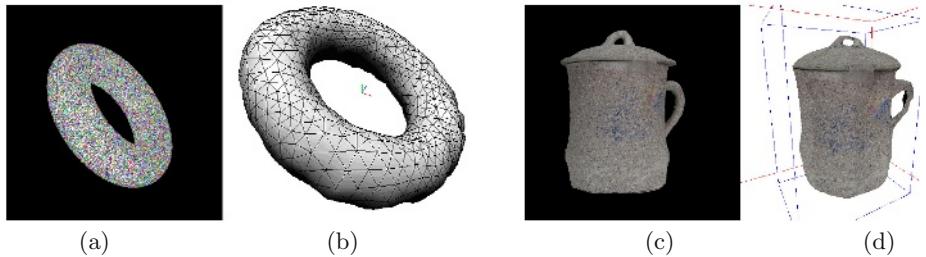


Fig. 6. Fitting to synthesized images with topology changes (a) one of 24 synthetic images of torus; (b) final mesh with 897 vertices after fitting to images (mean error of 2.544); (c) one of 16 real image of a mug; (d) final textured mesh after fitting to the images, the fine detail at the top is the result of local adaptive refinement.

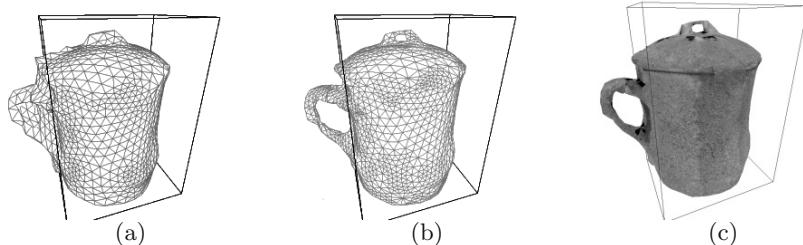


Fig. 7. Adaptive reconstruction. (a) initial reconstruction; (b) reconstruction after one level of adaptive refinement, the two handles of the cup are correctly recovered. (c) textured results from (b)

reconstruction result from 6 frontal images, and 9(b) is its mesh representation. The back of the model has not deformed due to the lack of image data. 9(c) is one of the 5 images added later. After adding images, the model further deforms in 9(d), and finally captures the complete shape shown in 9(e) and 9(f).

Table 1 gives the information of the recovered shape, including the number of vertices, edges and faces for each model, and the running time. The running time is measured on an Intel Pentium 4M 1.6GHZ Notebook PC with 384MB internal memory.

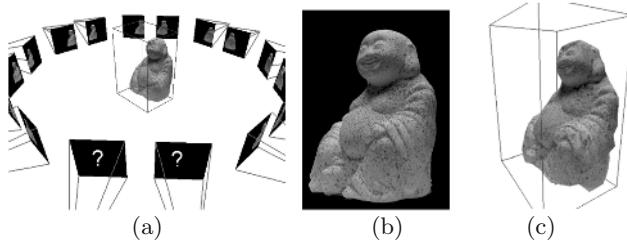


Fig. 8. (a) position of 16 real images of Buddha, the 2 images with question marks were not used in the reconstruction, one of them is shown in (b); (c) final textured mapped mesh rendered from a similar view point as images (b).

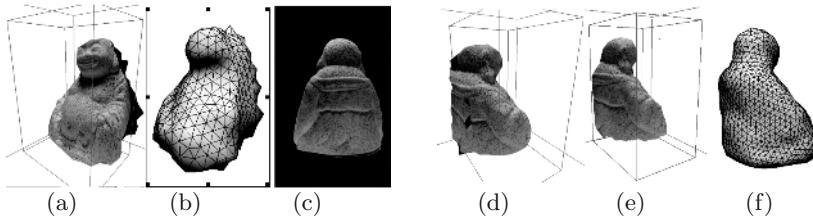


Fig. 9. Incremental reconstruction. (a): partial reconstruction result from 6 frontal images (Black area is not recovered due to lack of image of the data for that part of the model); (b): untextured fitting mesh of (a); (c): one of the 5 images (back views) added later; (d): intermediate result after adding more images; (e) complete reconstruction result; (f) mesh result of (e).

Table 1. Recovered shape information

Figure	Vertices	Faces	Edges	Time (sec)	Figure	Vertices	Faces	Edges	Time (sec)
4(e)	466	928	1392	23	5(d)	723	4774	7161	18
4(f)	3501	6998	10497	19	5(e)	2895	8688	5792	9
4(g)	8386	16768	25152	119	7(a)	1513	3018	4527	223
9(f)	1734	3464	5196	191	7(b)	2770	5540	8310	584

5 Discussion

In this paper, we proposed a new PDE-based deformable surface that is capable of automatically evolving its shape to capture geometric boundaries and simultaneously discover their underlying topological structure. The deformation behavior of the model is governed by partial differential equations, that are derived by the principle of variational analysis. The model ensures regularity and stability, and it can accurately represent sharp features. We have applied our model to shape reconstruction from volumetric data, unorganized point clouds and multi-view 2D images. The characteristics of the model make it especially useful in recovering 3D shape out of 2D multi-view images, handling visibility, occlusion and topology changes explicitly. The existence of a mesh representation

allows progressive refinement of the model when appropriate. Our mathematical formulation allows us to use the same model for different types of data, simply by using the appropriate data interface function. We plan to further exploit this property in future work to apply the model to heterogeneous data such as points, surfels, images and to incorporate other visual cues such as shading and optical flow.

Acknowledgements. This work was partially supported by DOE grant 068, NSF-ITR/NGS grant ACI-0313184, NSF ITR grant IIS-0082035, the NSF grant IIS-0097646, the NSF grant ACI-0328930, the ITR grant IIS-0326388, and Alfred P. Sloan Fellowship. We would like to thank Professor Tim McInerney from University of Toronto for the vertebra dataset and Dr. Hughes Hoppe from Microsoft Research for the mannequin head dataset, respectively.

References

1. N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. *SIGGRAPH*, pages 415–421, 1998.
2. P. Besl and H. McKay. A method for registration of 3d shapes. *PAMI*, 1992.
3. J.S. De Bonet and P. Viola. Poxels. Probabilistic voxelized volume reconstruction. *ICCV*, pages 418–425, 1999.
4. J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. *SIGGRAPH*, 2001.
5. V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *PAMI*, 19, 1997.
6. Charles. Smooth subdivision surfaces based on triangles. Master’s thesis, Mathematics, Univ. of Utah, August 1987.
7. B. Culbertson, T. Malzbender, and G. Slabaugh. Generalized voxel coloring. In *International Workshop on Vision Algorithms*, 1999.
8. D. DeCarlo and D.N. Metaxas. Blended deformable models. *PAMI*, 1996.
9. R. Deriche, C. Bouvin, and O. D. Faugeras. Front propagation and level-set approach for geodesic active stereovision. *ACCV*, 1:640–647, 1998.
10. H. Q. Dinh, G. Slabaugh, and G. Turk. Reconstructing surfaces using anisotropic basis functions. *ICCV*, 2001.
11. H. Edelsbrunner and E.P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72, 1994.
12. O.D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. *ECCV*, pages 379–393, 1998.
13. P. Fua. From multiple stereo views to multiple 3d surfaces. *IJCV*, 24:19–35, 1997.
14. H.K.Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. *VLSM Workshop*, July 2001.
15. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH*, 1992.
16. H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo beyond lambert. *CVPR*, 2003.
17. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, pages 321–331, 1988.

18. R. Kimmel. 3d shape reconstruction from autostereograms and stereo. *Journal of Visual Communication and Image Representation*, 13:324–333, 2002.
19. R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. In *Proceedings of National Academy of Sciences*, pages 8431–8435, July 1998.
20. L. Kobbett, T. Bareuther, and H.-P. Seidel. Multiresolution shape deformations for meshes with dynamic vertex connectivity. *Eurographics*, pages 249–260, 2000.
21. K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *ICCV*, 1999.
22. J.-O. Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine 3d surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999.
23. W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH*, 1997.
24. R. Malladi, J. Sethian, and B. Vemuri. Shape modeling with front propagation: A level set approach. *PAMI*, 17(2), 1995.
25. T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2), 1996.
26. T. McInerney and D. Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *TMI*, pages 840–850, 1999.
27. D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *PAMI*, 15(6), 1993.
28. J.V. Miller, D.E. Breen, W.E. Lorensen, R.M. O'Bara, and M.J. Wozny. Geometric deformed models: a method for extracting closed geometric models from volume data. *SIGGRAPH*, pages 217–226, 1991.
29. M. Pollefeys and L. J. Van Gool. From images to 3d models. *CACM*, 7:50–55, 2002.
30. S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. *Proceedings of 3D Digital Imaging and Modeling*, pages 145–152, 2001.
31. Szymon Rusinkiewicz, Olaf A. Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics*, 21(3):438–446, 2002.
32. S.M. Seitz and C.M. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):1–23, 1999.
33. Gregory G. Slabaugh, Ronald W. Schafer, and Mat C. Hans. Multi-resolution space carving using level sets methods. *ICIP*, 2002.
34. C.J. Taylor and D. Jelinek. Structure and motion from line segments in multiple images. *PAMI*, 17(11), 1995.
35. E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
36. Z. Wood, M. Desbrun, P. Schroder, and D. Breen. Semi-regular mesh extraction from volumes. In *Proceedings of IEEE Visualization*, pages 275–282, 2000.

Structure and Motion Problems for Multiple Rigidly Moving Cameras

Henrik Stewenius and Kalle Åström

Centre for Mathematical Sciences,
Lund University
`{stewe,kalle}@maths.lth.se`

Abstract. Vision (both using one-dimensional and two-dimensional retina) is useful for the autonomous navigation of vehicles. In this paper the case of a vehicle equipped with multiple cameras with non-overlapping views is considered. The geometry and algebra of such a moving platform of cameras are considered. In particular we formulate and solve structure and motion problems for a few novel cases of such moving platforms. For the case of two-dimensional retina cameras (ordinary cameras) there are two minimal cases of three points in two platform positions and two points in three platform positions. For the case of one-dimensional retina cameras there are three minimal structure and motion problems. In this paper we consider one of these (6 points in 3 platform positions). The theory has been tested on synthetic data.

1 Introduction

Vision (both using one-dimensional and two-dimensional retina) is useful for the autonomous navigation of vehicles. An interesting case is here when the vehicle is equipped with multiple cameras with different focal points pointing in different directions.

Our personal motivation for this work stems from the **autonomously guided vehicles**, called **AGV**, which are important components for factory automation. Such vehicles have traditionally been guided by wires buried in the factory floor. This gives a very rigid system. The removal and change of wires are cumbersome and costly. The system can be drastically simplified using navigation methods based on laser or vision sensors and computer vision algorithms. With such a system the position of the vehicle can be computed instantly. The vehicle can then be guided along any feasible path in the room.

Note that the discussion here is focused on finding initial estimates of structure and motion. In practice it is necessary to refine these estimates using non-linear optimisation or bundle adjustment, cf. [Sla80, Åst96].

Structure and motion recovery from a sequence of images is a classical problem within computer vision. A good overview of the techniques available for structure and motion recovery can be found in [HZ00]. Much is known about minimal cases, feature detection, tracking and structure and motion recovery

has been built, [Nis01,PKVG98]. Such systems are however difficult to build. There are ambiguous configurations [KHÅ01] for which structure and motion recovery is impossible. Many automatic systems rely on small image motions in order to solve the correspondence problem. In combination with most cameras small field of view, this limits the way the camera can be moved in order to make good 3D reconstruction. The problem is significantly more stable with a large field of view [ÖA98]. Recently there have been attempts at using rigs with several simple cameras in order to overcome this problem, cf. [Ple03,BFA01]. In this paper we study and solve some of the minimal cases for multi-camera platforms. Such solutions are necessary components for automatic structure and motion recovery systems.

The paper is organised as follows. In section 2 there are some common theory and problem formulation and in the following sections, the studies of three minimal cases.

2 The Geometry of Vision from a Moving Platform

In this paper we consider a platform (a vehicle, robot, car) moving with a planar motion. This vehicle has a number of cameras with different camera centres facing outwards so that they cover different viewing angles. The purpose here is to get a large combined field of view using simple and cheap cameras. The cameras are assumed to have different camera centres but it is assumed that the cameras are calibrated relative to the vehicle, i.e. it is assumed that the camera matrices \mathbf{P}_i are known for all cameras on the vehicle. It is known that a very large field of view improves stability [BFA01] and Pless derives the basic equations to deal with multiple cameras [Ple03].

As both 1D and 2D-retina cameras are studied the equations of both will be introduced in parallel, 1D cameras on the left hand and 2D on the right hand side of the paper.

Both types can be modeled $\lambda\mathbf{u} = \mathbf{PU}$, where the camera matrix \mathbf{P} is a 2×3 or 3×4 matrix. A scene point \mathbf{U} is in P^2 or P^3 and a measured image point

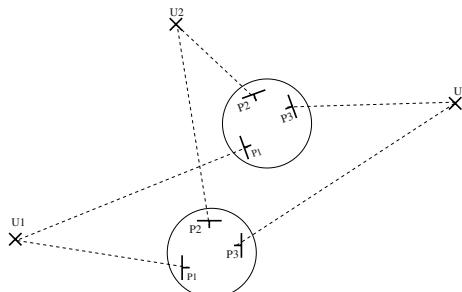


Fig. 1. Three calibrated cameras with constant and known relative positions taking two images each

\mathbf{u} is in P^1 or P^2 . It is sometimes useful to consider dual image coordinates. In the one-dimensional retina case each image point \mathbf{u} is dual to a vector \mathbf{v} , with $\mathbf{v}\mathbf{u} = 0$ and in the 2D case there are two linearly independent dual vectors \mathbf{v}^1 and \mathbf{v}^2 with $\mathbf{v}^i\mathbf{u} = 0$. The measurement equation then becomes

$$\mathbf{v}\mathbf{P}\mathbf{U} = 0 \quad \text{or} \quad \mathbf{v}^1\mathbf{P}\mathbf{U} = \mathbf{v}^2\mathbf{P}\mathbf{U} = 0.$$

As the platform moves the cameras move together. This is modeled as a transformation \mathbf{S}_i between the first position and position i . In the original coordinate system the camera matrix for camera j at position i is $\mathbf{P}_j\mathbf{S}_i$.

It is assumed here that the camera views do not necessarily have common points. In other words, a point is typically seen by only one camera. On the other hand it can be assumed that in a couple of neighboring frames a point can be seen in the same camera. Assume here that point j is visible in camera j . The measurement equation for the n points is then

$$\lambda_{ij}\mathbf{u}_{ij} = \mathbf{P}_j\mathbf{S}_i\mathbf{U}_j, \quad j = 1, \dots, n, i = 1, \dots, m.$$

Using the dual image coordinates we obtain

$$\mathbf{v}_{ij}\mathbf{P}_j\mathbf{S}_i\mathbf{U}_j = 0 \quad \text{or} \quad \begin{cases} \mathbf{v}_{ij}^1\mathbf{P}_j\mathbf{S}_i\mathbf{U}_j = 0, \\ \mathbf{v}_{ij}^2\mathbf{P}_j\mathbf{S}_i\mathbf{U}_j = 0, \end{cases} \quad j = 1, \dots, n, i = 1, \dots, m,$$

for the one respectively two-dimensional retina case.

Note that $\mathbf{l}_{ij}^T = \mathbf{v}_{ij}\mathbf{P}_j$ and $\mathbf{l}_{ij}^{kT} = \mathbf{v}_{ij}^k\mathbf{P}_j$ correspond to the viewing line or viewing plane in the vehicle coordinate system. Thus the constraint can be written

$$\mathbf{l}_{ij}\mathbf{S}_i\mathbf{U}_j = 0 \quad \text{or} \quad \begin{cases} \mathbf{l}_{ij}^1\mathbf{S}_i\mathbf{U}_j = 0, \\ \mathbf{l}_{ij}^2\mathbf{S}_i\mathbf{U}_j = 0, \end{cases} \quad j = 1, \dots, n, i = 1, \dots, m. \quad (1)$$

Here the lines or planes \mathbf{l} are measured. The question is if one can calculate structure \mathbf{U}_j and motion \mathbf{S}_i from these measurements. Based on the previous sections, the structure and motion problem will now be defined.

Problem 1. Given the mn images \mathbf{u}_{ij} of n points from m different platform positions and the camera matrices \mathbf{P}_j the **surveying problem** is to find reconstructed points \mathbf{U}_j and platform transformations \mathbf{S}_i such that

$$\lambda_{ij}\mathbf{u}_{ij} = \mathbf{P}_j\mathbf{S}_i\mathbf{U}_j, \quad \forall i = 1, \dots, m, j = 1, \dots, n$$

for some λ_{ij} .

2.1 Minimal Cases

In order to understand how much information is needed in order to solve the structure and motion problem, it is useful to calculate the number of degrees of freedom of the problem and the number of constraints given by the projection

equation. Each object point has two degrees of freedom in the two-dimensional world and three in the three-dimensional world. Vehicle location for a planarily moving vehicle has three degrees of freedom when using $a_i^2 + b_i^2 = 1$, that is Euclidian information and four degrees of freedom in the similarity case when not using this information. The word “image” is used to mean all the information collected by all our cameras at one instant in time.

For Euclidian reconstruction in three dimensions there are $2mn - (3n + 3(m - 1))$ excess constraints and as seen in table 1 there are two interesting cases

1. two images and three points ($m=2, n=3$).
2. three images and two points ($m=3, n=2$).

For the similarity case in two dimensions there are $mn - (2n + 4(m - 1))$ excess constraints and as seen in table 1 there are three interesting cases

1. three images of eight points ($m=3, n=8$).
2. four images of six points ($m=4, n=6$).
3. six images of five points ($m=6, n=5$).

For the Euclidean case in two dimensions there are $mn - (2n + 3(m - 1))$ excess constraints and as seen in table 1 there are three interesting cases

1. three images of six points ($m=3, n=6$).
2. four images of five points ($m=4, n=5$) overdetermined.
3. five images of four points ($m=5, n=4$).

All these will be called the **minimal cases of the structure and motion problem**.

Table 1. The number of excess constraints

2D Similarity		2D Euclidian		3D Euclidian	
n	m	n	m	n	m
1	1 2 3 4 5 6 7	1	1 -1 -3 -5 -7 -9 -11 -13	1	-1 -2 -3 -4 -5 -6 -7
2	-2 -4 -6 -8 -10 -12 -14	2	-2 -3 -4 -5 -6 -7 -8	2	-2 -1 0 1 2 3 4
3	-3 -4 -5 -6 -7 -8 -9	3	-3 -3 -3 -3 -3 -3 -3	3	-3 0 3 6 9 12 15
4	-4 -4 -4 -4 -4 -4 -4	4	-4 -3 -2 -1 0 1 2	4	-4 1 6 11 16 21 26
5	-5 -4 -3 -2 -1 0 1	5	-5 -3 -1 1 3 5 7	5	-5 2 9 16 23 30 37
6	-6 -4 -2 0 2 4 6	6	-6 -3 0 3 6 9 12	6	-6 3 12 21 30 39 48
7	-7 -4 -1 2 5 8 11	7	-7 -3 1 5 9 13 17	7	-7 4 15 26 37 48 59
8	-8 -4 0 4 8 12 16	8	-8 -3 2 7 12 17 22	8	-8 5 18 31 44 57 70
9	-9 -4 1 6 11 16 21	9	-9 -3 3 9 15 21 27	9	-9 6 21 36 51 66 81

3 Two-Dimensional Retina, Two Positions, and Three Points

Theorem 1. *For three calibrated cameras that are rigidly fixed with a known transformation relative to each other, each taking an image of a point at two distinct times there generally exist one or three real non-trivial solutions.*

Pure translation is a degenerate case and can only be computed up to scale.

Proof. See solution procedure.

Departing from equation (1) for two observations of point j gives

$$\underbrace{\begin{bmatrix} \mathbf{l}_{1j}^1 \mathbf{S}_1 \\ \mathbf{l}_{1j}^2 \mathbf{S}_1 \\ \mathbf{l}_{2j}^1 \mathbf{S}_2 \\ \mathbf{l}_{2j}^2 \mathbf{S}_2 \end{bmatrix}}_{M_j} \mathbf{U}_j = \mathbf{0}.$$

As $\mathbf{U}_j \neq \mathbf{0}$ there is a nonzero solution to the above homogeneous linear system i.e.

$$\det M_j = 0. \quad (2)$$

Note that the constraint above is in essence the same as in [Ple03]. The same constraint can be formulated as $L_1^T F L_2 = 0$, where L_1 and L_2 are plücker coordinate vectors for the space lines and F is a 6×6 matrix representing relative motion of the two platforms.

By a suitable choice of coordinate system it can be assumed that $\mathbf{S}_0 = I$, that is, $a_0 = b_0 = c_0 = d_0 = 0$ and to reduce the number of indices we set $a = a_1, b = b_1, c = c_1$ and $d = d_1$. The planes defined by $\mathbf{l}_{1j}^1, \mathbf{l}_{1j}^2, \mathbf{l}_{2j}^1$ and \mathbf{l}_{2j}^2 all comes from in camera j and pass through this camera centre in the vehicle coordinate system, that is through a common point, implying

$$\det \begin{bmatrix} \mathbf{l}_{1j}^1 \\ \mathbf{l}_{1j}^2 \\ \mathbf{l}_{2j}^1 \\ \mathbf{l}_{2j}^2 \end{bmatrix} = 0.$$

Computing the determinant in equation (2) gives

$$\alpha_a a + \alpha_{aj} b + \alpha_{cj} c + \alpha_{dj} d + \alpha_{A_j} (da + cb) + \alpha_{B_j} (db - ca) = 0$$

where $\alpha_j = \alpha(\mathbf{l}_{1j}^1, \mathbf{l}_{1j}^2, \mathbf{l}_{2j}^1, \mathbf{l}_{2j}^2)$. The assumption of rigid movement is equivalent to $(1+a)^2 + b^2 - 1 = 0$. With three points observed, one in each of the three cameras the above gives four polynomials in (a, b, c, d) . After homogenisation with t the polynomial equations are

$$\begin{cases} f_1 = \alpha_{a1}at + \alpha_{b1}bt + \alpha_{c1}ct + \alpha_{d1}dt + \alpha_{A1}(ad + bc) + \alpha_{B1}(ac - bd) = 0 \\ f_2 = \alpha_{a2}at + \alpha_{b2}bt + \alpha_{c2}ct + \alpha_{d2}dt + \alpha_{A2}(ad + bc) + \alpha_{B2}(ac - bd) = 0 \\ f_3 = \alpha_{a3}at + \alpha_{b3}bt + \alpha_{c3}ct + \alpha_{d3}dt + \alpha_{A3}(ad + bc) + \alpha_{B3}(ac - bd) = 0 \\ f_4 = a^2 + 2at + b^2 = 0. \end{cases} \quad (3)$$

Lemma 1. $(a, b, c, d, t) = (0, 0, 0, 0, 1)$ and $(a, b, c, d, t) = \lambda(1, \pm i, 0, 0, 0)$ are solutions to equation (3).

Proof. This follows by inserting the solutions in the equations.

3.1 Solving

Equation (3) is solved by using different combinations of the original equations to construct a polynomial in b which gives roots that can be used to solve for the other variables.

As equation (3) is homogeneous it can be de-homogenised by studying two cases, $a = 0$ and $a = 1$.

$\mathbf{a} = \mathbf{0}$ gives $b = 0$ (by $f_4 = 0$). Now remains

$$\begin{cases} t(\alpha_{c1}c + \alpha_{d1}d) = 0 \\ t(\alpha_{c2}c + \alpha_{d2}d) = 0 \\ t(\alpha_{c3}c + \alpha_{d3}d) = 0, \end{cases}$$

which in general has only the trivial solutions $(c, d, t) = (0, 0, t)$ and $(c, d, t) = (c, d, 0)$. If the 3 equations are linearly dependent more solutions exist on the form $(c, d, t) = (k_c s, k_d s, t)$. This means that pure translation can only be computed up to scale.

$\mathbf{a} = \mathbf{1}$ gives

$$\begin{cases} f_i = \alpha_{ia}t + \alpha_{ib}bt + (\alpha_{ic} + \alpha_{iB})ct + (\alpha_{id} + \alpha_{iA})dt \\ \quad + \alpha_{iA}(d + bc) + \alpha_{iB}(c - bd) = 0 & i = 1, 2, 3 \\ f_4 = 1 + 2t + b^2 = 0. \end{cases}$$

The 24 coefficients of the 17 polynomials

$$\begin{aligned} f_i, f_ib, f_ib^2, & \quad i = 1, 2, 3 \\ f_4db, f_4d, f_4cb, f_4c, f_4b^3, f_4b^2, f_4b, f_4b \end{aligned}$$

are ordered in the lex order $t < d < c < b$ into a 17×24 matrix M with one polynomial per row. As no solutions are lost by multiplying by a polynomial and the original polynomials are in the set, this implies

$$MX = \mathbf{0}, \tag{4}$$

where $X = [X_1^T \ X_2^T]^T$ and

$$\begin{aligned} X_1 &= [tdb^2, tdb, td, tcb^2, tcb, tc, tb^3, tb^2, tb, t, db^3, db^2, db, d, cb^3, cb^2, cb, c]^T \\ X_2 &= [b^5, b^4, b^3, b^2, b, 1]^T. \end{aligned}$$

Dividing $M = [M_1 \ M_2]$ where M_1 is 17×18 and M_2 is 17×6 , equation (4) can be written $M_1 X_1 + M_2 X_2 = 0$. Unfortunately these pages are too few to

show the resulting matrices but it is easily proven that $\text{rank } M_1 \leq 16$, the easiest way to do this is to use the fact that $\text{rank } M_1 = \text{rank } M_1^T$ and it is easy to find two vectors in the null-space of M_2^T . Therefore there exist v s.t. $vM_1 = \mathbf{0}$ and $\mathbf{0} = \mathbf{0}X = vX = vM_1X_1 + vM_2X_2 = vM_2X_2$ that is, a fifth order polynomial $p_1(b) = vM_2X_2 = 0$. Knowing that $b = \pm i$ are solutions to this equation, a new polynomial $p_2(b) = p_1(b)/(1+b^2)$ can be calculated and then solved for the remaining roots.

It is then possible to use f_4 to solve for t . Knowing t it is possible to change back to the original non-homogeneous variables (a, b, c, d) and solve for c and d .

3.2 Implementation and Code

The implementation is quite straightforward and the code is available for download [Ste]. As f_4 have fixed coefficients and the result is divided by $(1+b^2)$ it is possible to reduce M to a 9×12 matrix with M_1 of size 9×10 or 9×8 if the missing rank is used as well. M_2 is of size 9×6 but can be simplified to 9×4 as the roots $b = \pm i$ are known. Reducing the size of M is useful as the highest number of computations come from finding the null-space of M_1 .

4 Two-Dimensional Retina, Three Positions, and Two Points

Theorem 2. *Two cameras mounted rigidly with a known transformation with respect to each other for which calibration as well as relative positions are known are moved planarily to 3 different stations where they observe one point per camera. Under these circumstances there generally exist one or three non-trivial real solutions.*

Pure translation is a degenerate case and can only be computed up to scale.

Proof. The existence of a solver giving that number of solutions.

A point j observed in 3 images of the same camera gives

$$\underbrace{\begin{bmatrix} l_{0j}^1 \mathbf{S}_0 \\ l_{0j}^2 \mathbf{S}_0 \\ l_{1j}^1 \mathbf{S}_1 \\ l_{1j}^2 \mathbf{S}_1 \\ l_{2j}^1 \mathbf{S}_2 \\ l_{2j}^2 \mathbf{S}_2 \end{bmatrix}}_{M_j} \mathbf{U}_j = \mathbf{0}.$$

As a non-zero solution exists $\text{rank } M_j \leq 3$. This is equivalent to the fact that $\det(M_{sub}) = 0$ for all 4×4 sub-matrices of M . With a suitable choice of coordinate system $\mathbf{S}_0 = I$, that is, $a_0 = b_0 = c_0 = d_0 = 0$ and the 15 sub-matrices of

M_j can be computed and this gives 15 polynomial equations of the second and third degree in $(a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2)$. Rigid planar motion implies

$$a_i^2 + 2a_i + b_i^2 = 0 \quad i = 1, 2.$$

Observing two points through different but relatively fixed cameras cameras at 3 instants gives total of 32 equations in 8 unknowns.

4.1 Solving

A solver inspired by the previous case has been built but it is still very slow. Anyhow it generates a fifth order polynomial in b_1 for which we know the solutions $b_1 = \pm i$.

If $a_1 = a_2 = 0$ then $b_1 = b_2 = 0$ and the system is degenerate and the solution for (c_1, d_1, c_2, d_2) can only be computed up to scale.

4.2 Implementation and Code

The solver is still in an early stage and very slow. The matrix M will in this case be much larger than in the previous case but the simplifications used there will be possible here as well. As soon as a decent implementation is ready it will be available for download.

5 One-Dimensional Retina, Three Positions, and Six/Eight Points

5.1 Intersection and the Discrete Trilinear Constraint

In this section we will try to use the same technique for solving the structure and motion problem as in [ÅO00]. The idea is to study the equations for a particular point for three views. The fact that three planar lines intersect in a point gives a constraint that we are going to study.

The case of three cameras is of particular importance. Using three measured bearings from three different known locations, the object point is found by intersecting three lines. This is only possible if the three lines actually do intersect. This gives an additional constraint, which can be formulated in the following way

Theorem 3. *Let $\mathbf{l}_{1,j}$, $\mathbf{l}_{2,j}$ and $\mathbf{l}_{3,j}$ be the bearing directions to the same object point from three different camera states. Then the trilinear constraint*

$$\sum_{p,q,r} \mathbf{T}^{pqr} \mathbf{l}_{1j,p} \mathbf{l}_{2j,q} \mathbf{l}_{3j,r} = 0, \quad (5)$$

is fulfilled for some $3 \times 3 \times 3$ tensor \mathbf{T} .

Proof. By lining up the line equations

$$\underbrace{\begin{pmatrix} \mathbf{l}_{1,j}\mathbf{S}_1 \\ \mathbf{l}_{2,j}\mathbf{S}_2 \\ \mathbf{l}_{3,j}\mathbf{S}_3 \end{pmatrix}}_M (\mathbf{U}_j) = \mathbf{0}$$

we see that the 3×3 matrix M has a non-trivial right-nullspace. Therefore its determinant is zero. Since the determinant is linear in each row it follows that it can be written as

$$\det M = \sum_{p,q,r} \mathbf{T}^{pqr} \mathbf{l}_{1j,p} \mathbf{l}_{2j,q} \mathbf{l}_{3j,r} = 0,$$

for some $3 \times 3 \times 3$ tensor \mathbf{T} . Here $\mathbf{l}_{1j,p}$ denotes element p of vector \mathbf{l}_{1j} and similarly for $\mathbf{l}_{2j,q}$ and $\mathbf{l}_{3j,r}$.

The tensor $\mathbf{T} = \mathbf{T}^{pqr}$ in (5) will now be analysed in more detail. The mapping from the motion parameters $(\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3)$ to the tensor \mathbf{T} is invariant to changes of the coordinate system, i.e. by multiplying each of the transformation matrices with the same matrix. Thus without loss of generality one may assume that $\mathbf{S}_1 = I$. Introducing parameterisations according to (1) the tensor components are

$$\left\{ \begin{array}{lll} \mathbf{T}^{111} = b_2 c_3 - b_3 c_2, & \mathbf{T}^{112} = b_2 d_3 - a_3 c_2, & \mathbf{T}^{113} = b_2, \\ \mathbf{T}^{121} = a_2 c_3 - b_3 d_2, & \mathbf{T}^{122} = a_2 d_3 - a_3 d_2, & \mathbf{T}^{123} = a_2, \\ \mathbf{T}^{131} = -b_3, & \mathbf{T}^{132} = -a_3, & \mathbf{T}^{133} = 0, \\ \mathbf{T}^{211} = -a_2 c_3 + a_3 c_2, & \mathbf{T}^{212} = -a_2 d_3 - b_3 c_2, & \mathbf{T}^{213} = -a_2, \\ \mathbf{T}^{221} = b_2 c_3 + a_3 d_2, & \mathbf{T}^{222} = b_2 d_3 - b_3 d_2, & \mathbf{T}^{223} = b_2, \\ \mathbf{T}^{231} = a_3, & \mathbf{T}^{232} = -b_3, & \mathbf{T}^{233} = 0, \\ \mathbf{T}^{311} = a_2 b_3 - a_3 b_2, & \mathbf{T}^{312} = a_2 a_3 + b_3 b_2, & \mathbf{T}^{313} = 0, \\ \mathbf{T}^{321} = -b_3 b_2 - a_2 a_3, & \mathbf{T}^{322} = a_2 b_3 - a_3 b_2, & \mathbf{T}^{323} = 0, \\ \mathbf{T}^{331} = 0, & \mathbf{T}^{332} = 0, & \mathbf{T}^{333} = 0. \end{array} \right. \quad (6)$$

Note that the tensors have a number of zero components. It can be shown that there are 15 linearly independent linear constraints on the tensor components. These are

$$\left\{ \begin{array}{l} \mathbf{T}^{133} = \mathbf{T}^{233} = \mathbf{T}^{313} = \mathbf{T}^{323} = \mathbf{T}^{331} = \mathbf{T}^{332} = \mathbf{T}^{333} = 0, \\ \mathbf{T}^{131} - \mathbf{T}^{232} = \mathbf{T}^{132} + \mathbf{T}^{231} = \mathbf{T}^{113} - \mathbf{T}^{223} = 0 \\ \mathbf{T}^{123} + \mathbf{T}^{213} = \mathbf{T}^{311} - \mathbf{T}^{322} = \mathbf{T}^{321} + \mathbf{T}^{312} = 0, \\ \mathbf{T}^{111} - \mathbf{T}^{122} - \mathbf{T}^{212} - \mathbf{T}^{221} = 0, \\ \mathbf{T}^{112} - \mathbf{T}^{121} - \mathbf{T}^{211} - \mathbf{T}^{222} = 0. \end{array} \right. \quad (7)$$

There are also four non-linear constraints, i.e.

$$\mathbf{T}^{312}(\mathbf{T}^{123}\mathbf{T}^{131} + \mathbf{T}^{231}\mathbf{T}^{223}) + \mathbf{T}^{311}(\mathbf{T}^{123}\mathbf{T}^{231} - \mathbf{T}^{131}\mathbf{T}^{223}) = 0, \quad (8)$$

$$\begin{aligned} -\mathbf{T}^{231}\mathbf{T}^{223}\mathbf{T}^{131}\mathbf{T}^{221} + \mathbf{T}^{223}\mathbf{T}^{231}\mathbf{T}^{231}\mathbf{T}^{121} - \mathbf{T}^{123}\mathbf{T}^{231}\mathbf{T}^{231}\mathbf{T}^{111} \\ + \mathbf{T}^{231}\mathbf{T}^{223}\mathbf{T}^{131}\mathbf{T}^{111} - \mathbf{T}^{123}\mathbf{T}^{131}\mathbf{T}^{131}\mathbf{T}^{221} + \mathbf{T}^{123}\mathbf{T}^{131}\mathbf{T}^{231}\mathbf{T}^{121} \\ + \mathbf{T}^{123}\mathbf{T}^{131}\mathbf{T}^{231}\mathbf{T}^{211} - \mathbf{T}^{223}\mathbf{T}^{131}\mathbf{T}^{131}\mathbf{T}^{211} = 0, \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbf{T}^{131}\mathbf{T}^{123}\mathbf{T}^{222} - \mathbf{T}^{131}\mathbf{T}^{121}\mathbf{T}^{123} - \mathbf{T}^{131}\mathbf{T}^{123}\mathbf{T}^{211} + \mathbf{T}^{231}\mathbf{T}^{111}\mathbf{T}^{123} \\ - \mathbf{T}^{131}\mathbf{T}^{122}\mathbf{T}^{223} - \mathbf{T}^{231}\mathbf{T}^{121}\mathbf{T}^{223} = 0, \end{aligned} \quad (10)$$

$$\begin{aligned} \mathbf{T}^{231}\mathbf{T}^{123}\mathbf{T}^{123}\mathbf{T}^{222} - \mathbf{T}^{123}\mathbf{T}^{231}\mathbf{T}^{122}\mathbf{T}^{223} - \mathbf{T}^{123}\mathbf{T}^{231}\mathbf{T}^{221}\mathbf{T}^{223} \\ - \mathbf{T}^{131}\mathbf{T}^{123}\mathbf{T}^{123}\mathbf{T}^{221} - \mathbf{T}^{223}\mathbf{T}^{131}\mathbf{T}^{123}\mathbf{T}^{222} + \mathbf{T}^{131}\mathbf{T}^{122}\mathbf{T}^{223}\mathbf{T}^{223} \\ + \mathbf{T}^{231}\mathbf{T}^{121}\mathbf{T}^{223}\mathbf{T}^{223} + \mathbf{T}^{223}\mathbf{T}^{131}\mathbf{T}^{121}\mathbf{T}^{123} = 0. \end{aligned} \quad (11)$$

If only Euclidean transformations of the platform are allowed, which is reasonable, there are two additional constraints

$$(\mathbf{T}^{123})^2 + (\mathbf{T}^{113})^2 = a_2^2 + b_2^2 = 1, \quad (12)$$

$$(\mathbf{T}^{231})^2 + (\mathbf{T}^{232})^2 = a_3^2 + b_3^2 = 1. \quad (13)$$

These two last constraints are true only if the tensors are considered to be normalised with respect to scale. It is straightforward to generate corresponding non-normalised (and thus also homogeneous) constraints.

It is natural to think of the tensor as being defined only up to scale. Two tensors \mathbf{T} and $\tilde{\mathbf{T}}$ are considered equal if they differ only by a scale factor

$$\mathbf{T} \sim \tilde{\mathbf{T}}.$$

Let \mathcal{T} denote the set of equivalence classes of trilinear tensors fulfilling equations (7)-(13).

Definition 1. Let the manifold of relative motion of three platform positions be defined as the set of equivalence classes of three transformations

$$\mathcal{M}_S = \left\{ (\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3) \mid \mathbf{S}_I = \begin{pmatrix} a_I & b_I & c_I \\ -b_I & a_I & d_I \\ 0 & 0 & 1 \end{pmatrix} \right\} / \simeq,$$

where the equivalence is defined as

$$(\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3) \simeq (\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2, \tilde{\mathbf{S}}_3), \quad \exists \mathbf{S} \in \mathcal{S}, \tilde{\mathbf{S}}_I \sim \mathbf{S}_I \mathbf{S}, I = 1, 2, 3.$$

Thus the above discussion states that the map $(\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3) \mapsto \mathbf{T}$ is in fact a well defined map from the manifold of equivalence classes \mathcal{M}_S to \mathcal{T} .

Theorem 4. A tensor \mathbf{T}^{pqr} is a calibrated trilinear tensor if and only if equations (7)-(13) are fulfilled. When these constraints are fulfilled it is possible to solve (6) for \mathbf{S}_2 , \mathbf{S}_3 . The solution is in general unique.

Corollary 1. The map

$$\mathbf{T} : \mathcal{M}_S \longrightarrow \mathcal{T}$$

is a well defined one-to-one mapping.

5.2 Algorithm

The previous section on the calibrated trilinear tensor has provided us with the tool for solving the structure and motion problem for three platform positions of at least eight points.

Algorithm 51 (Structure and Motion from Three Platform Motions)

1. *Given three images of at least six points,*

$$\mathbf{u}_{ij}, \quad i = 1, \dots, 3, j = 1, \dots, n, n \geq 6.$$

2. *Calculate all possible trilinear tensors \mathbf{T} that fulfills the constraints (7) to (13) and $\sum_{p,q,r} \mathbf{T}^{pqr} \mathbf{l}_{1j,p} \mathbf{l}_{2j,q} \mathbf{l}_{3j,r} = 0, \forall j = 1, \dots, n$.*
3. *Calculate the platform motions $(\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3)$ from \mathbf{T} according to the proof of Theorem 4.*
4. *For each solution to the motion calculate structure using intersection.*

5.3 Homotopy Studies of Six Points in Three Stations

In step 2 of the above algorithm one has to find all solutions to a system of polynomial equations. We have not yet solved this system in detail, but rather experimented with simulated data and a numerical solver of such system of equations.

When studying the equations of six points and three stations in two dimensions under Euclidian assumption ($a_i^2 + b_i^2 = 1$) in the dual points formulation there are 8 unknowns and 8 variables. By inserting these equations into the homotopy software **PHC-pack** [Ver99] it is found that the mixed volume [CLO98] is 39 and there are up to 25 real solutions.

Based on these experimental investigations we postulate that there are in general up to 25 real solutions to the problem of 6 points in 3 images.

6 Conclusions

In this paper we have introduced the structure and motion problem for the notion of a platform of moving cameras. Three particular cases, (i) eight points in three one-dimensional views, (ii) three points in two two-dimensional views and (iii) two points in three two-dimensional views have been studied. Solutions to these problems are useful for structure and motion estimation of autonomous vehicles equipped with multiple cameras.

The existence of a fast solver for the two images and three points case in three dimensions is of interest when computing RANSAC. It is important to note that pure translation is a degenerate case and that the solution in this case suffers from the same unknown scale as for single camera solutions. Another important aspect is that the cameras has to have separate focal points.

References

- [ÅO00] K. Åström and M. Oskarsson. Solutions and ambiguities of the structure and motion problem for 1d retinal vision. *Journal of Mathematical Imaging and Vision*, 12(2):121–135, 2000.
- [Åst96] K. Åström. *Invariancy Methods for Points, Curves and Surfaces in Computational Vision*. PhD thesis, Dept of Mathematics, Lund University, Sweden, 1996.
- [BFA01] P. Baker, C. Fernmuller, and Y. Aloimonos. A spherical eye from multiple cameras (makes better models of the world). In *Proc. Conf. Computer Vision and Pattern Recognition, Hawaii, USA*, 2001.
- [CLO98] D. Cox, J. Little, and D. O’Shea. *Using Algebraic Geometry*. Springer Verlag, 1998.
- [GH97] R. Gupta and R. I. Hartley. Linear pushbroom cameras. *Pattern Analysis and Machine Intelligence*, 19(9):963–975, sep 1997.
- [HZ00] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [KHÅ01] F. Kahl, R. Hartley, and K. Åström. Critical configurations for n-view projective reconstruction. In *Proc. Conf. Computer Vision and Pattern Recognition, Hawaii, USA*, 2001.
- [MAS97] C. B. Madsen, C. S. Andersen, and J. S. Sørensen. A robustness analysis of triangulation-based robot self-positioning. In *The 5th Symposium for Intelligent Robotics Systems, Stockholm, Sweden*, 1997.
- [Nis01] D. Nistér. *Automatic Dense Reconstruction from Uncalibrated Video Sequences*. PhD thesis, Dept. of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 2001.
- [NRT97] J. Neira, I. Ribeiro, and J. D. Tardos. Mobile robot localization and map building using monocular vision. In *The 5th Symposium for Intelligent Robotics Systems, Stockholm, Sweden*, pages 275–284, 1997.
- [OÅ98] M. Oskarsson and K. Åström. Accurate and automatic surveying of beacon positions for a laser guided vehicle. In *Proc. European Consortium for Mathematics in Industry, Gothenburg, Sweden*, 1998.
- [PKVG98] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, 1998.
- [Ple03] R. Pless. Using many cameras as one. In *Proc. Conf. Computer Vision and Pattern Recognition, Madison, Canada*, 2003.
- [Sla80] C. C. Slaama. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, 1980.
- [Ste] H. Stewenius. Homepage with code for planar multicamera navigation. <http://www.maths.lth.se/~stewe/as3/>.
- [Tri95] B. Triggs. Matching constraints and the joint image. In *Proc. 5th Int. Conf. on Computer Vision, MIT, Boston, MA*, pages 338–343, 1995.
- [Ver99] J. Verschelde. Phcpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2):251–276, 1999.

Detection and Tracking Scheme for Line Scratch Removal in an Image Sequence

Bernard Besserer and Cedric Thiré

Laboratoire Informatique, Image, Interaction (L3i), Université La Rochelle,
17042 La Rochelle cedex 1, France
{bernard.besserer,cedric.thire}@univ-lr.fr
<http://www.univ-lr.fr>

Abstract. A detection and tracking approach is proposed for line scratch removal in a digital film restoration process. Unlike random impulsive distortions such as dirt spots, line scratch artifacts persist across several frames. Hence, motion compensated methods will fail, as well as single-frame methods if scratches are unsteady or fragmented.

The proposed method uses as input projections of each image of the input sequence. First, a 1D-extrema detector provides candidates. Next, a MHT (Multiple Hypothesis Tracker) uses these candidates to create and keep multiple hypothesis. As the tracking goes further through the sequence, each hypothesis gains or loses evidence. To avoid a combinatorial explosion, the hypothesis tree is sequentially pruned, preserving a list of the best ones. An energy function (quality of the candidates, comparison to a model) is used for the path hypothesis sorting. As hypotheses are set up at each iteration, even if no information is available, a tracked path might cross gaps (missed detection or speckled scratches). At last, the tracking stage feeds the correction process. Since this contribution focus on the detection stage, only tracking results are given.

1 Introduction

Despite of fast-growing use of digital media, the photochemical film is still the storage base in the motion picture industry and several million reels are stored at film archives. Film is a good medium for long term storage, but future mass-migration to digital media is ineluctable and digital processing at this step could ensure the removal of the various, typical film-related damages, see figure 1. Though traditional restoration techniques are necessary (the film should be able to withstand mechanically the digitisation step), digital restoration lets us expect results beyond today's limitations (automated processing, correction of previously photographed artifacts, etc.).

Digital restoration has only very recently been explored [1,2,3]. The main visual defects are dust spots, hairs and dirt, instabilities (both exposition and position) and scratches, some of them are now easily detected and removed, especially if the defect appears only in a single frame. This is not the case for scratches. Scratches are mainly vertical (parallel to the film transport direction),

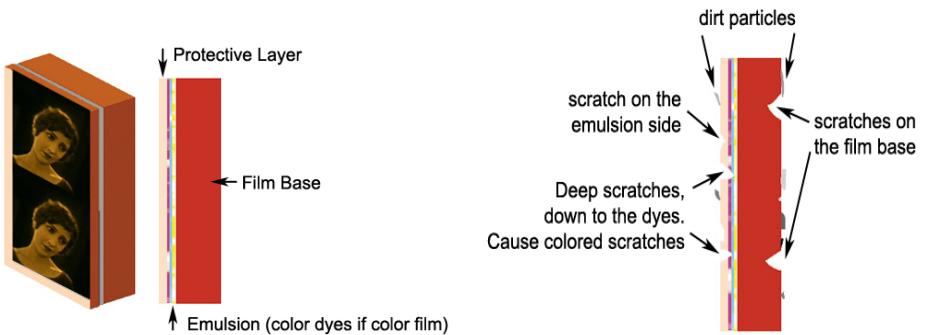


Fig. 1. Film structure and film damage

caused by slippage and abrasion during fast starts, stops and rewinding. Because the scratch is spread over many frames, and appears at the same location during projection, this damage is readily seen by the viewer, and also difficult to detect and correct using image processing.¹

Early work about digital line scratch removal can be related to Anil C. Kokaram's research activities [4,5,6]. His detection scheme, based on vertical mean, is still used today. Other approaches use vertical projections and local maxima or minima detection. Bretschneider et Al [7,8] suggest a wavelet decomposition using the low frequency image and the vertical components for a fast detection. Some recent work [9,10] improve Kokaram's approach, but most of the techniques are intraframe methods, neglecting the scratch tracking [11].

In our approach, we consider a large number of test sequences (old footage and new shoots). We state that a tracking mechanism considerably increases the detection quality because line scratches can be very unsteady. The x -position of the line scratch can move sideways up to 10 % of the image width (see figure 2). Consequently, the intra-frame shape of the scratch is not perfectly vertical and the corresponding slope might reach 5 degrees. All the methods based on full frame projection or vertical mean fail in this case.

A tracking improves the detection as well, essentially in noisy images. The localisation of the scratch detection is better and therefore its correction as well. At last, since a line scratch has a continuous life over many frames, our method allows an inter-frame tracking in order to assign a unique identifier to the detected scratch for its entire lifetime. This is important for our user interface (selection on a *per-scratch* basis instead of a *per-frame* basis).

The present work deals essentially with persistent line scratches (several consecutive frames). Other methods based on motion compensation and temporal discontinuity in image brightness are more suitable for short line scratches (ap-

¹ Proper film digitisation requires a wet gate process, which dramatically reduces visible scratches. In a wet gate, a liquid (perchloroethylene) fills the gaps, and most of fine scratches are no longer visible. But the wet gate process requires the use of chemicals and is not very compatible with a high digitalisation throughput

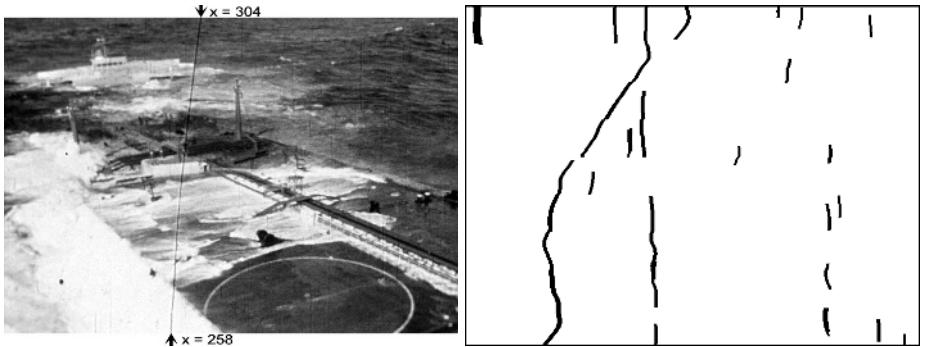


Fig. 2. Image exhibiting a notably slanted scratch, from the “marée noire, colère rouge” documentary and tracked path over 9 consecutive frames. Shown image is the 2nd one.

pearing randomly on a single frame only), as well as dust spots. Those methods fail with persistent scratches, present on the previous/next frame at nearly the same position and consequently matched and labelled as part of the scene.

2 Pre-processing : Image Projection

Though we cannot assume line scratches to be vertical over all an image, this hypothesis is locally true for a few consecutive horizontal lines in the original image I . We consider that scratch abscissa is locally constant over a band of H lines of I . Several advantages direct us to work with an image P , the vertically sub-sampled projection of the original image I . Each line of P is the vertical mean value of H lines of I (we call H the projection height) :

$$P(x, y) = \sum_{i=0}^{H-1} \frac{I(x, y \times H + i)}{H}. \quad (1)$$

- The amount of data (and processing time) is reduced by a factor of H .
- Noise is reduced by \sqrt{H} if gaussian.
- Line scratches intensity remains unaltered (assumed constant over H lines).

This simple method gives very good results, though more complex projection schemes may be used, for example overlapping bands or a weighted mean. Let us emphasize that the H parameter is of primary importance, because it will impact all the remaining processing steps and determine the maximum detectable scratch slope q . Above this maximum slope, scratches become attenuated after projection. H can be determined with respect to q by the following relation : $H = \frac{1}{\tan(q)}$; for $q=5$ degrees, we have $H=12$ pixels. According to image size, we use $H=8$, $H=12$ or $H=16$ (exact divisors). Figure 3 illustrates the projection transform.



Fig. 3. Image from the “lost world” movie and projected image $P(x, y)$ for 7 consecutive frames. The projection height H is 16. The scratches (dark one on the left side, bright one and dark one in the middle) are still visible.

3 Line Scratch Candidates Selection

The next step is the extraction of candidates which are used as input in the tracking process. The typical spatial signature for a scratch is a local extremum of the intensity curve along the x axis. So pixels candidate should be local maxima or minima horizontally, to find bright or dark scratches respectively. Many different methods exist in the literature to achieve this detection, and we experimented several ones [12]. For this work, we want our candidate extractor to meet the following requirements :

- Generate signed output : positive for bright scratches, negative for dark ones.
- Give a quality measure for each candidate, not only a simple binary output.
- Normalise quality measure between some known bounds, typically $\{-1, +1\}$.

The method we use relies on greyscale morphology [13,14]. Candidates for a line scratch are extracted by computing the difference between the original image and its opening or closing with a structuring element B_w . The opening will

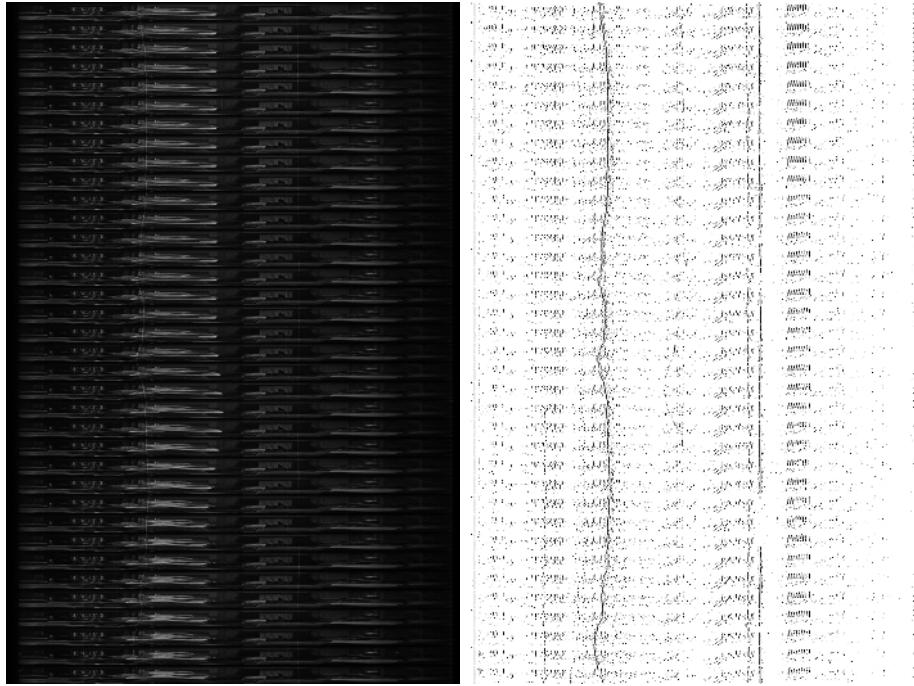


Fig. 4. The image $P(x,y)$ (left side) is computed from a 36 frames (1.5 second) sequence. The output $Q(x,y)$ (right side) is here shown as greyscale image ; real output values are signed so the tracker cannot confuse bright scratch candidates and dark ones.

remove thin structures brighter than the background, while closing will remove thin structures darker than the background. This way, to extract bright candidates, we subtract from P its opening with B_w , and symmetrically candidates for dark scratches are defined as the difference between the pixel values in P and their closing with B_w :

$$D^+(x,y) = P(x,y) - ((P(x,y) \ominus B_w) \oplus B_w) . \quad (2)$$

$$D^-(x,y) = ((P(x,y) \oplus B_w) \ominus B_w) - P(x,y) . \quad (3)$$

- $D^\pm(x,y)$ is the difference between the greyscale pixel value being considered, and a spatial neighbourhood of width w .
- \oplus stands for morphological dilatation, \ominus for morphological erosion and B_w is an unconstrained 1-D structuring element, of width w .

Because line scratches can be poorly contrasted relatively to their background, whereas natural image structures generally show a much stronger response, we locally normalise the result, to consider the significance of the extremum with respect to its spatial neighbourhood, using the following formula :

$$\begin{aligned}
 & \text{if } (((P(x, y) \oplus B_w) - (P(x, y) \ominus B_w)) > s) \\
 & \quad Q(x, y) = A \times \frac{D^+(x, y) - D^-(x, y)}{(P(x, y) \oplus B_w) - (P(x, y) \ominus B_w)} \\
 & \text{else } Q(x, y) = 0
 \end{aligned}$$

- $Q(x, y)$ stands for the output image of this detector. This image is signed ; positive values standing for local maxima and negative values for local minima. The tracking stage will use this image as input.
- $(P(x, y) \oplus B_w) - (P(x, y) \ominus B_w)$ is the local contrast.
- s is a threshold (see below).
- A is a scaling factor, which determines output values range $[-A \dots +A]$. We typically use $A = 127$, to store $Q(x, y)$ as an 8-bit greyscale image

The major tuning parameters are w and s . w defines the maximum scratch width, and the size of the neighbourhood used to normalise output. It strongly depends on the input image resolution. We obtained satisfactory results with $5 \leq w \leq 9$ at video resolution (720×576), and $9 \leq w \leq 13$ at high resolution (2048×1536). The threshold s has two goals : reduce the amount of false alarms, and inhibit candidate extraction in smooth areas. It controls the sensitivity and is usually set to some low value, but still required to eliminate spurious candidates. Figure 4 shows a result of a candidate detection.

4 Tracking Problem Formulation

4.1 General Background

After the pre-processing and detection stage, we still have to track the scratch candidates over the sequence. A human observer will easily locate the most visible scratches in figure 4, but the visual localisation of incomplete ones requires more concentration. An automated tracking system should be fooled by false alarms too, especially if vertical structures are present in the image.

The proposed tracking scheme should be able to distinguish real scratches from false alarms, to close the gaps caused by detection failures or discontinuous scratches, to find the optimum path through candidates and also uniquely identify the scratches (the detection process will assign an unique ID to each scratch, ranging from the frame where it appears to the frame where it vanishes).

The input of our tracking scheme is the image $Q(x, y)$. The lines are read sequentially, from the top to the bottom, so that the temporal axis matches the y axis of this image. Each line is a data set Z_t for the tracker (observation). In fact, such a representation (figure 4, right side) is very similar to a radar (or sonar) plot (echoes vs. time). Tracking schemes for such applications are common in the literature, and several approaches exist : Kalman filtering, AR methods, probabilistic data association filter (PDAF), multiple hypothesis trackers (MHT), Monte-Carlo and particle filters, ...; see an overview in [15]. Our problem is even simpler, since only one parameter should be estimated :

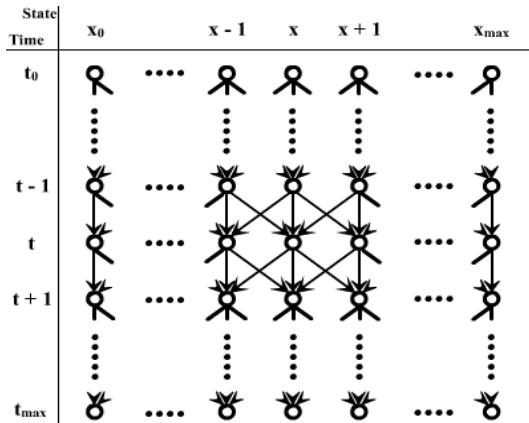


Fig. 5. Basis structure of trellis diagram built for each track. Since the representation space is matched against the state space, the possible transitions from state X_{t-1} to X_t are linked to a one-pixel deviation from the x position attached to state X_{t-1} .

the scratch localisation on the x-axis. Therefore, the state space and the world model (or representation space) are tightly matched.

Kalman filtering or PDAF approaches combine the different hypothesis at each step, while the MHT multiple hypothesis scheme keeps multiples hypothesis alive [16]. The idea is that by getting more observations Z_t , received at time t , and matching these to the hypotheses, the hypothesis corresponding to a real scratch path will gain more evidence, making it distinguishable from false ones. Besides, the hypothesis for a not perfectly continuous scratch will not disappear too quickly. Another advantage of such an approach is to unify in one concept path initialisation, path tracking and path decay.

Of course, the challenge is to maintain a reasonable number of hypothesis, according to available memory and computing power. Rejecting improbable hypothesis or keeping the best one at each stage are possible approaches. Since the whole sequence could be digitised prior to the processing, an exhaustive search of the optimum path for each scratch is also possible (although not reasonable); but our implementation heads to process data *on the fly* and therefore can be used in near-real time systems, gathering the images as outputted by a telecine.

4.2 Path Hypothesis Generation

The path hypothesis generation consists in building a trellis diagram in the state space. In this trellis diagram, we find possible states X_t at time t , and branches representing transitions from one state at time t to the next at time $t + 1$. The tracking process can follow simultaneously multiple scratches, but to keep the algorithm simple we will use a different trellis for each track, and consider simultaneous tracks as independent (accordingly, track merging is impossible).

A particular path hypothesis through the trellis is defined as a set of sequential states $X_{t_0}, X_{t_1}, \dots, X_{t-1}, X_t$, t_0 being the starting time (initialisation) for the track we consider. Sequentially, as shown on figure 5, each state at time t is linked with only 3 states at time $t - 1$, and conversely. This is due to the fact that we tolerate only, after projection, a horizontal displacement of one pixel between two consecutive lines for a track. As a consequence, the total number of possible paths for a particular track, at time t , is t^3 .

A new trellis for a new track (holding one state $X_{t_0} = x$ and one path hypothesis) is generated if, for a given data set Z_t (a line taken from the image $Q(x, y)$), an unmatched, isolated but relevant candidate is found.

4.3 Path Hypothesis Update

As stated earlier, the challenge is to prune the hypothesis tree, which grows up even if the new data set does not contain high detection values. Since the path hypothesis are represented as a trellis diagram in the state space, a practicable approach consist in weighting each transition from a state at stage $m - 1$ to a state at stage m . The well-known Viterbi algorithm can be used to sequentially prune the paths. All paths kept at the previous state are extended to the possible states at the next stage, and the best path leading to each state is selected. Other paths are eliminated after further consideration. The Viterbi algorithm can be used in its standard form when the transition costs between states depend only of the previous state and the current measurements. If this condition is not met (non-markovian) then the kept path might be sub-optimum.

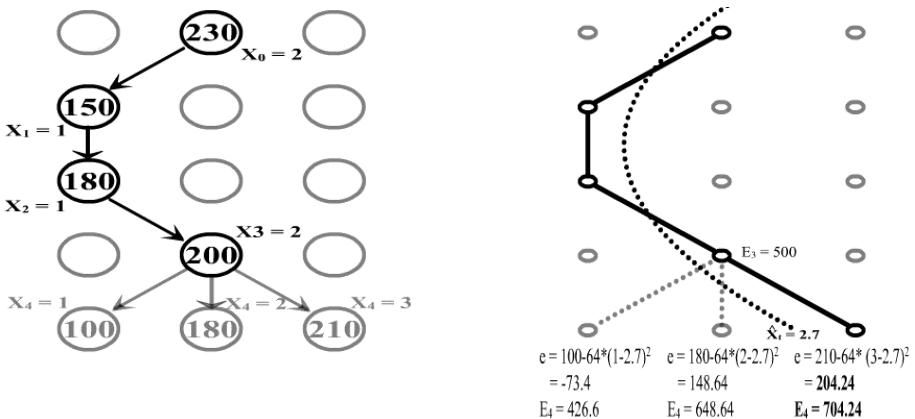


Fig. 6. These figures illustrate a possible path trough the state space, and therefore the representation space $Q(x, y)$ (see text). Each line of $Q(x, y)$ is used as observation Z_t for the tracker. To prune the paths and keep the L-best ones, the likelihood of a track is measured by a cost or energy function, based on both the quality of candidates (left figure) and closeness to an estimated path using short term history for parameter estimation (dotted line in the right figure). The cost computation is done for each path in the L-list, and for each possible new state.

To overcome this behaviour, we use the list-Viterbi algorithm (L-Viterbi) keeping a list of the L most valuable paths at each state and for each stage [17, 18] We will sequentially prune paths which are unlikely, and choose L so that no valid path is eliminated. The risk that an optimum path is rejected is alleviated as L increases. Like in the Viterbi algorithm, a value is computed along a path (cost function if the value should be minimised, else energy function). While the real Viterbi algorithm use plausibility value to score the transitions from state to state, we give below details on our implementation.

The set of possible paths at time t is noted C_t , with $C_{t,i}$ the i^{th} possible path at this time. At each branch in the trellis, we assign an cost or energy function $E_{t,i}$ which depends on the path $C_{t,i}$ being considered, and for each path $C_{t,i}$ we defined $E(C_{t,i})$ as the cumulative energy of its path branches :

$$E(C_{t,i}) = \sum_{n=0}^t E_{n,i} . \quad (4)$$

Our tracking problem can now be summarized as finding the L-optimal paths through this trellis, maximising the energy function E . For the path $C_{t,i}$ the energy assigned to the branch linking the state $X_{t-1,i}$ to $X_{t,i}$ is :

$$E_{t,i} = |Q(X_{t,i})| - W * (X_{t,i} - \hat{X}_{t,i})^2 . \quad (5)$$

The first term $Q(X_{t,i})$ is the quality criteria of the candidate associated with the state $X_{t,i}$. Using it as part of the cost function is quite obvious, since a line scratch is defined as set of sequential local extrema (candidates) extracted from $P(x,y)$. So a path should maximise the amount of candidates holding a strong quality criteria. The sign of $Q(x,y)$ is used as toggle to prevent mixing “bright” candidates and “dark” candidates, but $|Q(x,y)|$ is used in the energy function.

The second term $(X_{t,i} - \hat{X}_{t,i})^2$ is the squared difference between $X_{t,i}$ and $\hat{X}_{t,i}$, an estimate using the state history on path $C_{t,i}$. This is a tension constraint, a basic line scratch model, which will prevent paths which are not rigid enough to be chosen. The physical behaviour (inertia) of line scratches is reflected by this model. This constraint will prevent a path from locking on isolated extrema, especially when no more valid candidates seems available. The model used for the $\hat{X}(t,i)$ estimate is a 2-order polynomial, which is enough for our requirements :

$$\hat{x}(t) = \sum_{i=0}^2 (a_i t^i) . \quad (6)$$

We estimate the polynomial coefficients using a least square method, on N previous states : $X_{t-n,i}, X_{t-n+1,i}, \dots, X_{t-1,i}$. N must be high enough to prevent model divergence, and low enough to fit well the local trajectory. Kalman filter was taken in consideration for this task in earlier work [19].

Finally, W is a scaling factor, used to control the respective influence of both contributions (and so the rigidity of estimated tracks). It is strongly dependant of the parameter A used in the pre-processing stage. We obtained good results with $W = \frac{1}{4}A$, and choosing N according to the projected image height.

4.4 Track Ending Condition

As the update process handles the hypothesis tree pruning, the tracking mechanism is kept running until the track's ending condition is reached. If we introduce predominant history-related factors in the ending condition computation, long scratches will be *kept alive* while the ending condition is quickly reached for short ones. So, only the quality values of candidates are used. The mean value of the quality values $Q(x, y)$ associated to the states $X_t \dots X_{t-N}$ for the best path will be computed. The tracking is suspended if this value falls below a threshold ; we suppose the end of the scratch is reached. The path is stored in an intermediate data file for the subsequent removal processing, and the trellis representation is cleared from memory. This ending condition induces the tracking to overshoot the real scratch end. It could be improved by searching the strongest negative variation of $Q(x, t)$ along the best path if the ending condition is met.

4.5 General Algorithm

```

for each observation Z_t
    for each candidate in Z_t with non-zero quality value Q(x,y)
        initialise a new track
    end for
    for each track
        for each path hypothesis (from the L-list at t-1) related to this track
            estimate model parameter using N states along this path
            extend model for time t
            for each new state X_t reachable from state X_{t-1}
                compute cost for the transition X_{t-1} to X_t
                add new transition cost to overall path cost
            end for
        end for
        sort and keep L-best paths, clear other paths from memory
        if end condition for the best path is met
            store track and clear memory
        end if
    end for
end for

```

5 Evaluation and Conclusion

Theis contribution is focused on the detection side of the scratch removal process. The described detection and tracking scheme feeds the subsequent correction process with scratch trajectories ; this later process could rely on several approaches : interpolation, in-painting, Showing a result for the complete removal process on the basis of a single image is irrelevant in printed form, since the restoration quality could only be assessed by dynamic rendering². Generally, we do not have film samples before degradation, and working with synthetic data

² See video clips attached to the electronic version of this paper

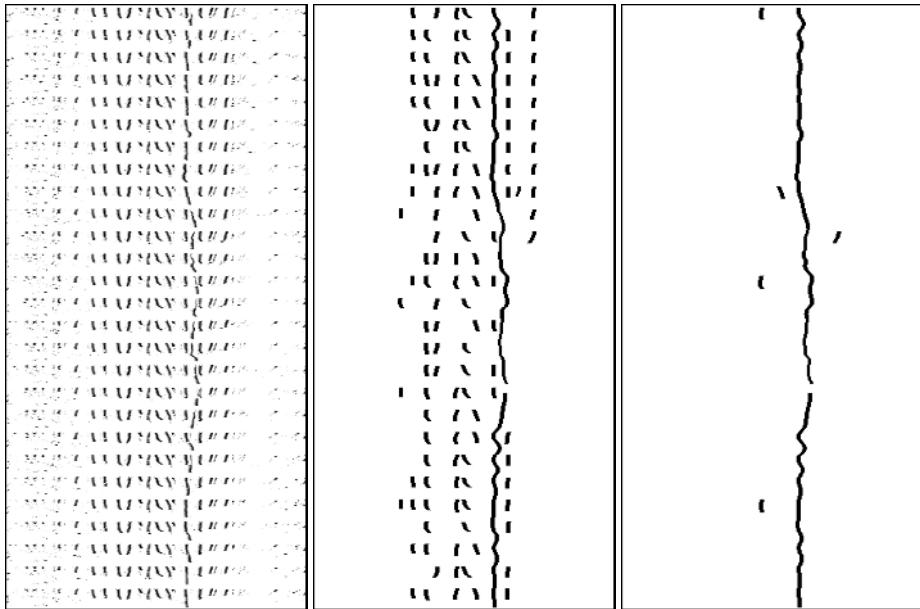


Fig. 7. Left : the image $Q(x, y)$ for 27 consecutive frames. This sequence shows many vertical structures (vertical curtain folds in the background of the scene) beside a real scratch. middle : Tracking results. Right : Tracking result keeping the longest paths

is nonsense (simulated scratches : what model to use), so a objective efficiency measurement is difficult. And because the goal of restoration is to improve the visual quality of degraded film sequences, the appropriate evaluation method is by subjective evaluation.

Even if we still find really weird images (for ex. with a lot of vertical structures, similar to the figure 7) overthrowing our algorithm, the overall efficiency of this detection scheme has been proved, and performs better than the previous ones or other known methods, for jittering scratches as well as steady ones. At present, we are improving the whole scratch removal process, especially the correction step by limiting the repetitive over-corrections..

At last, this tracking concept is used in our restoration software suite called RETOUCHE, used by the French post-production group **Centrimage** and by the **CNC** (French national film archives). RETOUCHE has been used for the digital restoration of 3 full-length features in 2K resolution and one video documentary (ca. 500000 frames) with convincing results. The algorithm and its implementation are also fast (less than a second per frame for 2K images).

Acknowledgements. Images from “The Lost World” (1925) by courtesy of Lobster Films, images from “Marée noire, colère rouge” (1978) by courtesy of the Cinémathèque de Bretagne.

References

1. Decencière, E., Serra, J.: Detection of local defects in old motion pictures. In: VII National Symposium on Pattern Recognition and Image Analysis, Barcelona, Spain (1997) 145–150
2. Joyeux, L., Boukir, S., Besserer, B., Buisson, O.: Reconstruction of degraded image sequences. application to film restoration. *Image and Vision Computing* **19** (2001) 503
3. Takahiro, S., Takashi, K., Toshiaki, O., Takamasa, S.: Image processing for restoration of heavily-corrupted old film sequences. In Society, I.C., ed.: 15th. International Conference on Pattern Recognition (ICPR'00). Volume 3., Barcelona, Spain (2000) 17–20
4. Kokaram, A., Morris, R., Fitzgerald, W., Rayner, P.: Detection of missing data in image sequences. *i3etip* **4** (1995) 1496–1508
5. Kokaram, A.: Detection and removal of line scratches in degraded motion picture sequences. In: Proceedings of EUSIPCO'96, Trieste, Italy (1996)
6. Kokaram, A.: Motion picture restoration. Springer-Verlag (1998)
7. Bretschneider, T., Kao, O.: Detection and removal of scratches in digitised film sequences. In: International Conference on Imaging Science, Systems, and Technology. (2001) 393–399
8. Bretschneider, T., Miller, C., Kao, O.: Interpolation of scratches in motion picture films. In: IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP). Volume 3. (2001) 1873 –1876
9. Vitulano, D., Bruni, V., Ciarlini, P.: Line scratch detection on digital images: An energy based model. In: International Conference in Central Europe on Computer Graphics and Visualization (WSCG). Volume 10. (2002) 477
10. Maddalena, L.: Efficient methods for scratch removal in image sequences. In: 11th International Conference on Image Analysis and Processing (ICIAP2001), IEEE Computer Society (2001) 547–552
11. Tegolo, D., Isgro, F.: Scratch detection and removal from static images using simple statistics and genetic algorithms. In: International Conference on Image Analysis and Processing. (2001) 507–511
12. Joyeux, L., Buisson, O., Besserer, B., Boukir, S.: Detection and removal of line scratches in motion picture films. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA (1999) 548–553
13. Serra, J.: *Image Analysis and Mathematical Morphology*. Volume 1. Academic Press, London, England (1982)
14. Serra, J.: *Image Analysis and Mathematical Morphology: Theoretical Advances*. Volume 2. Academic Press, London, England (1988)
15. Cox, I.J.: A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision* **10** (1993) 53–66
16. Cox, I.J., Hingorani, S.L.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In: IEEE Trans. on PAMI. Volume 18. (1996) 138–150
17. Perry, R., Vaddiraju, A., Buckley, K.: Trellis structure approach to multitarget tracking. In: Proc. Sixth Annual Workshop on ASAP. (1999)
18. Bradley, J., Buckley, K., Perry, R.: Time-recursive number-of-tracks estimation for mht. In: Signal and Data Processing of Small Targets, Orlando, FL (2000)
19. Joyeux, L., Boukir, S., Besserer, B.: Tracking and map reconstruction of line scratches in degraded motion pictures. *Machine Vision and Applications* **Volume 13, Number 3** (2002) 119–128

Color Constancy Using Local Color Shifts

Marc Ebner

Universität Würzburg, Lehrstuhl für Informatik II,
Am Hubland, 97074 Würzburg, Germany,
`ebner@informatik.uni-wuerzburg.de`,
<http://www2.informatik.uni-wuerzburg.de/staff/ebner/welcome.html>

Abstract. The human visual system is able to correctly determine the color of objects in view irrespective of the illuminant. This ability to compute color constant descriptors is known as color constancy. We have developed a parallel algorithm for color constancy. This algorithm is based on the computation of local space average color using a grid of processing elements. We have one processing element per image pixel. Each processing element has access to the data stored in neighboring elements. Local space average color is used to shift the color of the input pixel in the direction of the gray vector. The computations are executed inside the unit color cube. The color of the input pixel as well as local space average color is simply a vector inside this Euclidean space. We compute the component of local space average color which is orthogonal to the gray vector. This component is subtracted from the color of the input pixel to compute a color corrected image. Before performing the color correction step we can also normalize both colors. In this case, the resulting color is rescaled to the original intensity of the input color such that the image brightness remains unchanged.

1 Motivation

The human visual system is able to correctly determine the color of objects irrespective of the light which illuminates the objects. For instance, if we are in a room illuminated with yellow lights, we are nevertheless able to determine the correct color of the objects inside the room. If the room has a white wall, it will reflect more red and green light compared to the light reflected in the blue spectrum. Still, we are able to determine that the color of the wall is white. However, if we take a photograph of the wall, it will look yellow. This occurs because a camera measures the light reflected by the object. The light reflected by the object can be approximated as being proportional to the amount of light illuminating the object and the reflectance of the object for any given wavelength. The reflectance of the object specifies the percentage of the incident light which is reflected by the object's surface. The human visual system is somehow able to discount the illuminant and to calculate color constant descriptors if the scene is sufficiently complex. This ability is called color constancy. In other words, the human visual system is able to estimate the actual reflectance, i.e. the color of

the object. The perceived color stays constant irrespective of the illuminant used to illuminate the scene.

Land, a pioneer in color constancy research, has developed the retinex theory [1,2]. Others have added to this research and have proposed variants of the retinex theory [3,4,5,6,7,8]. Algorithms for color constancy include gamut-constraint methods [9,10,11], perspective color constancy [12], color by correlation [13,14], the gray world assumption [15,16], recovery of basis function coefficients [17,18,19], mechanisms of light adaptation coupled with eye movements [20], neural networks [21,22,23,24,25,26], minimization of an energy function [27], comprehensive color normalization [28], committee-based methods which combine the output of several different color constancy algorithms [29] or use of genetic programming [30]. Risson [31] describes a method to determine the illuminant by image segmentation and filtering of regions which do not agree with an assumed color model.

We have developed a parallel algorithm for color constancy [32,33,34]. The algorithm computes local space average color using a parallel grid of processing elements. Note that local space average color is not the same as global space average color. Global space average color assumes a single illuminant. In contrast, we do not assume a uniform illumination of the scene. Local space average color is taken as an estimate of the illuminant for each image pixel. This estimate of the illuminant is then used to perform a local color correction step for each image pixel.

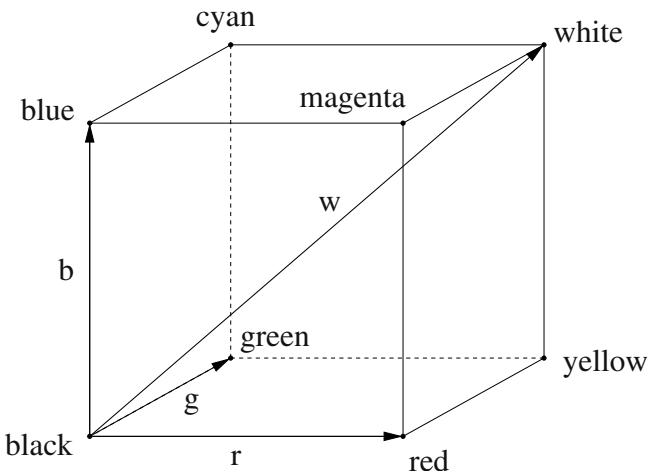


Fig. 1. RGB color space. The space is defined by the three vectors \mathbf{r} (red), \mathbf{g} (green), and \mathbf{b} (blue). The gray vector \mathbf{w} passes through the center of the cube from black to white.

2 RGB Color Space

Let us visualize the space of possible colors as a color cube [35,36] and let $\mathbf{r} = [1, 0, 0]^T$, $\mathbf{g} = [0, 1, 0]^T$, $\mathbf{b} = [0, 0, 1]^T$ be the three color vectors red, green and blue, which define the cube. The color components are normalized to the range $[0, 1]$. Therefore, all colors are located inside the unit cube. The eight corners of the cube can be labeled with the colors black, red, green, blue, magenta, cyan, yellow, and white. The gray vector passes through the cube from $[0, 0, 0]^T$ to $[1, 1, 1]^T$. This RGB color space is shown in Figure 1.

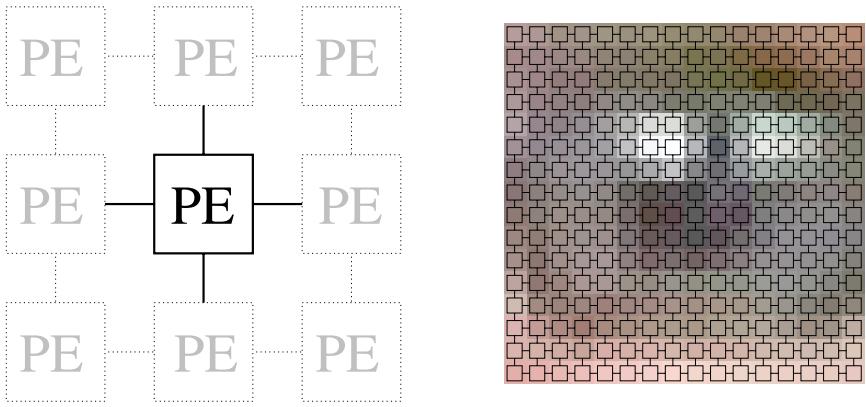


Fig. 2. Each processing element has access to information stored at neighboring processing elements (left). A matrix of processing elements with one processing element per pixel is used (right).

3 Parallel Computation of Local Space Average Color

The algorithm operates on a grid of processing elements. Each processing element has access to the color of a single image pixel. It also has access to data stored and computed by four neighboring processing elements (Figure 2). We have one processing element per image pixel. The algorithm first determines local space average color. Local space average color is calculated iteratively by averaging estimates of the local space average color from neighboring elements. Let $a_i(x, y)$ with $i \in \{r, g, b\}$ be the current estimate of the local space average color of channel i at position (x, y) in the image. Let $c_i(x, y)$ be the intensity of channel i at position (x, y) in the image. Let p be a small percentage greater than zero. We iterate the following two steps indefinitely.

- 1.) $a'_i(x, y) = (a_i(x - 1, y) + a_i(x + 1, y) + a_i(x, y - 1) + a_i(x, y + 1))/4.0$
- 2.) $a_i(x, y) = c_i(x, y) \cdot p + a'_i(x, y) \cdot (1 - p)$

The first step averages the estimate obtained from the neighboring elements on the left and right, as well as above and below. The second step slowly fades the color of the current pixel $c_i(x, y)$ into the current average. As a result, we obtain local space average color for each pixel of the image. Note that initialization of a_i can be arbitrary as it decays over time due to the multiplication by the factor $(1 - p)$. Figure 3 shows how local space average color is computed with this method for an input image. Since the intensities are averaged for each color channel, it is important that the input data is linear. If necessary, it must be linearized by applying a gamma correction.



Fig. 3. Local space average color is computed iteratively. The images show local space average color after 1, 50, 200, 1000, 5000, and 20000 steps. For this image 22893 steps were needed until convergence.

As the estimate of local space average color is handed from one element to the next, it is multiplied with the factor $(1 - p)$. Therefore, a pixel located n steps from the current pixel will only contribute with a factor of $(1 - p)^n$ to the new estimate of local space average color. The above computation is equivalent to the convolution of the input image with the function $e^{-\frac{|r|}{\sigma}}$ where $r = \sqrt{x^2 + y^2}$ is the distance from the current pixel and σ is a scaling factor. In this case, local space average color \mathbf{a} is given by

$$\mathbf{a} = k \int_{x,y} \mathbf{c} e^{-\frac{|r|}{\sigma}} dx dy \quad (1)$$

where k is chosen such that

$$k \int_{x,y} e^{-\frac{|r|}{\sigma}} dx dy = 1. \quad (2)$$

Thus, the factor p essentially determines the radius over which local space average color is computed. Figure 4 shows the result for different values of p . In practice, this type of computation can be performed using a resistive grid [7,25].



Fig. 4. The parameter p determines the extent over which local space average color will be computed. If p is large, then local space average color will be computed for a small area. If p is small, then local space average color will be computed for a large area.

Alternatively, instead of using a convolution with $e^{-\frac{|r|}{\sigma}}$ to compute local space average color, one can also compute space average color by convolving the input image with a Gaussian. In this case, local space average color is given by

$$\mathbf{a} = k \int_{x,y} \mathbf{c} e^{-\frac{r^2}{\sigma^2}} dx dy \quad (3)$$

where k is chosen such that

$$k \int_{x,y} e^{-\frac{r^2}{\sigma^2}} dx dy = 1. \quad (4)$$

This type of convolution is used by Rahman et al. [8] to perform color correction using several Gaussians of varying extent for each image pixel.

4 Color Constancy Using Color Shifts

According to the gray world hypothesis, on average, the color of the world is gray [15,16]. If space average color deviates from gray, it has to be corrected. Instead of rescaling the color channels, we use local space average color to shift the color vector in the direction of the gray vector. The distance between space average color and the gray vector determines how much local space average color deviates from the assumption that, on average, the world is gray. Let $\mathbf{w} = \frac{1}{\sqrt{3}}[1, 1, 1]^T$ be the normalized gray vector. Let $\mathbf{c} = [c_r, c_g, c_b]^T$ be the color of the current pixel and $\mathbf{a} = [a_r, a_g, a_b]^T$ be local space average color. We first project local space average color onto the gray vector. The result is subtracted from local space average color. This gives us a vector which is orthogonal to the gray vector. It

points from the gray vector to the local space average color. Its length is equal to the distance between local space average color and the gray vector.

$$\mathbf{a}_{\perp} = \mathbf{a} - (\mathbf{a} \cdot \mathbf{w})\mathbf{w} \quad (5)$$

This vector is subtracted from the color of the current pixel in order to undo the color change due to the illuminant. The output color is calculated as

$$\mathbf{o} = \mathbf{c} - \mathbf{a}_{\perp} \quad (6)$$

or

$$o_i = c_i - a_i + \frac{1}{3}(a_r + a_g + a_b) \quad (7)$$

where $i \in \{r, g, b\}$. If we define $\bar{a} = \frac{1}{3}(a_r + a_g + a_b)$, we have

$$o_i = c_i - a_i + \bar{a}. \quad (8)$$

This operation is visualized in Figure 5 for two vectors \mathbf{c} and \mathbf{a} .

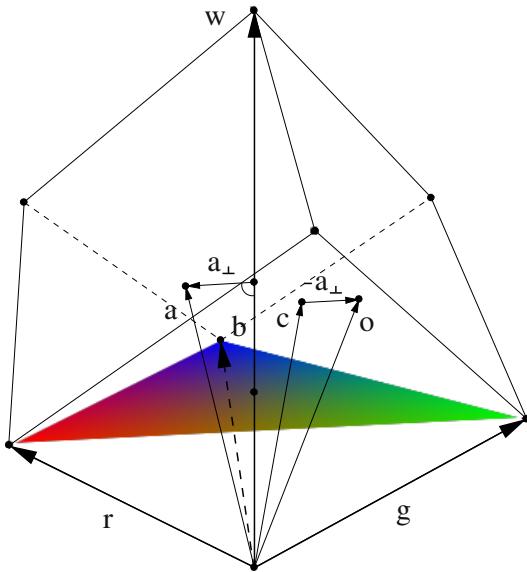


Fig. 5. First, the vector \mathbf{a} is projected onto the white vector \mathbf{w} . The projection is subtracted from \mathbf{a} which gives us \mathbf{a}_{\perp} , the component perpendicular to \mathbf{w} . This vector is subtracted from \mathbf{c} to obtain a color corrected image.

The entire algorithm can be realized easily in hardware. The averaging operation can be realized using a resistive grid [7,25]. See Koosh [37] for a realization

of analog computations in VLSI. We only require local connections. Therefore, the algorithm is scalable to arbitrary image sizes.

Instead of subtracting the component of space average color which is orthogonal to the white vector, one can also normalize both colors first

$$\hat{\mathbf{a}} = \frac{1}{a_r + a_g + a_b} [a_r, a_g, a_b]^T \quad (9)$$

$$\hat{\mathbf{c}} = \frac{1}{c_r + c_g + c_b} [c_r, c_g, c_b]^T \quad (10)$$

In this case, both space average color \mathbf{a} and the color of the current pixel \mathbf{c} are projected onto to the HSI plane $r + g + b = 1$ [35,38]. We again calculate the component of $\hat{\mathbf{a}}$ which is orthogonal to the vector \mathbf{w} .

$$\hat{\mathbf{a}}_{\perp} = \hat{\mathbf{a}} - (\hat{\mathbf{a}} \cdot \mathbf{w})\mathbf{w} \quad (11)$$

This component is subtracted from the normalized color vector $\hat{\mathbf{c}}$

$$\hat{\mathbf{o}} = \hat{\mathbf{c}} - \hat{\mathbf{a}}_{\perp} \quad (12)$$

or

$$\hat{o}_i = \hat{c}_i - \hat{a}_i + \frac{1}{3}. \quad (13)$$

The normalized output is then scaled using the illuminance component of the original pixel color.

$$o_i = (c_r + c_g + c_b)\hat{o}_i \quad (14)$$

$$= c_i - (c_r + c_g + c_b)(\hat{a}_i - \frac{1}{3}) \quad (15)$$

$$= c_i - \frac{c_r + c_g + c_b}{a_r + a_g + a_b}(a_i - \frac{1}{3}(a_r + a_g + a_b)) \quad (16)$$

$$= c_i - \frac{\bar{c}}{\bar{a}}(a_i - \bar{a}) \quad (17)$$

The calculations needed for this algorithm are shown in Figure 6.

5 Results

Both algorithms were tested on a series of images where different lighting conditions were used. Results for both algorithms are shown for an office scene in Figure 7. The images in the top row show the input image. The images in the second row show local space average color which was computed for each input image. The images in the third row show the output images of the first algorithm. In this case, the component of local space average color which is orthogonal to the gray vector was subtracted from the color of the input pixel. The images in the last row show the output images of the second algorithm. In this case,

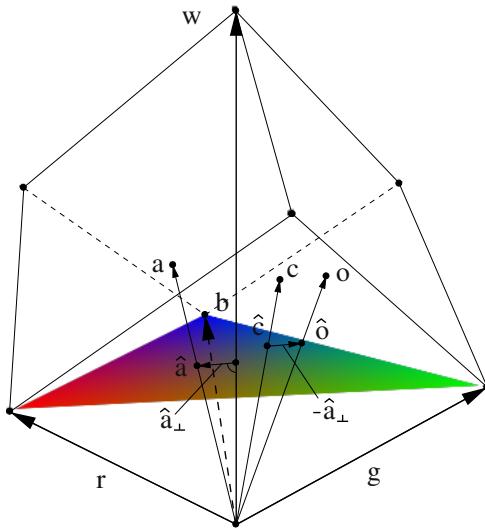


Fig. 6. First, the vectors c and a are projected onto the plane $r + g + b = 1$. The corresponding points are \hat{c} and \hat{a} respectively. Next, the vector \hat{a} is projected onto the white vector w . The projection is subtracted from \hat{a} which gives us \hat{a}_\perp , the component perpendicular to w . This vector is subtracted from \hat{c} and finally scaled back to the original intensity to obtain a color corrected image.

local space average color and the color of the input pixel was normalized before performing the color correction step.

The input images were taken with a standard SLR camera. A CD-ROM with the images was produced when the film was developed. All three images show a desk with some utensils. The first image is very blue due to blue curtains which were closed at the time the image was taken. Sun was shining through the curtains which produced the blue background illumination. For the second input image a yellow light bulb was used to illuminate the room. Finally, the desk lamp was switched on for the third image. Again, the blue background illumination is caused by sunlight shining through the blue curtains. Note that the output images are much closer to what a human observed would expect.

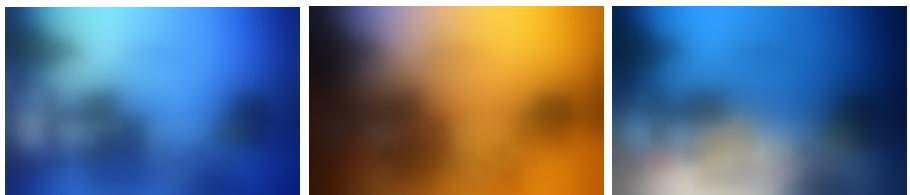
6 Discussion

In the following we discuss differences to other color constancy algorithms which are related to the algorithm described in this contribution. The algorithms of Horn [6,7], Land [2], Moore et al. [25] and Rahman et al. [8] do not accurately reproduce human color perception. Helson [39] performed an extensive study with human subjects. The subjects task was to name the perceived color of gray stimuli which were illuminated with colored light. The stimuli were placed on a

Input Images:



Local Space Average Color:



Results for Algorithm 1:



Results for Algorithm 2:



Fig. 7. Experimental results for an office scene. The first row of images show the input images. The second row of images show local space average color. The third row shows the output images which were computed by subtracting the component of local space average color which is orthogonal to the gray vector from the color of the input pixel. The last row shows the results when local space average color and the color of the input pixel were normalized before performing the color correction step.

gray background. Helsons drew the following conclusions from his experiments. If the stimuli has a higher reflectance than the background, then the stimuli seems to have the color of the illuminant. If the stimuli has the same reflectance as the background then the stimuli is achromatic. If the stimuli has a lower

reflectance than the background then the subject perceives the stimuli as having the complementary color of the illuminant. The algorithms of Horn [6,7], Land [2], Moore et al. [25], and Rahman et al. [8] do not show this behavior. If the fraction between the color of the pixel and local space average color is computed then the color of the illuminant falls out of the equation and the stimuli will always appear to be achromatic.

All of the above methods require a normalization step which brings the output to the range [0, 1]. This normalization step can either be performed independently for each color band or the normalization can be performed uniformly across all color bands. In any case, one needs to loop over all pixels of the image. The algorithm which is described in this contribution does not require such a normalization step. All of the computations are performed inside the color cube. Values outside this color cube are clipped to the border of the cube.

In our previous work [32,34] we already discussed in depth the computation of local space average color using a grid of processing elements. Previously, we divided the color of the input pixel by the local space average color. This is exactly the gray world assumption [15,16] applied locally. This algorithm does not show the behavior described by Helson [39]. In [33] we subtract local space average color from the color of the input pixel followed by a rescaling operation. This method also does not correspond to human color perception described by Helson.

The algorithms described in this contribution performs color correction inside the unit color cube. The color cube is viewed as an Euclidian space. The color of the pixel is shifted in a direction perpendicular to the gray vector. The extent of the color shift is computed using local space average color. The first of the two algorithms which are described in this contribution shows the same response as a human observer for similar stimuli as was used in Helson's experiments.

7 Conclusion

In light of the current transition away from analog cameras towards digital cameras it is now possible to post-process the digital images before development to achieve accurate reproduction of the scene viewed. Such post-processing can either be done by the CPU of the camera or by post-processing the images on external hardware before the images are printed. Accurate color reproduction is very important for automatic object recognition. However, one of the largest markets will probably be consumer photography.

We have developed an algorithm for color constancy. The method consists of two parts: (a) a parallel grid of processing elements which is used to compute local space average color and (b) a method to estimate the original colors of the viewed objects. Instead of rescaling the red, green, and blue intensities using the inverse of local space average color, we shift the color vector into the direction of the gray vector. The color shift is based on local space average color. Therefore, the algorithm can also be used in the presence of varying illumination.

References

1. Land, E.H.: The retinex theory of colour vision. Proc. Royal Inst. Great Britain **47** (1974) 23–58
2. Land, E.H.: An alternative technique for the computation of the designator in the retinex theory of color vision. Proc. Natl. Acad. Sci. USA **83** (1986) 3078–3080
3. Brainard, D.H., Wandell, B.A.: Analysis of the retinex theory of color vision. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: Color, Boston, Jones and Bartlett Publishers (1992) 208–218
4. Brill, M., West, G.: Contributions to the theory of invariance of color under the condition of varying illumination. Journal of Math. Biology **11** (1981) 337–350
5. Funt, B.V., Drew, M.S.: Color constancy computation in near-mondrian scenes using a finite dimensional linear model. In Jain, R., Davis, L., eds.: Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition, Ann Arbor, MI, Computer Society Press (1988) 544–549
6. Horn, B.K.P.: Determining lightness from an image. Computer Graphics and Image Processing **3** (1974) 277–299
7. Horn, B.K.P.: Robot Vision. The MIT Press, Cambridge, Massachusetts (1986)
8. Rahman, Z., Jobson, D.J., Woodell, G.A.: Method of improving a digital image. United States Patent No. 5,991,456 (1999)
9. Barnard, K., Finlayson, G., Funt, B.: Color constancy for scenes with varying illumination. Computer Vision and Image Understanding **65** (1997) 311–321
10. Forsyth, D.A.: A novel approach to colour constancy. In: Second International Conference on Computer Vision (Tampa, FL, Dec. 5–8), IEEE Press (1988) 9–18
11. Forsyth, D.A.: A novel algorithm for color constancy. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: Color, Boston, Jones and Bartlett Publishers (1992) 241–271
12. Finlayson, G.D.: Color in perspective. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996) 1034–1038
13. Barnard, K., Martin, L., Funt, B.: Colour by correlation in a three dimensional colour space. In Vernon, D., ed.: Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland, Berlin, Springer-Verlag (2000) 375–389
14. Finlayson, G.D., Hubel, P.M., Hordley, S.: Color by correlation. In: Proceedings of IS&T/SID. The Fifth Color Imaging Conference: Color Science, Systems, and Applications, Nov 17–20, The Radisson Resort, Scottsdale, AZ. (1997) 6–11
15. Buchsbaum, G.: A spatial processor model for object colour perception. Journal of the Franklin Institute **310** (1980) 337–350
16. Gershon, R., Jepson, A.D., Tsotsos, J.K.: From [R,G,B] to surface reflectance: Computing color constant descriptors in images. In McDermott, J.P., ed.: Proc. of the 10th Int. Joint Conf. on Artificial Intelligence, Milan, Italy. Volume 2., Morgan Kaufmann (1987) 755–758
17. Funt, B.V., Drew, M.S., Ho, J.: Color constancy from mutual reflection. International Journal of Computer Vision **6** (1991) 5–24
18. Ho, J., Funt, B.V., Drew, M.S.: Separating a color signal into illumination and surface reflectance components: Theory and applications. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: Color, Boston, Jones and Bartlett Publishers (1992) 272–283
19. Maloney, L.T., Wandell, B.A.: Color constancy: a method for recovering surface spectral reflectance. Journal of the Optical Society of America A **3** (1986) 29–33
20. D'Zmura, M., Lennie, P.: Mechanisms of color constancy. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: Color, Boston, Jones and Bartlett Publishers (1992) 224–234
21. Courtney, S.M., Finkel, L.H., Buchsbaum, G.: A multistage neural network for color constancy and color induction. IEEE Trans. on Neural Networks **6** (1995) 972–985

22. Dufort, P.A., Lumsden, C.J.: Color categorization and color constancy in a neural network model of v4. *Biological Cybernetics* **65** (1991) 293–303
23. Funt, B., Cardei, V., Barnard, K.: Learning color constancy. In: Proceedings of the IS&T/SID Fourth Color Imaging Conference, Scottsdale (1996) 58–60
24. Herault, J.: A model of colour processing in the retina of vertebrates: From photoreceptors to colour opposition and colour constancy phenomena. *Neurocomputing* **12** (1996) 113–129
25. Moore, A., Allman, J., Goodman, R.M.: A real-time neural system for color constancy. *IEEE Transactions on Neural Networks* **2** (1991) 237–247
26. Novak, C.L., Shafer, S.A.: Supervised color constancy for machine vision. In Healey, G.E., Shafer, S.A., Wolff, L.B., eds.: *Color*, Boston, Jones and Bartlett Publishers (1992) 284–299
27. Usui, S., Nakauchi, S.: A neurocomputational model for colour constancy. In Dickinson, C., Murray, I., Carden, D., eds.: *John Dalton's Colour Vision Legacy. Selected Proc. of the Int. Conf.*, London, Taylor & Francis (1997) 475–482
28. Finlayson, G.D., Schiele, B., Crowley, J.L.: Comprehensive colour image normalization. In Burkhardt, H., Neumann, B., eds.: *Fifth European Conf. on Computer Vision (ECCV '98)*, Freiburg, Germany, Berlin, Springer-Verlag (1998) 475–490
29. Cardei, V.C., Funt, B.: Committee-based color constancy. In: Proceedings of the IS&T/SID Seventh Color Imaging Conference: Color Science, Systems and Applications, Scottsdale, Arizona. (1999) 311–313
30. Ebner, M.: Evolving color constancy for an artificial retina. In Miller, J., Tomassini, M., Lanzi, P.L., Ryan, C., Tettamanzi, A.G.B., Langdon, W.B., eds.: *Genetic Programming: Proceedings of the 4th European Conference*, Lake Como, Italy, Berlin, Springer-Verlag (2001) 11–22
31. Risson, V.J.: Determination of an illuminant of digital color image by segmentation and filtering. United States Patent Application, Pub. No. US 2003/0095704 A1 (2003)
32. Ebner, M.: A parallel algorithm for color constancy. Technical Report 296, Universität Würzburg, Lehrstuhl für Informatik II, Am Hubland, 97074 Würzburg, Germany (2002)
33. Ebner, M.: Combining white-patch retinex and the gray world assumption to achieve color constancy for multiple illuminants. In Michaelis, B., Krell, G., eds.: *Pattern Recognition. Proceedings of the 25th DAGM Symposium*, Magdeburg, Germany, Berlin, Springer-Verlag (2003) 60–67
34. Ebner, M.: A parallel algorithm for color constancy. *Journal of Parallel and Distributed Computing* **64** (2004) 79–88
35. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Addison-Wesley Publishing Company, Reading, Massachusetts (1992)
36. Hanbury, A., Serra, J.: Colour image analysis in 3d-polar coordinates. In Michaelis, B., Krell, G., eds.: *Pattern Recognition. Proc. of the 25th DAGM Symposium*, Magdeburg, Germany, September 10–12, Berlin, Springer-Verlag (2003) 124–131
37. Koosh, V.F.: *Analog Computation and Learning in VLSI*. PhD thesis, California Institute of Technology Pasadena, California (2001)
38. Jain, R., Kasturi, R., Schunck, B.G.: *Machine Vision*. McGraw-Hill, Inc., New York (1995)
39. Helson, H.: Fundamental problems in color vision. i. the principle governing changes in hue, saturation, and lightness of non-selective samples in chromatic illumination. *Journal of Experimental Psychology* **23** (1938) 439–476

Image Anisotropic Diffusion Based on Gradient Vector Flow Fields

Hongchuan Yu and Chin-Seng Chua

School of Electrical and Electronic Engineering,
Nanyang Technological University,
639798 Singapore
`{ehcyy, ecschua}@ntu.edu.sg`

Abstract. In this paper, the gradient vector flow fields are introduced in the image anisotropic diffusion, and the shock filter, mean curvature flow and Perona-Malik equation are reformulated respectively in the context of this flow fields. Many advantages over the original models can be obtained, such as numerical stability, a large capture range, and computational simplification etc. In addition, the fairing process is introduced in the anisotropic diffusion, which contains the fourth order derivative and is reformulated as the intrinsic Laplacian of curvature under the level set framework. By this fairing process, the boundaries of shape will become more outstanding. In order to overcome numerical errors, the intrinsic Laplacian of curvature is computed from the gradient vector flow fields, but not directly from the observed images.

1 Introduction

The image anisotropic diffusion is to smooth the image in the direction of an edge, but not perpendicular to it, so that the location and strength of edges can be maintained. Since Perona-Malik equation was presented as an anisotropic diffusion model in [1], there have been extensive literatures that presented a lot of the anisotropic diffusion models and offered the numerical schemes to obtain steady state solution [2-10]. In this paper, we would pay attention to the following three classic anisotropic diffusion models, shock filter, mean curvature flow scheme, and Perona-Malik equation.

The shock filter scheme was presented in [12] as a stable deblurring algorithm approximating deconvolution. It is well known that it is extremely sensitive to noise. Many further researches almost focus on how to define more precise and robust coefficient function so as to smooth noise while preserve shape and geometric features. The frequent idea is to add some kind of anisotropic diffusion term with a weight between the shock and the diffusion processes. In [2], a combination form to couple shock with a diffusion term was proposed, $I_t = -\text{sign}(G_\sigma * I_{\eta\eta})|\nabla I| + cI_{\xi\xi}$, where c is a positive scale, η is the direction of gradient and ξ is the direction perpendicular to the gradient. In [4], a complex diffusion model was presented, $I_t = -(2/\pi)\arctan(a \text{Im}(I/\theta))|\nabla I| + \alpha_1 I_{\eta\eta} + \alpha_2 I_{\xi\xi}$, where the first term is shock term, a is a parameter that controls the sharpness, $\alpha_1 = re^{i\theta}$ is a complex scale, α_2 is a real scale.

The mean curvature flow model was presented in [16,8] as an edge enhancement algorithm in the presence of noise. In [10], the mean curvature flow was applied to enhancement and denoising under the Min/Max flow scheme. In the above applications, only the pure curvature flow was used. While a deconvolution model was introduced in the mean curvature flow model in [7] for deblurring and denosing.

For the Perona-Malik equation, a frequent idea is to design the coefficient functions directly so as to weight the terms $I_{\eta\eta}$ and $I_{\xi\xi}$ adaptively. In [5], the coefficient function was defined as, $g(x) = \left(1 + (x/k_1)^n\right)^{-1} - \alpha\left(1 + ((x - k_2)/w)^{2m}\right)^{-1}$, where k_1 is a threshold that is the limit of gradients to be smoothed out, k_2 and w are threshold and range that control the inverse diffusion process. In [6], the general form of Perona-Malik equation was presented, $I_t = c(aI_{\eta\eta} + bI_{\xi\xi})$, where the parameters a, b, c are defined respectively. And the two eigenvalues of its Hessian Matrix are instead of the second order derivatives $I_{\eta\eta}$ and $I_{\xi\xi}$ in order to suppress the influence of noise. From a numerical view, these mentioned algorithms are too complicated, and over many parameters need to be determined.

In this paper, the Gradient Vector Flow fields [11] are introduced in the anisotropic diffusion. Since this flow fields can be determined in advance, (i.e. they are invariable during image diffusion), and besides, provide a large capture range to the object boundaries, they perform well on noise or spurious edges. Another particular advantage is to simplify computation. We will demonstrate these advantages by applying the gradient vector flow fields to the shock filter, the mean curvature flow and the Perona-Malik equation respectively. Since many proposed anisotropic diffusion models usually regarded these three models as their basic prototypes. Besides that, in order to make the enhanced boundaries vivid, the fourth order flow model of the plane curve [13] is introduced in the anisotropic diffusion. Because of the gradient vector flow fields, the computation for the fourth order flow will become simple and reliable.

This paper is organized as follows: in section 2, the gradient vector flow fields are briefly introduced. In section 3, the shock filter, mean curvature flow, Perona-Malik model and fourth order flow model based on the gradient vector flow fields are presented respectively. The implementation details of our presented models and experimental results are shown in section 4. Finally, our conclusions and ideas for future research appear in section 5.

2 Gradient Vector Flow (GVF) Fields

The gradient vector flow (GVF) field was firstly presented for the active contour models, in which the GVF field was used as an external force in [11]. Because the GVF fields are computed as a diffusion of the intensity gradient vectors, i.e. the GVF is estimated directly from the continuous gradient vector space, and its measurement is contextual and not equivalent with the distance from the closest point. Thus, the noise can be suppressed. Besides that, the GVF provides a bi-directional force field that can capture the object boundaries from either side without any prior knowledge about whether to shrink or expand toward the object boundaries. Hence, the GVF fields can provide a large capture range to the object boundaries.

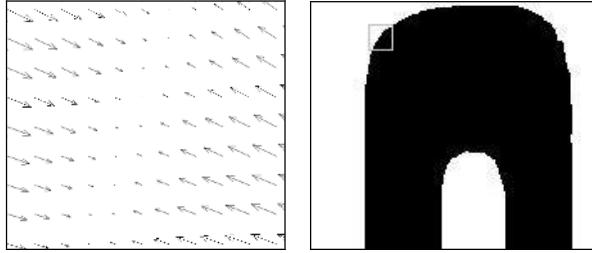


Fig. 1. GVF field corresponding to the rectangle on right image

First, a Gaussian edge detector (zero mean, with σ_E variance) is used in the edge map defining [14], $f(\mathbf{x}) = 1 - \frac{1}{\sqrt{2\pi}\sigma_E} \exp\left(-\frac{|\nabla(G_\sigma * I)(\mathbf{x})|^2}{2\sigma_E^2}\right)$, $\mathbf{x} \in R^2$. In fact, if only

the boundary information (be usually the intensity gradient) is taken into account, the edge map $f(\mathbf{x})$ can be defined as other forms too [11]. The GVF field $\vec{\mathbf{v}}(\mathbf{x}, t)$ is defined as the equilibrium solution to the following vector diffusion equation,

$$\begin{cases} \vec{\mathbf{v}}(\mathbf{x}, t)_t = \mu \nabla^2 \vec{\mathbf{v}}(\mathbf{x}, t) - f(\mathbf{x})(\vec{\mathbf{v}}(\mathbf{x}, t) - \nabla f(\mathbf{x})) |\nabla f(\mathbf{x})|^2, \\ \vec{\mathbf{v}}(\mathbf{x}, 0) = \nabla f(\mathbf{x}) \end{cases}$$

where μ is a blending parameter. We can note that the flow vectors of this obtained vector fields $\vec{\mathbf{v}}(\mathbf{x}, t)$ always point to the closest object boundaries, i.e. the vector $\vec{\mathbf{v}}(\mathbf{x}, t)$ indicates the correct evolving direction of the curvature flow all the time, but not the gradient direction. In Fig.1, the vectors $\vec{\mathbf{v}}(\mathbf{x}, t)$ in the GVF fields always point to the closest boundaries of object. It is clear that a large capture range to the desired edges is achieved through a diffusion process of the intensity gradient vectors that doesn't smooth the edges themselves. In the following proposed anisotropic diffusion models, the GVF fields are invariable during the image diffusion. They can be determined in advance. Thus, this will decrease the influence of noise and improve the numerical stability for image evolution.

In order to reveal its intrinsic properties, let's expand $\nabla f(|\nabla I|)$,

$$\vec{\mathbf{v}}(\mathbf{x}, t) = \nabla f(|\nabla I(\mathbf{x})|) = \lambda_1 \nabla |\nabla I(\mathbf{x})|, \quad \text{for } t > 0 \quad (1)$$

where, $\lambda_1 = \frac{|\nabla I|}{\sqrt{2\pi}\sigma_E^3} \exp\left(-\frac{|\nabla I|^2}{2\sigma_E^2}\right) > 0$. For the convenience, the Gaussian operator

G_σ in $f(\mathbf{x})$ is omitted. It will lead to the different geometry interpretations to deform Eq.(1) by the different means in the following section.

3 Diffusion Models Based on GVF Fields

3.1 Shock Filter

The heat equation will result in a smoothing process, while the inverse heat equation will lead to a deblurring process to approximate deconvolution. But the inverse heat equation is extremely ill-posed. The shock filter tries to get as close as possible to the inverse heat equation to reach a stable solution. It is formulated as, $I_t = -\text{sign}(I_{\eta\eta})|\nabla I|$, where η is the direction of the gradient.

In this section, the GVF fields are introduced in designing shock filter. The either side of Eq.(1) is dot-multiplied by the intensity normal vector, $\mathbf{N} = \nabla I / |\nabla I|$, respectively, as follows,

$$\bar{\mathbf{v}} \cdot \mathbf{N} = \lambda_1 \nabla |\nabla I| \cdot \mathbf{N} = \lambda_1 \left\langle D^2 I \frac{\nabla I}{|\nabla I|}, \mathbf{N} \right\rangle, \quad (2)$$

where $D^2 I$ denotes the Hessian of intensity I . Obviously, the above equation is equal to the second derivative of I in the direction of the intensity gradient, $I_{\eta\eta}$, up to a positive scale, λ_1 . So, the shock filter equation can be reformulated as,

$$I_t = -\text{sign}(\bar{\mathbf{v}} \cdot \mathbf{N})|\nabla I|. \quad (3)$$

When the GVF and the normal vector have the same direction, the current position is not at edges. On the other hand, when these vectors have opposite directions, the current position should be at boundaries. So, Eq.(3) lets the image develop true edges. The worse case is when the GVF is tangential to the intensity normal. Obviously, no diffusion takes place. From the implementation view, the Eq.(3) simplifies the computation of $I_{\eta\eta}$. However, as a matter of fact, the original shock filter scheme is extremely sensitive to noise because of lack of the diffusion processes (see [4] for details). While the term $(\bar{\mathbf{v}} \cdot \mathbf{N})$ in Eq.(3) is only a second derivative of I in the direction of the intensity gradient but not diffusion term, it can not remove noise. Thus the noise sensitive problem still exists in Eq.(3) as in the original scheme.

3.2 Mean Curvature Flow Equation

The mean curvature flow equation is only one of the anisotropic diffusion models. The key idea is that an image is interpreted as a collection of iso-intensity contours which can be evolved. Usually its standard form can be formulated as $I_t = \kappa |\nabla I|$, where κ is the curvature of intensity contours, $\kappa = \nabla \cdot (\nabla I / |\nabla I|)$. It has received a lot of attention because of its geometrical interpretation: the level sets of the solution move in the normal direction with a speed proportional to their mean curvature. Many theoretical aspects of this evolution equation, such as the theory of weak solutions based upon the so-called viscosity solution theory, have been summarized in [15]. In image nonlinear diffusion applications, it had been proved that the curvature flow

equation was well-posed, and the curvature flow was used as image selective smoothing filter in [16,8]. However, according to the Grayson's theorem, we know that all the structures would eventually be removed through continued application of the curvature flow scheme. In order to preserve the essential structures while remove noise, the Max/Min flow framework based on the curvature flow equation was proposed in [10]. In these above algorithms, only the pure mean curvature flow is used. Indeed, we could introduce some constraint terms in this mean curvature flow scheme just as in the active contour models. In this section, our starting point is to balance between the internal force which is from the curvature of evolution curve and external force. The GVF fields will provide the curvature flow scheme a new external force, which can overcome the noise or spurious edges effectively.

Consider the Eq.(2). We know that $(\vec{v} \cdot N)$ indicates the second derivative of I in the direction of gradient. Obviously, the sign of $(\vec{v} \cdot N)$ will change along the normal to the boundaries in the neighborhood of boundaries even if the direction of gradient N doesn't change. Thus, the GVF indicates a correct evolution direction of the curvature flow, but not the gradient direction. In our approach, the GVF is introduced as a new external force into the original curvature evolution equation directly from a force balance condition. According to Eq.(2), we can determine a contextual flow as, $C_t = (\vec{v} \cdot N)N$, where, $C \in R^2$. The interpretation of this flow is clear. An important fact is that the propagation driven by the curvature flow always takes place in the inward normal direction (i.e. $-N$). Obviously, the optimal way to reach the boundaries is to move along the direction of GVF. Because of the noise or spurious edge, the gradient vector can't always align to the GVF. Thus, the optimal propagation is obtained when the unit vector of $\vec{v}(x)$ and the inward normal direction are identical. On the other hand, the worse case occurs when $\vec{v}(x)$ is tangential to the normal, i.e. $\vec{v} \perp N$.

Under the level set framework, it is easy to introduce this contextual flow from the GVF fields into the curvature evolution equation,

$$I_t = r\kappa|\nabla I| - (1-r)\vec{v} \cdot \nabla I, \quad (4)$$

where, $0 \leq r \leq 1$. When the GVF and the inward normal have the same direction then the flow will be accelerated. On the other hand when these vectors have opposite directions, the flow will be weakened, even stopped. When the GVF is tangent to the normal then the curvature flow κ will dominate the evolution process. Besides that, it is known that the strength of the new external force, $(\vec{v} \cdot N)$, can be adjusted by the parameter λ , adaptively. For the homogeneous regions or boundaries, λ becomes so small as to weaken the external force from the GVF fields. On the contrary, it becomes too large near boundaries to ignore this new external force in Eq.(4).

The proposed scheme (4) is similar to the model presented in [7], in which a model of convolution was introduced in the mean curvature flow scheme. Indeed, whether deconvolution or deblurring processes are all sensitive to noise, while the mean curvature term could make them well-posed (see [7] for details).

From an implementation perspective, the scheme of Eq.(4) has a particular advantage over the original curvature evolution equation that since the GVF fields are invariable, the flow would not be evolved endlessly. When the internal and external force balance is reached, the flow evolution will be terminated at the object

boundaries. Furthermore, since the GVF fields provide a large capture range to the object boundaries, the flow would not fall in the noise points or spurious edges. Thus, the scheme of Eq.(4) is able to suppress noise effectively.

3.3 Perona-Malik Equation

The Perona-Malik equation (P-M Equation) introduced in [1] has been successful in dealing with the nonlinear diffusion problem in a wide range of images. The key idea is roughly to smooth out the irrelevant, homogeneous regions like the heat equation when $|\nabla I|$ is small and to enhance the boundaries instead like an inverse heat equation when $|\nabla I|$ is large. Such is the P-M Equation of the divergence form, $I_t = \nabla \cdot (g(|\nabla I|)\nabla I)$, where, $g(\cdot) > 0$ is a decreasing function of the gradient $|\nabla I|$, and

$$I(t=0) = I_0. \text{ Usually let } g(|\nabla I|) = \exp\left(-\frac{|\nabla I|^2}{2\sigma^2}\right), \text{ where } \sigma \text{ is a positive parameter that}$$

control the level of contrast of boundaries. The coefficient function $g(|\nabla I|)$ is close to 1 for $|\nabla I| << \sigma$, while $g(|\nabla I|)$ is close to 0 for $|\nabla I| >> \sigma$. A theoretical analysis shows that solutions of P-M equation can actually exhibit an inverse diffusion near boundaries, and enhance edges that have gradients greater than σ . In this section, the GVF fields are introduced in the P-M Equation. As the GVF is determined in advance, it will weaken the influence of noise or spurious edges in inverse diffusion process.

Let's expand the P-M equation, $I_t = g'|\nabla I|I_{yy} + g\Delta I$, where is the direction of gradient. In general, the first term is an inverse diffusion term for sharpening the boundaries while the second term is a Laplacian term for smoothing the regions that are relatively flat. In order to accord with the GVF fields, the coefficient function is

$$\text{defined as } g(|\nabla I|) = 1 - f(|\nabla I|) = \frac{1}{\sqrt{2\pi\sigma_E}} \exp\left(-\frac{|\nabla I|^2}{2\sigma_E^2}\right). \text{ Comparing Eq.(1) with P-M}$$

Equation, we can note that it is lack of the gradient vector term ∇I in Eq.(1). Consider the equation, $-\vec{v} \cdot \nabla I = g'|\nabla I|I_{yy}$. It is clear that the term $(-\vec{v} \cdot \nabla I)$ is the inverse diffusion term in P-M equation. Thus, the P-M equation is rewritten as

$$I_t = -\vec{v} \cdot \nabla I + g\Delta I, \quad (5)$$

where $g(\cdot)$ can be estimated directly using the gradient $|\nabla I|$. The advantages of Eq.(5) over the traditional P-M equation are very distinct,

- As only the gradient and Laplacian terms need to be computed directly from the observed images, the computation is simplified.
- The worst case for the first term is that when the GVF is tangent to the gradient direction then this inverse diffusion term is equal to 0. In fact, it is noise or spurious edges that cause these vectors orthogonal. Thus the regions around these points should be smoothed, but not enhanced.
- The scheme of Eq.(5) is an open framework. Under this scheme, the inverse diffusion term and Laplacian term can be easily controlled by redefining their coefficients respectively, $g(\cdot), g'(\cdot)$, according to smoothing effect.

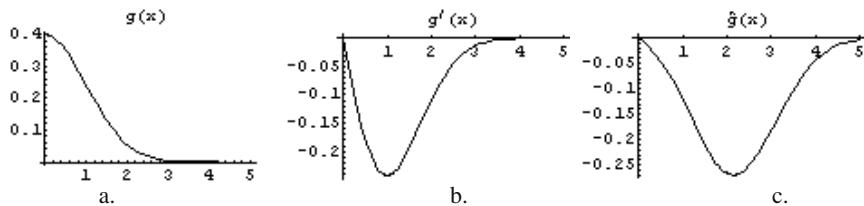


Fig. 2. The comparison of the coefficients $g(x)$, $g'(x)$, $\hat{g}(x)$, plotted as the functions of gradient magnitude. ($\sigma_E = 1$, $a = 2$)

Consider the coefficient function $g(x)$ and its derivative $g'(x)$. By adjusting the parameter σ_E , we can control the smoothing effect. But, the coefficient of the inverse diffusion term can't be adjusted freely (see Fig.2(a,b)). An intuition method is to redefine the coefficient of the inverse diffusion term as a Gaussian form,

$$\hat{g}(x) = \frac{g'(x)}{1+x} \cdot \exp\left(-\frac{a^2 - 2ax}{2\sigma_E^2}\right) = -\frac{x}{(1+x)\sqrt{2\pi\sigma_E^3}} \exp\left(-\frac{(x-a)^2}{2\sigma_E^2}\right),$$

where, a is an inverse diffusion parameter. It is clear that the inverse diffusion can be controlled freely by changing the parameter a (see Fig.2(c)).

Obviously, the parameter a is adjusted for inverse diffusing, and is independent of σ_E . Unfortunately, there is a numerical drawback in the coefficient $\hat{g}(x)$. From a numerical view, it is very difficult that the estimate σ_E in $\hat{g}(x)$ is equal to the ones in the coefficient $g'(x)$. Since the later is from the GVF. Thus, this will lead to an exponential term in $\hat{g}(x)$, e^{-bx} , $b \in R$, which leads to a very large errors for the noise points with large gradient magnitude.

Another robust method for the coefficient of the inverse diffusion term is to redefine it as follow,

$$\hat{g}(x) = sx^m g'(x), \quad m \geq 1,$$

where, s is a magnitude scale, m is a parameter which controls the location of wave crest. This function is a bimodal function, in which each wave crest is similar to Gaussian distribution. Since $x = |\nabla I|$ in our case, we only need to consider a single wave crest of this function. When parameter m is changed from one to some integer, the central of the coefficient function $\hat{g}(x)$ will be moved from low to high. They are illustrated in Fig.3.

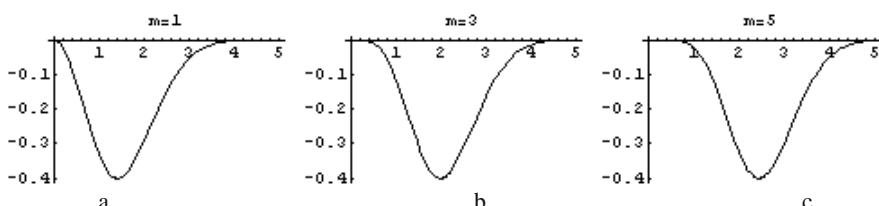


Fig. 3. The coefficient $\hat{g}(x)$ varies with parameter m changed ($\sigma_E = 1$).

Since x^m in $\hat{g}(x)$ is a power function, it could be eliminated by the exponential term e^{-bx^2} , $b \in R$ in $g'(x)$ for the noise points with large gradient magnitude. Thus, the extension of Eq.(5) can be re-expressed as,

$$I_t = -s|\nabla I|^m \bar{\mathbf{v}} \cdot \nabla I + g\Delta I, \quad m \geq 1 \quad (6)$$

It is obvious that the parameter m controls the inverse diffusion while parameter σ_E controls the smoothing effect. And this two parameters are independent each other.

However, the scheme of Eq.(6) is only one of the various forms of P-M Equation. The scheme of Eq.(5) is an open framework, in which there are plenty of other choices for designing the coefficient functions $g(x), g'(x)$.

3.4 Fairing Process and Fourth-Order Flow Model

The fairing process is derived from CAD/CAM modeling techniques. A basic approach is to minimizing the two energy functions, i.e. $\min \alpha \int_L \kappa^2 ds$ for curve fairness, and $\min \int_S (\kappa_1^2 + \kappa_2^2) ds$ where κ_1 and κ_2 are the principal curvatures for surface fairness [17]. They are usually called the least total curvature. Recently, they were introduced in the anisotropic diffusion model of the curve and surface in [19,20]. The main idea is to smooth complicated noisy surface while preserve sharp geometric features. Under a variational framework, a fourth order PDE's can be deduced from the least total curvature. In this section, we focus on the fourth order flow model in the plane, which will be introduced in the image anisotropic diffusion.

In [21], the fourth order flow was presented as the intrinsic Laplacian of curvature under the level set framework,

$$\kappa_{ss} = \frac{\kappa_{xx}\phi_y^2 - 2\kappa_{xy}\phi_x\phi_y + \kappa_{yy}\phi_x^2}{\phi_x^2 + \phi_y^2} - \kappa \frac{\kappa_x\phi_x + \kappa_y\phi_y}{\sqrt{\phi_x^2 + \phi_y^2}}$$

which is the second derivative of the local curvature κ with respect to arc length parameter s . The particular geometric property of this flow is to improve the isoperimetric ratio, but not to reduce the enclosed area like the mean curvature flow. For comparison, the evolutions of a star shape under the intrinsic Laplacian of curvature flow, $I_t = \kappa_{ss}|\nabla I|$, and the mean curvature flow, $I_t = \kappa|\nabla I|$, are shown in Fig.4.

It is obvious that the flow under the intrinsic Laplacian of curvature will converge to a circle finally. Because the isoperimetric ratio of circle is maximum if the perimeter is fixed. And the derivatives of the curvature converge uniformly to zero. Thus, the final solution to the flow under the intrinsic Laplacian of curvature should be a circle. In image diffusion, this fourth order flow model could preserve the boundaries of shapes but not smooth out them. Simultaneously, some small oscillations around the boundaries would be smoothed out.



a. flow under the intrinsic Laplacian of curvature (This example is from [21].)



b. flow under the mean curvature

Fig. 4. Evolution of star shape, iteration=15000 in (a), while iteration=1000 in (b)

However, owing to the fourth order derivative term in evolution equation, it becomes highly sensitive to errors. And the fourth order derivative term leads to numerical schemes with very small time steps. In Fig.4(a), the space step $\Delta x = 0.0667$, the time step $\Delta t = 5 \times 10^{-6}$, and more than 40 reinitializations are used. In fact, it is ill-posed to minimize the total squared curvature $\int_L K^2 ds$ in the plane closed curves, (i.e. plane curve raveling). Because the total squared curvature is scale-dependent, it can be reduced as far as the gradient flow inflates any closed curve without limit. In order to make it well-posed, the total squared curvature was modified as, $\int_L (\kappa^2 + \alpha^2) ds$, in [13] and the corresponding gradient flow was deduced under a variational framework,

$$C_t = (\kappa_{ss} + \kappa(\kappa^2 - \alpha^2)/2)\mathbf{N}, \quad (7)$$

where C is a plane closed evolving curve, \mathbf{N} is normal vector and $\alpha \neq 0$ is the penalty function to make the problem well-posed. A few important conclusions to Eq.(7) from [13] need to be highlighted as follows,

- The long time existence of solution to Eq.(7) is proven. And a stationary solution can be reached.
- If the descent flow corresponding to the ‘pure’ energy $\int_L K^2 ds$ is considered, the normal speed is simply $F = \kappa_{ss} + \kappa^3/2$.
- Not only can the flow smooth the embedded curves, but also the immersed curves.

The intrinsic Laplacian of curvature has been introduced in active contours as a frigid force for 2D and 3D segmentation in [18]. In fact, it also could be applied in the anisotropic diffusion of images. Because of the isoperimetric property from the intrinsic Laplacian of curvature term, the boundaries of shapes in the evolving image will become vivid. We will deduce the intrinsic Laplacian of curvature directly from the GVF fields for simplifying computation and numerical stability.

Consider the GVF form of Eq.(2). We can obtain the second derivative of I in the direction orthogonal to the gradient, which can be formulated as, $\lambda_1 I_{\xi\xi} = \lambda_1 \Delta I - \vec{v} \cdot \mathbf{N}$. Because $I_{\xi\xi}$ can be written as a “quasi divergence form” [16], $I_{\xi\xi} = |\nabla I| \nabla \cdot (\nabla I / |\nabla I|)$, we have, $\lambda_1 \Delta I - \vec{v} \cdot \mathbf{N} \equiv \lambda_2 \hat{\kappa}$, where, $\lambda_2 = |\nabla I| \lambda_1 > 0$, $\hat{\kappa} = \nabla \cdot (\nabla I / |\nabla I|)$, which can be looked upon as a curvature flow. In general, the curvature flow evolves along the direction of gradient. The above equation can be defined as a force field along the direction of gradient, $\mathbf{E} = \lambda_2 \hat{\kappa} \mathbf{N}$. The derivative of the field \mathbf{E} with respect to the arc length is followed from the Frenet-Serret Formulation,

$$\mathbf{E}_s = (\lambda_2 \hat{\kappa})_s \mathbf{N} - (\lambda_2 \hat{\kappa}) \kappa \mathbf{T},$$

where, \mathbf{T} is an unit tangent vector, and $\mathbf{T} \cdot \mathbf{N} = 0$, κ is the intensity contour curvature which is only from the observed images. Furthermore, the second derivative of the fields \mathbf{E} with respect to the arc length is yielded,

$$\mathbf{E}_{ss} = ((\lambda_2 \hat{\kappa})_{ss} - (\lambda_2 \hat{\kappa}) \kappa^2) \mathbf{N} - (2(\lambda_2 \hat{\kappa})_s \kappa + (\lambda_2 \hat{\kappa}) \kappa_s) \mathbf{T}.$$

For the gradient flow, we have, $\mathbf{E}_{ss} \cdot \mathbf{N} = (\lambda_2 \hat{\kappa})_{ss} - (\lambda_2 \hat{\kappa}) \kappa^2$. Obviously, it is the normal speed of the gradient flow corresponding to the ‘pure’ energy $\int \kappa^2 ds$. Denote $K = \lambda_2 \hat{\kappa}$ for convenience. The second derivative of K can be expressed,

$$K_{ss} = \frac{K_{xx} I_y^2 + K_{yy} I_x^2 - 2K_{xy} I_x I_y}{I_x^2 + I_y^2} - \kappa \frac{K_x I_x + K_y I_y}{\sqrt{I_x^2 + I_y^2}},$$

where K can be estimated from the GVF and the observed image, \mathbf{N} is from the observed image and λ_1, λ_2 can be estimated using the gradient $|\nabla I|$. Hence, the flow under the intrinsic Laplacian of curvature is rewritten as,

$$I_t = -(\mathbf{E}_{ss} \cdot \mathbf{N}) |\nabla I| = (K \kappa^2 - K_{ss}) |\nabla I|.$$

It can be noticed that all the fourth order derivate terms in the above intrinsic Laplacian of curvature flow are estimated from the GVF fields, but not directly from the observed image. This improves the numerical stability effectively.

However, the above intrinsic Laplacian of curvature flow equation is ill-posed. Compared to the scheme of Eq.(7), it is lack of a mean curvature term. In order to make it well-posed, we couple the scheme of Eq.(4) to the above equation. It can be easily addressed,

$$I_t = \beta(r \kappa |\nabla I| - (1-r) \vec{v} \cdot \nabla I) + (K \kappa^2 - K_{ss}) |\nabla I|, \quad (8)$$

where β is a constant that balances the contribution between the mean curvature flow and the fourth order flow.

4 Experiments

We first illustrate the GVF-based shock filter on a Mammographic image in Fig.5(a-c). Since this kind of medical images is usually very blurry, which need be deblurred, the shock filter is a stable deconvolution filter. By way of comparison, the experimental result of the original shock filter is shown too. It could be noticed that the original shock filter over-enhanced the details in Fig.5(b), while the scheme of (3) could enhance the essential structures and suppress some trivial details in Fig.5(c). Fig.5(h) demonstrate that the scheme Eq.(3) can reach a steady state solution more quickly than the original scheme with the evolution error, which is calculated as,

$$\text{error}(t) = \frac{1}{M \times N} \sum_{i,j}^{M,N} |I_{i,j}^{(t)} - I_{i,j}^{(0)}|, \text{ where } M \text{ and } N \text{ are the width and height of image}$$

respectively (the following experiments all adopted this error equation to generate error diagrams).

The original mean curvature flow and the proposed scheme of (4) are illustrated on a noisy and blurry image in Fig.5(d-g) respectively. The original image is firstly degraded with Gaussian Noise (zero mean, with 0.1 variance), then blurred by Gaussian lowpass filter with 0.25 variance. We adopted $I_t = \kappa |\nabla I|$ in section 3.2 as the original mean curvature flow scheme. It can be noticed that the features of the lily image are enhanced and denoised effectively by the scheme of (4) in Fig.5(g), while all the features are smoothed out by the original scheme in Fig.5(f). The error diagrams in Fig.5(i) demonstrate that the scheme of (4) could reach a steady state solution and preserve the essential structures of shapes, while the original curvature flow scheme would eventually smooth out all the information.

In the experiments of the P-M equation, we first compared the original scheme with the proposed scheme of (5). The original water lily image is degraded with Gaussian noise, and blurred as in Fig.5(e). The diffusion results are shown in Fig.6(a,b). Their diffusion effects seem to be close. But then, the error diagram in Fig.6(c) demonstrates that the scheme of (5) could reach a steady state solution more quickly than the original P-M equation.

Then, in the successive experiments, we also illustrated the scheme of (6) on the water lily image, which is blurred and degraded with Gaussian noise as in Fig.5(e). The experimental results are shown in Fig.7. Obviously, when the coefficient m is becoming large, some details with large gradient are enhanced, while others with small gradient are eroded gradually. Since the center of the inverse diffusion term in Eq.(6) would be moved along with the change of the coefficient m .

In the experiment of the 4th order flow scheme, we also illustrated the scheme of (8) on the noised and blurry water-lily image as in Fig.5(e). The result is compared with the one of the scheme Eq.(5) in Fig.8. Obviously, because of the intrinsic Laplacian of curvature term in Eq.(8), the boundaries of objects become very vivid in Fig.8(b). It is clear to indicate that the isoperimetric property from the intrinsic Laplacian of curvature term would make the boundaries of shapes enhanced and smooth but not eroded in image anisotropic diffusion.

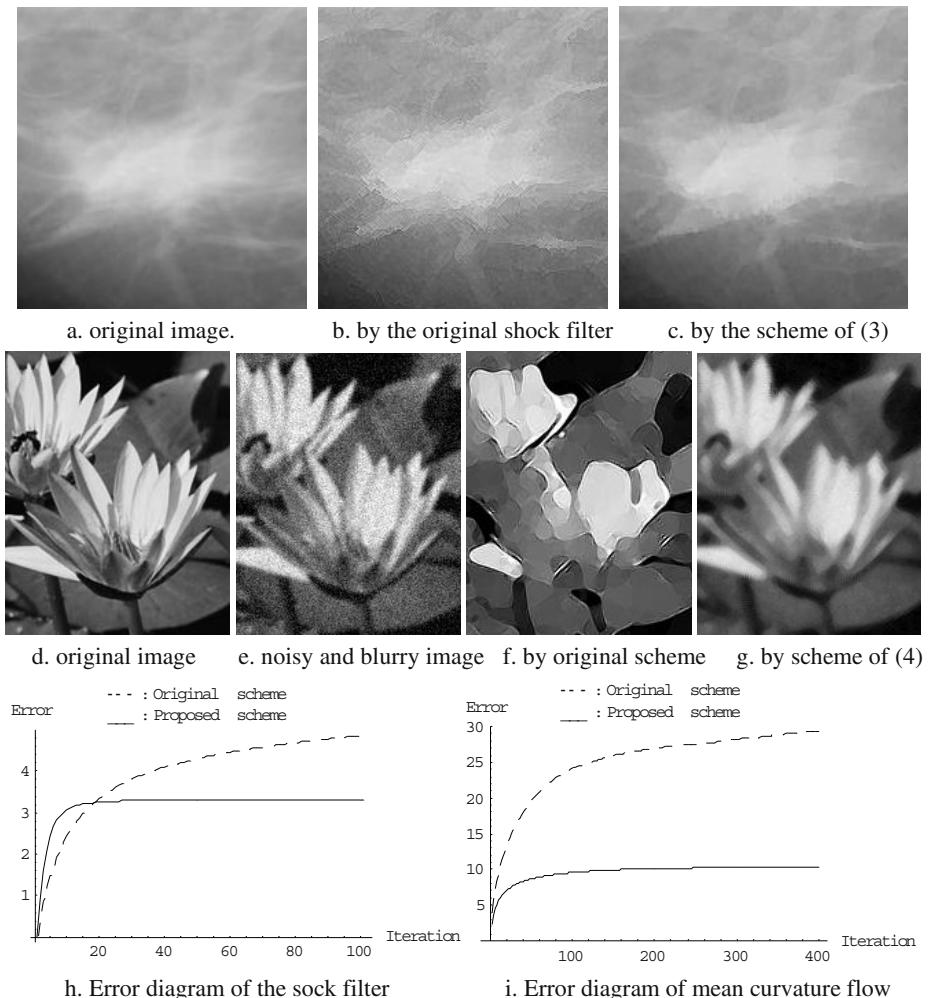


Fig. 5. Evolutions of shock filter and mean curvature flow scheme

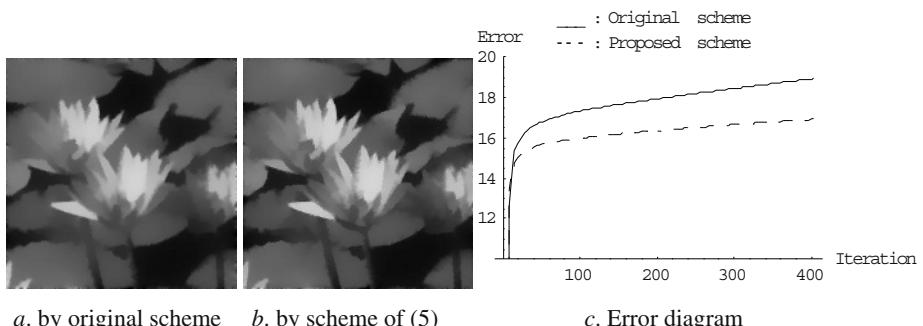


Fig. 6. Evolutions of P-M equation with $\sigma_E = 0.45$, (a) iteration=150, (b) iteration=200

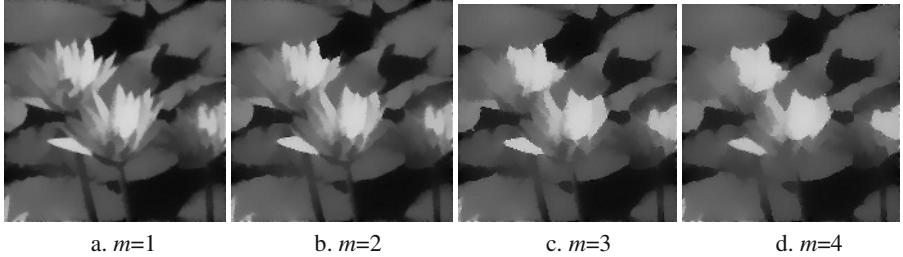


Fig. 7. Evolutions of the scheme of (6) with $\sigma_E = 0.45$, $s=1$ and iteration=200



Fig. 8. Evolution of schemes (5) and (8) on water lily image at iteration=200

5 Conclusions

In this paper, we firstly introduced the gradient vector flow (GVF) fields in the image anisotropic diffusion. Some well-known PDE's diffusion models were reformulated based on the GVF fields, such as the shock filter, the mean curvature flow and the P-M diffusion model. The particular advantages that the GVF leads to are to simplify computation, improve numerical stability, and perform well on noisy images. One of the most distinct advantages is that the proposed GVF-based anisotropic diffusion models could reach a steady state solution more quickly than the original ones. Besides that, in order to enhance and smooth the boundaries of object but not to erode them, the intrinsic Laplacian of curvature was firstly introduced in the anisotropic diffusion of images. Since this flow contains a fourth order derivative term, it is very sensitive to errors. We can obtain the robust estimate of this flow from the GVF fields. The experiments indicate that our proposed models are robust and practical on account of the GVF fields.

In future works, the GVF-based P-M diffusion equation is going on being concerned. How to design the coefficient in reverse diffusion term is always a key problem. In addition, we will try to apply these proposed diffusion models on 3D volume data for visualization. Since the classification process is critical to direct volume rendering, while the anisotropic diffusion could provide us the desired classification results.

Acknowledgement. The work was supported in part by a NSFC grant 60203003.

References

1. P. Perona & J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. On PAMI, Vol.12, No.7, pp.629-639, 1990
2. L. Alvarez and L. Mazorra, Signal and image restoration using shock filters and anisotropic diffusion, SIAM J. Numer. Anal., Vol.31, No.2, pp.590-605, 1994
3. P. Kornprobst, R. Deriche and G. Aubert, Image coupling, restoration and enhancement via PDE's, in Proc. Int. Conf. On Image Processing 1997, pp.458-461, Santa-Barbara, USA
4. G. Gilboa, N.A. Sochen and Y.Y. Zeevi, Regularized shock filters and complex diffusion, in Proc. Europe Conf. On Computer Vision 2002, A. Heyden et al. (Eds.) LNCS 2350, pp.399-413, Springer-Verlag Berlin Heidelberg
5. G. Gilboa, N.A. Sochen and Y.Y. Zeevi, Forward-and-backward diffusion processes for adaptive image enhancement and denoising, IEEE Trans. On Image Processing, Vol.11, No.7, pp.689-703, 2002
6. R.A. Carmona and S. Zhong, Adaptive smoothing respecting feature directions, IEEE Trans. On Image Processing, Vol.7, No.3, pp.353-358, 1998
7. A. Marquina and S. Osher, Explicit Algorithms For A New Time Dependent Model Based On Level Set Motion For Nonlinear Deblurring And Noise Removal, SIAM J. Sci. Comput., Vol.22, No.2, pp.387-405, 2000
8. L. Alvarez, F. Guichard, P-L. Lions and J-M. Morel, Axioms and fundamental equation of image processing, in Archive for rational mechanics and analysis, Vol.123, pp.199-257, Springer-Verlag, 1993
9. J. Weickert, B. Romeny and M.A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, IEEE Trans. On Image Processing, Vol.7, No.3, pp. 398-409, 1998
10. R. Malladi and J.A. Sethian, A unified approach to noise removal, image enhancement, and shape recovery, IEEE Trans. On Image Processing, Vol.5, No.11, pp.1554-1568, 1996
11. C. Xu and J.L. Prince, Snake, Shapes, and Gradient vector flow, IEEE Trans. On Image Processing, Vol.7, No.3, pp.359-369, 1998
12. S. Osher and L. Rudin, Feature-oriented image enhancement using shock filters, SIAM J. Numer. Anal., Vol.27, pp.919-940, 1990
13. A. Polden, Curves and surfaces of least total curvature and fourth-order flows, Ph.D. Thesis, University of Tubingen, Germany, 1996
14. N. Paragios, O. Mellina-Gottardo and V. Ramesh, Gradient vector flow fast geodesic active contours, in Proc. Int. Conf. On Computer Vision 2001, Vancouver, Canada, 2001
15. J.A. Sethian, Level set methods and fast matching methods, Cambridge University Press, 1999
16. L. Alvarez, P.-L. Lions and J.-M. Morel, Image selective smoothing and edge detection by nonlinear diffusion II, SIAM J. Numer. Anal., Vol.29, No.3, pp.845-866, 1992
17. N.J. Lott and D.I. Pullin, Method for fairing B-Spline surfaces, CAD, Vol.20, No.10, pp.597-604, 1988
18. C. Xu, J.L. Prince and A. Yezzi, A summary of geometric level-set analogues for a general class of parametric active contour and surface models, in Proc. 1st IEEE workshop on Variational and Level Set Methods in Computer Vision, pp.104-111, Vancouver, Canada, 2001
19. G. Taubin, A signal processing approach to fair surface design, in Proc. of SIGGRAPH'95, pp.351-358, Los Angeles, California, USA, 1995
20. M. Desbrun and M. Meyer et al., Implicit fairing of irregular meshes diffusion and curvature flow, in Proc. of SIGGRAPH'99, pp.317-324, Los Angeles, California, USA, 1999
21. D.L. Chopp and J.A. Sethian, Motion by intrinsic Laplacian of curvature, Interfaces and Free Boundaries, Vol.1, pp.1-18, Oxford University Press, 1999

Optimal Importance Sampling for Tracking in Image Sequences: Application to Point Tracking

Elise Arnaud and Etienne Mémin

IRISA, Université de Rennes 1,
Campus de Beaulieu,
35 042 Rennes Cedex, France
{earnaud,memin}@irisa.fr

Abstract. In this paper, we propose a particle filtering approach for tracking applications in image sequences. The system we propose combines a measurement equation and a dynamic equation which both depend on the image sequence. Taking into account several possible observations, the likelihood is modeled as a linear combination of Gaussian laws. Such a model allows inferring an analytic expression of the optimal importance function used in the diffusion process of the particle filter. It also enables building a relevant approximation of a validation gate. We demonstrate the significance of this model for a point tracking application.

1 Introduction

When tracking features of any kind from image sequences, several specific problems appear. In particular, one has to face difficult and ambiguous situations generated by cluttered backgrounds, occlusions, large geometric deformations, illumination changes or noisy data. To design trackers robust to outliers and occlusions, a classical way consists in resorting to stochastic filtering techniques such as Kalman filter [13,15] or sequential Monte Carlo approximation methods (called particle filters) [7,10,11,16].

Resorting to stochastic filters consists in modeling the problem by a discrete hidden Markov state process $\mathbf{x}_{0:n} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\}$ of transition equation $p(\mathbf{x}_k | \mathbf{x}_{k-1})$. The sequence of incomplete measurements of the state is denoted $\mathbf{z}_{1:n} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$, of marginal conditional distribution $p(\mathbf{z}_k | \mathbf{x}_k)$. Stochastic filters give efficient procedures to accurately approximate the posterior probability density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$. This problem may be solved exactly through a Bayesian recursive solution, named the optimal filter [10]. In the case of linear Gaussian models, the Kalman filter [1] gives the optimal solution since the distribution of interest $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ is Gaussian. In the nonlinear case, an efficient approximation consists in resorting to sequential Monte Carlo techniques [4,9]. These methods consists in approximating $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ in terms of a finite weighted sum of Diracs centered in elements of the state space named particles. At each discrete instant, the particles are displaced according to a probability density function named *importance function* and the corresponding weights are updated through the likelihood.

For a given problem, a relevant expression of the *importance function* is a crucial point to achieve efficient and robust particle filters. As a matter of fact, since this function is used for the diffusion of the particle swarm, the particle repartition - or the state-space exploration – strongly depends on it. It can be demonstrated that the *optimal*

importance function in the sense of a minimal weight variance criterion is the distribution $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ [9]. As it will be demonstrated in the experimental section, the knowledge of this density improves significantly the obtained tracking results for a point tracking application.

However, the expression of $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ is totally unknown in most vision applications. In such a context, the importance function is simply fixed to the prediction density $p(\mathbf{x}_k | \mathbf{x}_{k-1})$. This constitutes a crude model which is counterbalanced by a systematic re-sampling step of the particles together with sound models of highly multimodal likelihood [7,11,16].

In this paper, an opposite choice is proposed. We investigate simpler forms of likelihood but for which the optimal importance function may be inferred. The considered likelihood is a linear combination of Gaussian laws. In addition, such a modelization allows expressing a validation gate in a simple way. A validation gate defines a bounded research region where the measurements are looked for at each time instant.

Besides, it is interesting to focus on features for which none dynamic model can be set *a priori* or even learned. This is the case when considering the most general situation without any knowledge on the involved sequence. To tackle this situation, we propose to rely on dynamic models directly estimated from the image sequence.

For point tracking applications, such a choice is all the more interesting that any dynamic model of a feature point is very difficult to establish without any *a priori* knowledge on the evolution law of the surrounding object. As a consequence, the system we propose for point tracking depends entirely on the image data. It combines (i) a state equation which relies on a local polynomial velocity model, estimated from the image sequence and (ii) a measurement equation ensuing from a correlation surface between a reference frame and the current frame. The association of these two approaches allows dealing with trajectories undergoing abrupt changes, occlusions and cluttered background situations.

The proposed method has been applied and validated on different sequences. It has been compared to the Shi-Tomasi-Kanade tracker [17] and to a CONDENSATION-like algorithm [11].

2 Nonlinear Image Sequence Based Filtering

Classical formulation of filtering systems implies to *a priori* know the density $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$, and to be able to extract, from the image sequence, an information used as a measurement of the state. However, in our point of view, feature tracking from image sequences may require in some cases to slightly modify the traditional filtering framework. These modifications are motivated by the fact that an *a priori* state model is not always available, especially during the tracking of features whose nature is not previously known. A solution to this problem may be devised relying on an estimation from the image sequences data of the target dynamics [2,3]. In that case, it is important to distinguish (i) the observation data which constitute the measurements of the state from (ii) the data used to extract such a dynamics model. These two pieces of information are of different kinds even if they are both estimated from the image sequence – and therefore depend statistically on each other. In this unconventional situation where

dynamics and measurements are both captured from the sequence, it is possible to build a proper filtering framework by considering a conditioning with respect to the image sequence data.

2.1 Image Sequence Based Filtering

Let us first fix our notations. We note \mathbf{I}_k an image obtained at time k . $\mathbf{I}_{0:n}$ represents the finite sequence of random variables $\{\mathbf{I}_k, k = 0, \dots, n\}$. Knowing a realization of $\mathbf{I}_{0:k}$, our tracking problem is modeled by the following dynamic and measurement equation:

$$\begin{aligned}\mathbf{x}_k &= f_k^{\mathbf{I}_{0:k}}(\mathbf{x}_{k-1}, \mathbf{w}_k^{\mathbf{I}_{0:k}}), \\ \mathbf{z}_k &= h_k^{\mathbf{I}_{0:k}}(\mathbf{x}_k, \mathbf{v}_k^{\mathbf{I}_{0:k}}).\end{aligned}$$

At each time k , a realization of \mathbf{z}_k is provided by an estimation process based on image sequence $\mathbf{I}_{0:k}$. Functions $f_k^{\mathbf{I}_{0:k}}$ and $h_k^{\mathbf{I}_{0:k}}$ are assumed to be any kind of possibly nonlinear functions. These functions may be estimated from $\mathbf{I}_{0:k}$. The state noise $\mathbf{w}_k^{\mathbf{I}_{0:k}}$ and the measurement noise $\mathbf{v}_k^{\mathbf{I}_{0:k}}$ may also depend on $\mathbf{I}_{0:k}$ as well, and are not necessarily Gaussian. We assume that the associated probability distributions are such that

$$\begin{aligned}p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1}, \mathbf{I}_{0:n}) &= p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:n}), \\ p(\mathbf{z}_k | \mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}, \mathbf{I}_{0:n}) &= p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:n}).\end{aligned}$$

By analogy with the classical filtering formulation the Markovian assumption, as well as the conditional independence of the observations are maintained conditionally to the sequence. A causal hypothesis with respect to the temporal image acquisition is added. Such an hypothesis means that the state \mathbf{x}_k and the measurement \mathbf{z}_k are assumed to be independent from $\mathbf{I}_{k+1:n}$. The optimal filter's equations can be applied to the proposed model. The expected posterior reads now $p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{I}_{0:k})$. Supposing $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k-1})$ known, the recursive Bayesian optimal solution is:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{I}_{0:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k}) \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k-1}) d\mathbf{x}_{k-1}}{\int p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k}) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k}) d\mathbf{x}_k}.$$

To solve this conditional tracking problem, standard filters have to be derived in a conditional version. The linear version of this framework, relying on a linear minimal conditional variance estimator, is presented in [2,3]. The nonlinear version is implemented with a particle filter and is called *Conditional NonLinear Filter*.

2.2 Conditional NonLinear Filter

Facing a system with a nonlinear dynamic and/or a nonlinear likelihood, it is not possible anymore to construct an exact recursive expression of the posterior density function of the state given all available past data. To overcome these computational difficulties, particle filtering techniques propose to implement recursively an approximation of this density (see [4,9] for an extended review). These methods consist in approximating the

posterior density by a finite weighted sum of Dirac centered on hypothesized trajectories – called particles – of the initial system \mathbf{x}_0 :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{I}_{0:k}) \approx \sum_{i=1}^N w_k^{(i)} \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}).$$

At each time instant (or iteration), the set of particles $\{\mathbf{x}_{0:k}^{(i)}, i = 1, \dots, N\}$ is drawn from an approximation of the true distribution $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}, \mathbf{I}_{0:k})$, called the *importance function* and denoted $\pi(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}, \mathbf{I}_{0:k})$. The closer is the approximation from the true distribution, the more efficient is the filter. The particle weights $w_k^{(i)}$ account for the deviation with regard to the unknown true distribution. The weights are updated according to importance sampling principle:

$$w_k^{(i)} = \frac{p(\mathbf{z}_{1:k} | \mathbf{x}_{0:k}^{(i)}, \mathbf{I}_{0:k}) p(\mathbf{x}_{0:k}^{(i)} | \mathbf{I}_{0:k})}{\pi(\mathbf{x}_{0:k}^{(i)} | \mathbf{z}_{1:k}, \mathbf{I}_{0:k})}.$$

Choosing an importance function that recursively factorizes such as:

$$\pi(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}, \mathbf{I}_{0:k}) = \pi(\mathbf{x}_{0:k-1} | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k-1}) \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}, \mathbf{I}_{0:k})$$

allows recursive evaluations in time of the particle weights as new measurements \mathbf{z}_k become available. Such an expression implies naturally a causal assumption of the importance function w.r.t. observations and image data. The recursive weights read then:

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k | \mathbf{x}_k^{(i)}, \mathbf{I}_{0:k}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{I}_{0:k}) / \pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}, \mathbf{I}_{0:k}).$$

Unfortunately, such a recursive assumption of the importance function induces an increase over time of the weight variance [12]. In practice, this makes the number of significant particles decrease dramatically over time. To limit such a degeneracy, two methods have been proposed (here presented in the conditional framework).

A first solution consists in selecting an *optimal* importance function which minimizes the variance of the weights conditioned upon $\mathbf{x}_{0:k-1}$, $\mathbf{z}_{1:k}$ and $\mathbf{I}_{0:k}$ in our case. It is then possible to demonstrate that $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k})$ corresponds to this optimal distribution. With this distribution, the recursive formulation of w_k becomes then:

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{I}_{0:k}). \quad (1)$$

The problem with this approach is related to the fact that it requires to be able to sample from the optimal importance function $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k})$, and to have an expression of $p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k})$. In vision applications, the optimal importance function is usually not accessible. The importance function is then set to the prediction density (i.e. $\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = p(\mathbf{x}_k | \mathbf{x}_{k-1})$). Such a choice excludes the measurements from the diffusion step.

A second solution to tackle the problem of weight variance increase relies on the use of re-sampling methods. Such methods consist in removing trajectories with weak normalized weights, and in adding copies of the trajectories associated to strong weights,

as soon as the number of significant particles is too weak [9]. Obviously, these two solutions may be coupled for a better efficiency. Nevertheless it is important to outline that the resampling step introduces errors and is only the results of the discrepancy between the unknown true pdf and the importance function. As a consequence, the resampling step is necessary in practice, but should be used as rarely as possible. It can be noticed that setting the importance function to the diffusion process and resampling at each iteration leads to weight directly the particles with the likelihood. This choice has been made in the CONDENSATION algorithm [11].

As mentioned previously, it may be beneficial to know the expression of the optimal importance function. As developed in the next section, it is possible to infer this function for a specific class of systems.

3 Gaussian Systems and Optimal Importance Function

Filtering models for tracking in vision applications are traditionally composed of a simple dynamic and a highly multimodal and complex likelihood [3]. For such models, an evaluation of the optimal importance function is usually not accessible. In this section, we present some filtering systems relying on a class of likelihoods (eventually multimodal) for which it is possible to sample from the optimal importance function.

3.1 Gaussian System with Monomodal Likelihood

We consider first a conditional nonlinear system, composed of a nonlinear state equation, with an additive Gaussian noise, and a linear Gaussian likelihood:

$$\mathbf{x}_k = f_k^{\mathbf{I}_{0:k}}(\mathbf{x}_{k-1}) + \mathbf{w}_k^{\mathbf{I}_{0:k}}, \quad \mathbf{w}_k^{\mathbf{I}_{0:k}} \sim \mathcal{N}(\mathbf{w}_k^{\mathbf{I}_{0:k}}; \mathbf{0}, Q_k^{\mathbf{I}_{0:k}}) \quad (2)$$

$$\mathbf{z}_k = H_k^{\mathbf{I}_{0:k}} \mathbf{x}_k + \mathbf{v}_k^{\mathbf{I}_{0:k}}, \quad \mathbf{v}_k^{\mathbf{I}_{0:k}} \sim \mathcal{N}(\mathbf{v}_k^{\mathbf{I}_{0:k}}; \mathbf{0}, R_k^{\mathbf{I}_{0:k}}). \quad (3)$$

For these models the analytic expression of the optimal importance function may be inferred. As a matter of fact, noticing that:

$$p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) = \int p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k}) p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) d\mathbf{x}_k, \quad (4)$$

we deduce:

$$\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k} \sim \mathcal{N}(\mathbf{z}_k; H_k f_k(\mathbf{x}_{k-1}), R_k + H_k Q_k H_k^t), \quad (5)$$

which yields a simple tractable expression for the weight calculation (1) (for the sake of clarity, the index $\mathbf{I}_{0:k}$ has been omitted). As for the optimal importance function we have:

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k}) = p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k}) p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) / p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) \quad (6)$$

and thus,

$$\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k} \sim \mathcal{N}(\mathbf{x}_k; \mu_k, \Sigma_k), \quad (7)$$

with

$$\begin{aligned}\Sigma_k &= (Q_k^{-1} + H_k^t R_k^{-1} H_k)^{-1} \\ \mu_k &= \Sigma_k (Q_k^{-1} f_k(\mathbf{x}_{k-1}) + H_k^t R_k^{-1} \mathbf{z}_k).\end{aligned}$$

In that particular case, all the expressions used in the diffusion process (7), and in the update step (5) are Gaussian. The filter corresponding to these models is therefore particularly simple to implement. The unconditional version of this result is described in [9].

3.2 Extension to Multimodal Likelihood

Considering only one single measurement can be too restrictive facing ambiguous situations or cluttered background. We describe here an extension of the previous monomodal case to devise a multimodal likelihood.

Let us now consider a vector of M measurements $\mathbf{z}_k = \{\mathbf{z}_{k,1}, \mathbf{z}_{k,2}, \dots, \mathbf{z}_{k,M}\}$. As it is commonly done in target tracking [6] and computer vision [11], we assume that a unique measurement corresponds to a true match and that the others are due to false alarms or clutter. Noting Φ_k a random variable which takes its values in $0, \dots, M$, we designate by $p(\Phi_k = m)$ the probability that measurement $\mathbf{z}_{k,m}$ corresponds to the *true* measurement at time k ; $p(\Phi_k = 0)$ is the probability that none of the measurements corresponds to the *true* one. Denoting $p_{k,m} = p(\Phi_k = m | \mathbf{x}_k, \mathbf{I}_{0:k})$, and assuming that $\forall m = 1, \dots, M$, the measurements $\mathbf{z}_{k,1:M}$ are independent conditionally to $\mathbf{x}_k, \mathbf{I}_{0:k}$ and $\Phi_k = m$, then the likelihood can be written as:

$$\begin{aligned}p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k}) &= p_{k,0} p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k}, \Phi_k = 0) \\ &+ \sum_{m=1}^M \{p_{k,m} p(\mathbf{z}_{k,m} | \mathbf{x}_k, \mathbf{I}_{0:k}, \Phi_k = m) \prod_{j \neq m} p(\mathbf{z}_{k,j} | \mathbf{x}_k, \mathbf{I}_{0:k}, \Phi_k = m)\}. \quad (8)\end{aligned}$$

In order to devise a tractable likelihood for which an analytic expression of the optimal importance function may be derived, we make the following hypothesis. We assume that (i) the set of mode occurrence probabilities $\{p_{k,i}, i = 1, \dots, M\}$ is estimated from the images at each instant ; (ii) the probability of having no *true* measurement is set to zero ($p_{k,0} = 0$). Such a choice differs from classical tracking assumptions [6,11] and may be of problematic in case of occlusions. Nevertheless, as we will see it, this potential deficiency is well compensated by an efficient estimation of the measurement noise covariances. We also assume that (iii) considering $\mathbf{z}_{k,m}$ as being the *true* target-originated observation, it is distributed according to a Gaussian law of mean $H_{k,m} \mathbf{x}_k$ and covariance $R_{k,m}$. As a last hypothesis (iv), we assume that the false alarms are uniformly distributed over a measurement region (also called gate) at time k . The total area of the validation gate V_k will be denoted $|V_k|$.

All these assumptions lead to an observation model which can be written as a linear combination of Gaussian laws:

$$\begin{aligned} \mathbf{z}_k | \mathbf{x}_k, \mathbf{I}_{0:k} &\rightsquigarrow \sum_{m=1}^M \left[p_{k,m} \mathcal{N}(\mathbf{z}_{k,m}; H_{k,m} \mathbf{x}_k, R_{k,m}) \prod_{j \neq m} \frac{\mathbb{1}_{\mathbf{z}_{k,j} \in V_k}}{|V_k|} \right] \\ &= \frac{1}{|V_k|^{M-1}} \sum_{m=1}^M p_{k,m} \mathcal{N}(\mathbf{z}_{k,m}; H_{k,m} \mathbf{x}_k, R_{k,m}). \end{aligned} \quad (9)$$

In the same way as for the monomodal measurement equation (§3.1), it is possible for such a likelihood associated to a Gaussian state equation of form (2) to know the optimal importance function. Let us remind that in our case the considered diffusion process requires to evaluate $p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k})$ and to sample from $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k})$. Applying identity (4), with expression (9), the density used for the weight recursion reads:

$$\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k} \rightsquigarrow \frac{1}{|V_k|^{M-1}} \sum_{m=1}^M p_{k,m} \mathcal{N}(\mathbf{z}_{k,m}; H_{k,m} f_k(\mathbf{x}_{k-1}), H_{k,m} R_{k,m} H_{k,m}^t + Q_k). \quad (10)$$

The optimal importance function is deduced using identity (6) and expression (10):

$$\begin{aligned} \mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k} &\rightsquigarrow \\ &\frac{\mathcal{N}(\mathbf{x}_k; f_k(\mathbf{x}_{k-1}), Q_k) \sum_{m=1}^M p_{k,m} \mathcal{N}(\mathbf{x}_k; H_{k,m}^{-1} \mathbf{z}_k, H_{k,m}^{-1} R_{k,m} H_{k,m}^{-1 t})}{|V_k|^{M-1} p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k})} \end{aligned}$$

Through Gaussian identities this expression reads as a Gaussian mixture of the form:

$$\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k, \mathbf{I}_{0:k} \rightsquigarrow \sum_{m=1}^M p_{k,m} \frac{\alpha_{k,m}}{S} \mathcal{N}(\mathbf{x}_k; \mu_{k,m}, \Sigma_{k,m}) \quad (11)$$

with

$$\left\{ \begin{array}{l} \Sigma_{k,m} = (Q_k^{-1} + H_{k,m}^t R_{k,m}^{-1} H_{k,m})^{-1} \\ \mu_{k,m} = \Sigma_{k,m} (Q_k^{-1} f_k(\mathbf{x}_{k-1}) + H_{k,m}^t R_{k,m}^{-1} \mathbf{z}_{k,m}) \\ S = |V_k|^{M-1} p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) \\ \alpha_{k,m} = \frac{|\Sigma_{k,m}|^{\frac{1}{2}}}{2\pi|R_{k,m}|^{\frac{1}{2}}|Q_k|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\|f_k(\mathbf{x}_{k-1})\|_{Q_k^{-1}}^2 + \|\mathbf{z}_{k,m}\|_{R_{k,m}^{-1}}^2 - \|\mu_{k,m}\|_{\Sigma_{k,m}^{-1}}^2)) \end{array} \right.$$

Let us point out that the proposed systems lead to a simple implementation as the involved distributions are all combinations of Gaussian laws. In addition, as described in the next subsection, such systems allow to define a relevant validation gate for the measurements.

3.3 Validation Gate

When tracking in cluttered environment, an important issue resides in the definition of a region delimiting the space where future observations are likely to occur [6]. Such a region is called *validation region* or *gate*. Selecting a too small gate size may lead to miss the target-originated measurement, whereas selecting a too large size is computationally expensive and increases the probability of selecting false observations.

In our framework, the validation gate is defined through the use of the probability distribution $p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k})$. For linear Gaussian systems, an analytic expression of this distribution may be obtained. This leads to an ellipsoidal probability concentration region. For nonlinear models, the validation gate can be approximated by a rectangular or an ellipsoidal region, whose parameters are usually complex to define. Breidt [8] proposes to use Monte Carlo simulations in order to approximate the density $p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k})$, but this solution appears to be time consuming. For the systems we propose, it is possible to approximate efficiently this density by a Gaussian mixture. The corresponding validation gate V_k consists in an union of ellipses. Observing that:

$$\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k} \rightsquigarrow \int p(\mathbf{z}_k | \mathbf{x}_{k-1}, \mathbf{I}_{0:k}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k-1}) d\mathbf{x}_{k-1},$$

and reminding that an approximation of $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k-1})$ is given by the weighted swarm of particles $(\mathbf{x}_{k-1}^{(i)}, w_{k-1}^{(i)})$, the following approximation can be done:

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k}) \simeq \sum_i w_{k-1}^{(i)} p(\mathbf{z}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{I}_{0:k}). \quad (12)$$

Introducing expression (10) in (12) leads to an expression of $p(\mathbf{z}_k | \mathbf{z}_{1:k-1}, \mathbf{I}_{0:k})$ as a combination of $N \times M$ Gaussian distributions (N is the number of particles). As considering $N \times M$ ellipses is computationally expensive, we approximate the density by a sum of M Gaussian laws. We then finally obtain an approximation of V_k as an ellipse union $V_k = \bigcup_{m=1:M} \Psi_m = \{\epsilon_m : (\epsilon_m - \xi_{k,m})^t C_{k,m}^{-1} (\epsilon_m - \xi_{k,m}) \leq \gamma_m\}$ with the first and second moments defined as:

$$\begin{cases} \xi_{k,m} = \sum_i w_{k-1}^{(i)} H_{k,m} f_k(\mathbf{x}_{k-1}^{(i)}) \\ C_{k,m} = \sum_i w_{k-1}^{(i)} [H_{k,m} R_k H_{k,m}^t + Q_k^{(i)} + H_{k,m} f_k(\mathbf{x}_{k-1}^{(i)}) f_k^t(\mathbf{x}_{k-1}^{(i)}) H_{k,m}^t] - \\ \left(\sum_i w_{k-1}^{(i)} H_{k,m} f_k(\mathbf{x}_{k-1}^{(i)}) \right) \left(\sum_i w_{k-1}^{(i)} H_{k,m} f_k(\mathbf{x}_{k-1}^{(i)}) \right)^t. \end{cases}$$

The parameter γ_m is chosen in practice as the 99th percentile of the probability for $\mathbf{z}_{k,m}$ to be the *true* target-originated measurement.

In addition to a simple and optimal sampling process, the possibility to build a relevant approximation of a validation gate constitutes another advantage of the Gaussian models we propose. In order to demonstrate experimentally their significance, these systems have been applied to a point tracking application.

4 Application to Point Tracking

The objective of point tracking consists in reconstructing the 2D point trajectory along the image sequence. To that purpose, it is necessary to make some conservation assumptions on some information related to the feature point. These hypotheses may concern the point motion, or a photometric/geometric invariance in a neighborhood of the point.

The usual assumption of luminance pattern conservation along a trajectory has led to devise two kinds of methods. The first ones are intuitive methods based on correlation [5]. The second ones are defined as differential trackers, built on a differential formulation of a similarity criterion. In particular, the well-known Shi-Tomasi-Kanade tracker [17] belongs to this latter class.

In this paper, the proposed approach for point tracking is also built on the basis of luminance pattern consistency. In this application, each state \mathbf{x}_k represents the location of the point projection at time k , in image \mathbf{I}_k . In order to benefit from the advantages of the two class of method, we propose to combine a dynamic relying on a differential method and measurements based on a correlation criterion. The system we focus on is therefore composed of measurements and dynamic equations which both depend on $\mathbf{I}_{0:k}$. The noise covariance considered at each time is also automatically estimated on the image sequence. To properly handle such a system, the point tracker is built from the filtering framework presented in § 2.

4.1 Likelihood

At time k , we assume that \mathbf{x}_k is observable through a matching process whose goal is to provide the most similar points to \mathbf{x}_0 from images \mathbf{I}_0 and \mathbf{I}_k . The result of this process is the measurement vector \mathbf{z}_k . Each observation $z_{k,m}$ corresponds to a correlation peak. The number of correlation peaks (or components of \mathbf{z}_k) is fixed to a given number. Several matching criteria can be used to quantify the similarity between two points. The consistency assumption of a luminance pattern has simply led to consider the sum-of-squared-differences criterion.

As in [18] the correlation surface, denoted $r_k(x, y)$ and computed over the validation gate V_k , is converted into a response distribution: $\mathcal{D}_k \triangleq \exp(-c r_k(x, y))$, where c is a normalizing factor, fixed such as $\int_{V_k} \mathcal{D}_k = 1$. This distribution is assumed to represent the probability distribution associated to the matching process. The relative height of the different peaks defines the probability $p_{k,m}$ of the different measurements $z_{k,m}$. The covariance matrices $R_{k,m}$ are estimated from the response distribution on local supports centered around each observation. A Chi-Square “goodness of fit” test is realized, in order to check if this distribution is locally better approximated by a Gaussian or by a uniform law [3]. An approximation by a Gaussian distribution indicates a clear discrimination of the measurement, and $R_{k,m}$ is therefore set to the local covariance of the distribution. At the opposite, an approximation by a uniform distribution indicates an unclear peak detection on the response distribution. This may be due to an absence of correlation in presence of occlusions or noisy situations. In this case, the diagonal terms of $R_{k,m}$ are fixed to infinity, and the off-diagonal terms are set to 0. Finally, in this application, matrices $H_{k,m}$ are set to identity.

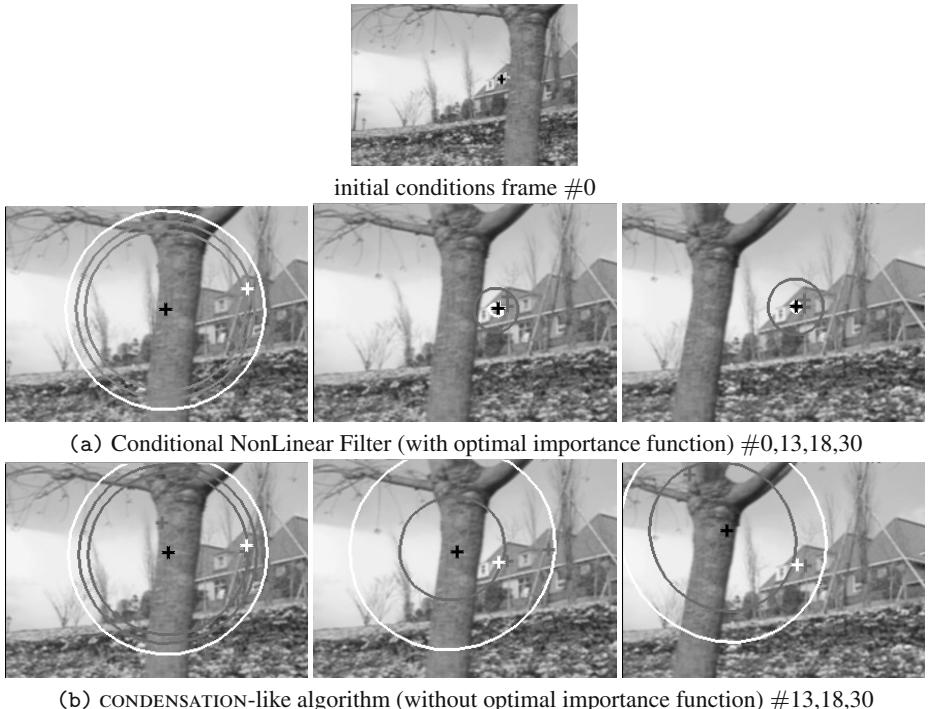
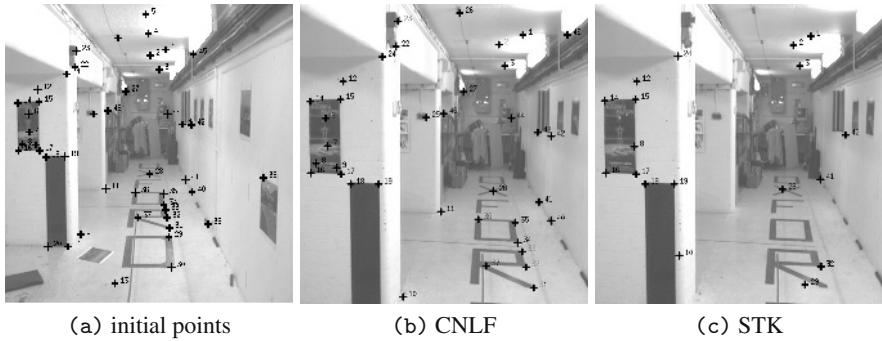


Fig. 1. Interest of the optimal interest function in case of occlusion. Tracking results obtained with (a) the Conditional NonLinear Filter, with the use of optimal importance function and (b) a CONDENSATION-like algorithm, without the use of optimal importance function. For both algorithms, the considered filtering system is the one described in §3.2. Black crosses present the estimates. White and gray crosses corresponds to the observations, and the ellipses to their associated validation gates. The white crosses show the measurement of highest probability.

4.2 Dynamic Equation

As we wish to manage situations where no *a priori* knowledge on the dynamic of the surrounding object is available, and in order to be reactive to any unpredictable change of speed and direction of the feature point, the dynamic we consider is estimated from $\mathbf{I}_{0:k}$. The state equation describes the motion of a point \mathbf{x}_{k-1} between images $k-1$ and k , and allows a prediction of \mathbf{x}_k . A robust parametric motion estimation technique [14] is used to estimate reliably a 2D parametric model representing the dominant apparent velocity field on a given support \mathcal{R} . The use of such a method on an appropriate local support around \mathbf{x}_{k-1} provides an estimate of the motion vector at the point \mathbf{x}_{k-1} from images \mathbf{I}_{k-1} and \mathbf{I}_k . As \mathcal{R} is a local domain centered at \mathbf{x}_{k-1} , the estimated parameter vector depends in a nonlinear way on \mathbf{x}_{k-1} . The noise variable \mathbf{w}_k accounts for errors related to the local motion model. It is assumed to follow a zero mean Gaussian distribution of fixed covariance .

**Fig. 2.** Corridor sequence

5 Experimental Results

In this section, we present some experimental results on four different sequences to demonstrate the efficiency of the proposed point tracker.

The first result is presented to demonstrate the interest of the optimal importance function. To that purpose, we have chosen to study an occlusion case, on the **Garden** sequence. This sequence shows a garden and a house occluded by a tree. Let us focus on a peculiar feature point located on the top of a house roof. This point is visible in the two first images and stays hidden from frame #3 to frame #15. Two algorithms have been tested for the tracking of this point. Both of them rely on the same filtering system (the one described in section § 3.2). The first one is the method we propose (namely the Conditional NonLinear Filter (CNLF), with the use of the optimal importance function), whereas the second one is a CONDENDATION-like algorithm, for which the considered importance function is identified to the diffusion process. Figure 1 presents the obtained results. The use of the optimal importance function allows us to recover the actual point location after a long occlusion. This shows clearly the benefit that can be obtained when taking into account the measurement in the diffusion process.

The second sequence, **Corridor**, constitutes a very difficult situation, since it combines large geometric deformations, high contrast, and ambiguities. The initial points and the final tracking results provided by the Shi-Tomasi-Kanade (STK) tracker, and the CNLF are presented in figure 2. In such a sequence, it can be noticed that the STK leads to good tracking results only for a small number of points. On the opposite, for the CNLF, the trajectories of all the feature points are well-recovered. Let us point out that for this sequence, considering one or several observations per point leads nearly to the same results. Another result of the CNLF, with a multimodal likelihood, is presented on the sequence **Caltra**. This sequence shows the motion of two balls, fixed on a rotating rigid circle, on a cluttered background. Compared to STK (fig.3), the CNLF succeeds in discriminating the balls from the wall-paper, and provides the exact trajectories. Such a result shows the ability of this tracker to deal with complex trajectories in a cluttered environment.

The last result on the **hand** sequence demonstrates that considering several observations improves the tracking results in case of ambiguous situations. This sequence presents finger motions of one hand. Figure 4 illustrates the results obtained with the

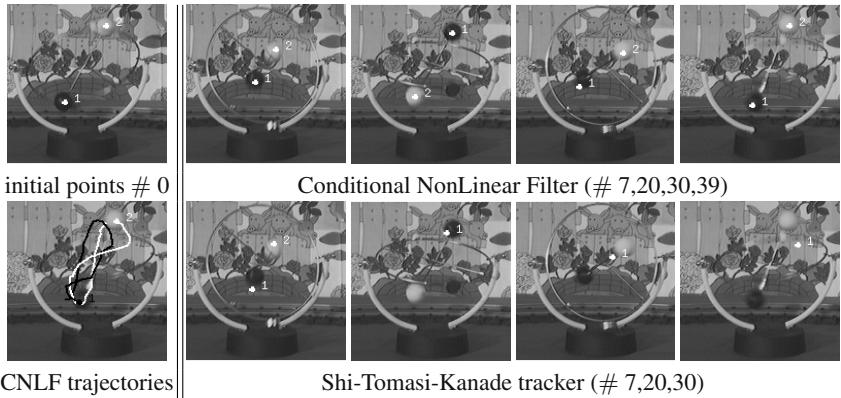


Fig. 3. Caltra sequence

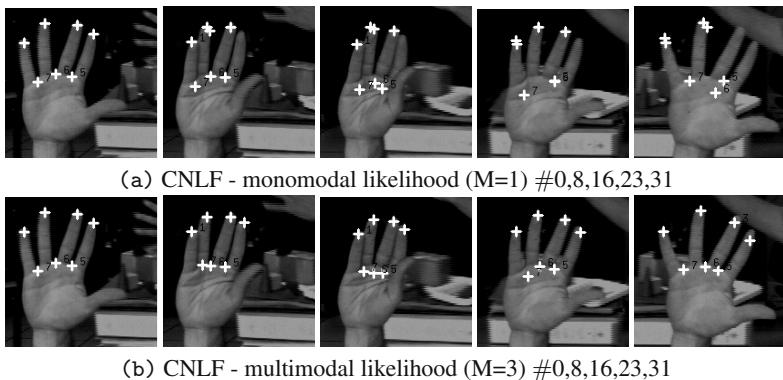


Fig. 4. Hand sequence: Conditional NonLinear Filter results. (a) Only one observation is considered per point and the involved system is the one described in §3.1; (b) 3 observations are considered per point and the involved system is the one described in §3.2

CNLF, considering a monomodal likelihood (a) and a multimodal likelihood (b). As it can be observed, considering only one correlation peak per point leads here to mistake the different fingers. This confusing situations are solved by taking into account several (here, 3) observations.

6 Conclusion

In this paper, we proposed a Conditional NonLinear Filter for point tracking in image sequences. This tracker has the particularity of dealing with *a priori*-free systems, which entirely depend on the image data. In that framework, a new filtering system has been described. To be robust to cluttered background, we have proposed a '*<*peculiar class of multimodal likelihood. Unlike usual systems used in vision applications within non linear stochastic filtering framework, we deal with system which allows an exact estimate

of the optimal importance function. The knowledge of the optimal function enables to include naturally measurements into the diffusion process and authorizes to build a relevant approximation of a validation gate. Such a framework, applied to a point tracking application, enables to significantly improve the result of traditional trackers. The resulting point tracker has been shown to be robust to occlusions and complex trajectories.

References

1. B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Englewood Cliffs, 1979.
2. E. Arnaud, E. Mémin and B. Cernuschi-Frías. A robust stochastic filter for point tracking in image sequences. *ACCV*, 2004.
3. E. Arnaud, E. Mémin and B. Cernuschi-Frías. Conditional filters for image sequence based tracking - application to point tracker. accepted for publication *IEEE trans. on Im. Proc*, 2004.
4. M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *TSP*, 50(2), 2002.
5. P. Aschwanden and W. Guggenbühl. Experimental results from a comparative study on correlation-type registration algorithms. In *W. Förstner and St. Ruwiedel, editors, Robust Computer Vision*, p. 268–289, 1992.
6. Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
7. M.J. Black and A.D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *ECCV*, p. 909–924, 1998.
8. F.J. Breidt and A.L. Carriquiry. Highest density gates for target traking. *IEEE Trans. on Aerospace and Electronic Systems*, 36(1):47–55, 2000.
9. A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
10. N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEEE Processing-F (Radar and Signal Processing)*, 140(2), 1993.
11. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.
12. A. Kong, J.S. Liu, and W.H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.
13. F. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP:IU*, 60(2):119–140, 1994.
14. J.-M. Odobez, P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journ. of Vis. Com. and Im. Repr.*, 6(4):348–365, 1995.
15. N.P. Papanikolopoulos, P.K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. *IEEE Trans. on Robotics and Automation*, 9(1):14–35, 1993.
16. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV*, p. 661–675, 2002.
17. J. Shi and C. Tomasi. Good features to track. In *CVPR*, p. 593–600, 1994.
18. A. Singh and P. Allen. Image-flow computation : An estimation-theoretic framework and a unified perspective. *CVGIP: IU*, 56(2):152–177, 1992.

Learning to Segment

Eran Borenstein and Shimon Ullman*

Faculty of Mathematics and Computer Science

Weizmann Institute of Science

Rehovot, Israel 76100

{eran.borenstein,shimon.ullman}@weizmann.ac.il

Abstract. We describe a new approach for learning to perform class-based segmentation using only unsegmented training examples. As in previous methods, we first use training images to extract fragments that contain common object parts. We then show how these parts can be segmented into their figure and ground regions in an automatic learning process. This is in contrast with previous approaches, which required complete manual segmentation of the objects in the training examples. The figure-ground learning combines top-down and bottom-up processes and proceeds in two stages, an initial approximation followed by iterative refinement. The initial approximation produces figure-ground labeling of individual image fragments using the unsegmented training images. It is based on the fact that on average, points inside the object are covered by more fragments than points outside it. The initial labeling is then improved by an iterative refinement process, which converges in up to three steps. At each step, the figure-ground labeling of individual fragments produces a segmentation of complete objects in the training images, which in turn induce a refined figure-ground labeling of the individual fragments. In this manner, we obtain a scheme that starts from unsegmented training images, learns the figure-ground labeling of image fragments, and then uses this labeling to segment novel images. Our experiments demonstrate that the learned segmentation achieves the same level of accuracy as methods using manual segmentation of training images, producing an automatic and robust top-down segmentation.

1 Introduction

The goal of figure-ground segmentation is to identify an object in the image and separate it from the background. One approach to segmentation – the *bottom-up approach* – is to first segment the image into regions and then identify the image regions that correspond to a single object. The initial segmentation mainly relies on image-based criteria, such as the grey level or texture uniformity of image regions, as well as the smoothness and continuity of bounding contours. One of the major shortcomings of the bottom-up approach is that an object may be segmented into multiple regions, some of which may incorrectly merge the object

* This research was supported in part by the Moross Laboratory at the Weizmann Institute of Science.

with its background. These shortcomings as well as evidence from human vision [1,2] suggest that different classes of objects require different rules and criteria to achieve meaningful image segmentation. A complementary approach, called *top-down segmentation*, is therefore to use prior knowledge about the object at hand such as its possible shape, color, texture and so on. The relative merits of bottom-up and top-down approaches are illustrated in Fig. 1.

A number of recent approaches have used fragments (or patches) to perform object detection and recognition [3,4,5,6]. Another recent work [7] has extended this fragment approach to segment and delineate the boundaries of objects from cluttered backgrounds. The overall scheme of this segmentation approach, including the novel learning component developed in this paper, is illustrated schematically in Fig. 2. The first stage in this scheme is fragment extraction (F.E.), which uses unsegmented class and non-class training images to extract and store image fragments. These fragments represent local structure of common object parts (such as a nose, leg, neck region etc. for the class of horses) and are used as shape primitives. This stage applies previously developed methods for extracting such fragments, including [8,4,5]. In the detection and segmentation stage a novel class image is covered by a subset of the stored fragments. A critical assumption is that the figure-ground segmentation of these covering fragments is already known, and consequently they induce figure-ground segmentation of the object. In the past, this figure-ground segmentation of the basic fragments, termed the fragment labeling stage (F.L.), was obtained manually. The focus of this paper is to extend this top-down approach by providing the capacity to learn the segmentation scheme from unsegmented training images, and avoiding the requirement for manual segmentation of the fragments.

The underlying principle of our learning process is that class images are classified according to their figure rather than background parts. While figure regions in a collection of class-image samples share common sub-parts, the background regions are generally arbitrary and highly variable. Fragments are therefore more likely to be detected on the figure region of a class image rather than in the background. We use these fragments to estimate the variability of regions within sampled class images. This estimation is in turn applied to segment the fragments themselves into their figure and background parts.

1.1 Related Work

As mentioned, segmentation methods can be divided into bottom-up and top-down schemes. Bottom-up segmentation approaches use different image-based uniformity criteria and search algorithms to find homogenous segments within the image. The approaches vary in the selected image-based similarity criteria, such as color uniformity, smoothness of bounding contours, texture etc. as well as in their implementation.

Top-down approaches that use class-based (or object-specific) criteria to achieve figure-ground segmentation include deformable templates [10], active shape models (ASM) [11] and active contours (snakes) [12]. In the work on deformable templates, the template is designed manually for each class of objects.



Fig. 1. Bottom-up and Top-down segmentation (two examples): Left – input images. Middle – state-of-the-art bottom-up segmentation ([9]). Each colored region (middle-left) represents a segment and the edge map (middle-right) represents the segments’ boundaries. Right – class-specific segmentation (white contour) as learned automatically by our system. The bottom-up approach may segment objects into multiple parts and merge background and object parts as it follows prominent image-based boundaries. The top-down approach uses stored class-specific representation to give an approximation for the object boundaries. This approximation can then be combined with bottom-up segmentation to provide an accurate and complete segmentation of the object.

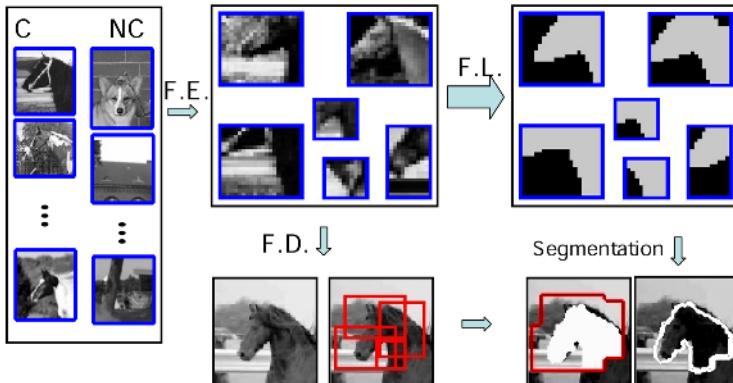


Fig. 2. The approach starts from a set of class (C) and non-class (NC) training images. The first stage is fragment extraction (F.E.) that extracts a set of informative fragments. This is followed by fragment-labeling (F.L.), the focus of this work, in which each fragment is divided into figure and background. During recognition, fragments are detected in input images (fragment detection, F.D.). The fragments’ labeling and detection are then combined to segment the input images.

In schemes using active shapes, the training data are manually segmented to produce aligned training contours. The object or class-specific information in the active contours approach is usually expressed in the initial contour and in the definition of the external force. In all of the above top-down segmentation schemes, the class learning stage requires extensive manual intervention.

In this work we describe a scheme that automatically segments shape fragments into their figure and ground relations using unsegmented training images, and then uses this information to segment class objects in novel images from their background. The system is given a set of class images and non-class images and requires only one additional bit for each image in this set (“class” / “non-class”).

2 Constructing a Fragment Set (Fragment Extraction)

The first step in the fragment-based approach is the construction of a set of fragments that represents the class and can be used to effectively recognize and segment class images. We give below a brief overview of the fragment extraction process. (further details of this and similar approaches can be found in [8,4,5].) The construction process starts by randomly collecting a large set of candidate fragments of different sizes extracted from images of a general class, such as faces, cars, etc. The second step is to select from the initial pool of fragments a smaller subset of the more useful fragments for detection and classification. These fragments are selected using an information measure criterion. The aim is that the resulting set be highly informative, so that a reliable classification decision can be made based on the detection of these fragments. Detected fragments should also be highly overlapping as well as being well-distributed across the object, so that together they are likely to cover it completely. The approach in [8] sets for each candidate fragment a detection threshold selected to maximize the mutual information between the fragment detection and the class. A fragment is subsequently detected in an image region if the similarity measure (absolute value of the normalized linear correlation in our case) between the fragment and that region exceeds the threshold. Candidates f_j are added to the fragment set F^s one by one so as to maximize the gain in mutual information $I(F^s; C)$ between the fragment set and the class:

$$f_j = \arg \max_f (I(F^s \cup f; C) - I(F^s; C)) \quad (1)$$

This selection process produces a set of fragments that are more likely to be detected in class compared with non-class images. In addition, the selected fragments are highly overlapping and well distributed. These properties are obtained by the selection method and the fragment set size: a fragment is unlikely to be added to the set if the set already contains a similar fragment since the mutual information gained by this fragment would be small. The set size is determined in such a way that the class representation is over-complete and, on average, each detected fragment overlaps with several other detected fragments (at least 3 in our implementation).

3 Learning the Fragments Figure-Ground Segmentation

To use the image fragments for segmentation, we next need to learn the figure-ground segmentation of each fragment. The learning process relies on two main criteria: *border consistency* and the *degree of cover*, which is related to the variability of the background. We initialize the process by performing a stage of bottom-up segmentation that divides the fragment into a collection of uniform regions. The goal of this segmentation is to give a good starting point for the learning process – pixels belonging to a uniform subregion are likely to have the same figure-ground labeling. This starting point is improved later (Sect. 5). A

number of bottom-up segmentation algorithms were developed in the past to identify such regions. In our implementation we use the algorithm developed by [9], which is fast (less than one second for an image with 240×180 pixels) and segments images on several scales. We used scales in which the fragments are over-segmented (on average they divide the fragments into 9 subregions) providing subregions that are likely to be highly uniform. The algorithm was found to be insensitive to this choice of scale (scales that give on average 4 – 16 sub-regions produce almost identical results). We denote the different regions of a fragment F by R_1, R_2, \dots, R_n . Each region in the fragment (R_j) defines a subset of fragment points that are likely to have the same figure-ground label.

3.1 Degree of Cover

The main stage of the learning process is to determine for each region whether it is part of the figure or background. In our fragment-based scheme, a region R_j that belongs to the figure, will be covered on average by significantly more fragments than a background region R_i , for two reasons. First, the set of extracted fragments is sufficiently large to cover the object several times (7.2 on average in our scheme). Second, the fragment selection process extracts regions that are common to multiple training examples and consequently most of the fragments come from the figure rather than from background regions. Therefore, the number of fragments detected in the image that cover a fragment's region R_j can serve to indicate whether R_j belongs to the figure (high degree of cover) or background (low degree of cover). The average degree of cover of each region over multiple images, (denoted by r_j), can therefore be used to determine its figure-ground label. The value r_j is calculated by counting the average number of fragments overlapping with the region over all the class images in the training set. The higher r_j , the higher its likelihood to be a figure region (in our scheme, an average of 7.0 for figure points compared with 2.2 for background points). The degree of cover therefore provides a powerful tool to determine the figure-ground segmentation of the fragments. Using the degree of cover r_j $j = 1, \dots, n$, for the n regions in the fragment, we select as the figure part all the regions with $r_j \geq \bar{r}$ for some selected threshold \bar{r} . That is, the figure part is defined by:

$$P(\bar{r}) = \bigcup_{\{j: r_j \geq \bar{r}\}} R_j \quad (2)$$

In this manner, all the regions contained in a chosen figure part $P(\bar{r})$ have a degree of cover higher or equal to \bar{r} , while all other regions have a degree of cover lower than \bar{r} . The segmentation of the fragment into figure and background parts is therefore determined by a single parameter, the degree of cover \bar{r} . Since $\bar{r} = r_k$ for some $k = 1, \dots, n$, the number of possible segmentations is now reduced from 2^n to n . This stage, of dividing the fragment into uniform regions and then ranking them using the degree of cover, is illustrated in Fig. 3. We next show how to choose from these options a partition that is also consistent with edges found in image patches covered by the fragment.

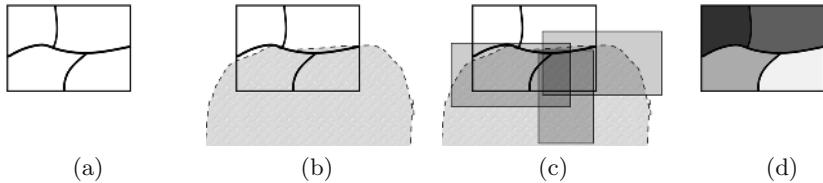


Fig. 3. Degree of cover: a fragment segmented into uniform regions (a) is detected on a given object (b). The degree of cover by overlapping fragments (also detected on the object) indicates the likelihood of a region to be a figure sub-region, indicated in (d) by the brightness of the region.

3.2 Border Consistency

The degree of cover indicates the likelihood of a fragment region to belong to the figure part. We next determine the boundary that optimally separates figure from background regions (such a boundary will exist in the fragment, unless it is an internal fragment). A fragment often contains multiple edges, and it is not evident which of these corresponds to the figure-ground boundary we are looking for. Using the training image set, we detect the fragment in different class-images. We collect the image patches where the fragment was detected, and denote this collection by H_1, H_2, \dots, H_k . Each patch in this collection, H_j , is called a *fragment hit* and $H_j(x, y)$ denotes the grey level value of pixel (x, y) in this hit. In each one of these hits we apply an edge detector. Some edges, the class-specific edges, will be consistently present among hits, while other edges are arbitrary and change from one hit to the other. We learn the fragment’s consistent edges by averaging the edges detected in these hits. Pixels residing on consistent edges will get a high average value, whereas pixels residing on noise or background edges will get a lower average, defined by:

$$D(x, y) = \frac{1}{k} \sum_{j=1}^k \text{edge}(H_j(x, y)) \quad (3)$$

Where $\text{edge}(\cdot)$ is the output of an edge detector acting on a given image. By the end of this process $D(x, y)$ is used to define the consistent edges of the fragment (see also Fig. 4).

We differentiate between three types of edges seen in this collection of hits. The first, defined here as the *border edge*, is an edge that separates the figure part of the fragment from its background part. This is the edge we are looking for. The second, defined here as an *interior edge*, is an edge within the figure part of the object. For instance, a human eye fragment may contain interior edges at the pupil or eyebrow boundaries. The last type, *noise edge*, is arbitrary and can appear anywhere in the fragment hit. It usually results from background texture or from artifacts coming from the edge detector. The first two types of edges are the consistent edges and in the next section we show how to use them to segment the fragment.

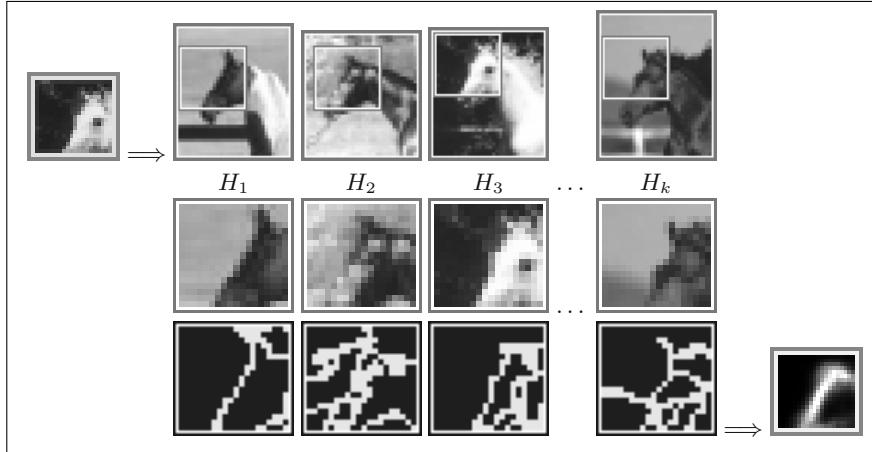


Fig. 4. Learning consistent edges. Fragment (top left) and the consistent boundary located in it (bottom right). To detect the consistent boundary, fragment hits (H_1, \dots, H_k) are extracted from a large collection of training class images where the fragment is detected (Top row shows the hit location in the images, middle row shows the hits themselves). An edge detector is used to detect the edge map of these hits (bottom row). The average of these edge maps gives the consistent edge (bottom right).

3.3 Determining the Figure-Ground Segmentation

In this section we combine the information supplied by the consistent edges computed in the last step with the degree of cover indicating the likelihood of fragment regions to be labeled as figure. The goal is to divide each fragment F , into a figure part P , and a complementary background part P^c in an optimal manner. The boundary between P and P^c will be denoted by ∂P . As mentioned, the set of consistent edges includes both the figure-ground boundary in the fragment (if such exists), as well as consistent internal boundaries within the object. Therefore, all the consistent edges should be either contained in the figure regions, or should lie along the boundary ∂P separating P from the background part P^c . A good segmentation will therefore maximize the following functional:

$$P = \arg \max_{P(\bar{r})} \left(\sum_{(x,y) \in P(\bar{r})} D(x,y) + \lambda \sum_{(x,y) \in \partial P(\bar{r})} D(x,y) \right) \quad (4)$$

The first term in this functional is maximized when the fragment's figure part contains as many as possible of the consistent edges. The second term is maximized when the boundary ∂P separating figure from ground in the fragment is supported by consistent edges. The parameter λ ($\lambda = 10$ in our implementation) controls the relative weights of the two terms.

Solving this problem is straightforward. As noted in (2), there are n possible values for \bar{r} , and each defines a possible segmentation of the fragment into a figure part $P(\bar{r})$ and background $P^c(\bar{r})$. It is therefore necessary to check which of the n

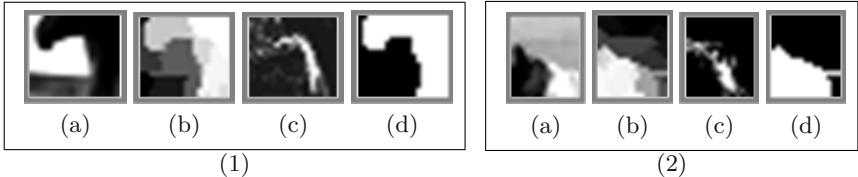


Fig. 5. Stages of fragment figure-ground segmentation (two examples). Given a fragment (a), we divide it into regions likely to have same figure-ground label. We then use the degree of cover to rank the likelihood of each region to be in the figure part of the fragment (b). Next, the fragment hits are used to determine its consistent edges (c). In the last stage, the degree of cover and the consistent edges are used to determine the figure-ground segmentation of the fragment (d).

options maximizes (4). This procedure alone produces good segmentation results, as discussed in the results section. The overall process is illustrated in Fig. 5. The figure depicts the stages of labeling two fragments that are difficult to segment. Note that by using the degree of cover and border consistency criteria it becomes possible to solve problems that are difficult to address using bottom-up criteria alone. Some parts of the contours (Fig. 5(1)) separating the figure from the background are missing in the fragment but are reconstructed by the consistent edges. Similarly, using the border consistency and degree of cover criteria, it is possible to group together dissimilar regions (eg. the black and white regions of the horse head in Fig. 5(2))

4 Image Segmentation by Covering Fragments

Once the figure-ground labels of the fragments are assigned, we can use them to segment new class images in the following manner. The detected fragments in a given image serve to classify covered pixels as belonging to either figure or background. Each detected fragment applies its figure-ground label to “vote” for the classification of all the pixels it covers. For each pixel we count the number of votes classifying it as figure versus the number of votes classifying it as background. In our implementation, the vote of each fragment had a weight $w(i)$. This value was set to the class-specificity of the fragment; namely the ratio between its detection rate and false alarms rate. The classification decision for the pixel was based on the voting result:

$$S(x, y) = \begin{cases} +1 & \text{if } \sum_i w(i)L_i(x, y) > 0 \\ -1 & \text{if } \sum_i w(i)L_i(x, y) \leq 0 \end{cases} \quad (5)$$

Where $\sum_i w(i)L_i(x, y)$ is the total votes received by pixel (x, y) , and $L_i(x, y) = +1$ when the figure-ground label of detected fragment F_i votes for pixel (x, y) to be figure, $L_i(x, y) = -1$ when it votes for the pixel to be background. $S(x, y)$ denotes the figure-ground segmentation of the image: figure pixels are characterized by $S(x, y) = +1$ and background pixels by $S(x, y) = -1$.

The segmentation obtained in this manner can be improved using an additional stage, which removes fragments that are inconsistent with the overall cover using the following procedure. We check the consistency between the figure-ground label of each fragment L_i and the classification of the corresponding pixels it covers, given by $S(x, y)$, using normalized linear correlation. Fragments with low correlation (we used 0.65 as threshold) are regarded as inconsistent and removed from the cover. In the new cover, the figure-ground labels of covering fragments will consistently classify overlapping regions. The voting procedure (5) is applied again, this time only with the consistent fragments, to determine the final figure-ground segmentation of the image. The construction of a consistent cover can thus be summarized in two stages. In the first stage, all detected fragments are used to vote for the figure or ground labeling of pixels they cover. In the second stage, inconsistent fragments that “vote” against the majority are removed from the cover and the final segmentation of the image is determined.

5 Improving the Figure-Ground Labeling of Fragments

The figure-ground labeling of individual fragments as described in Sect. 3 can be iteratively refined using the consistency of labeling between fragments. Once the labeled fragments produce consistent covers that segment complete objects in the training images, a region’s degree of cover can be estimated more accurately. This is done using the average number of times its pixels cover figure parts in the segmented training images, rather than the average number of times its pixels overlap with other detected fragments. The refined degree of cover is then used to update the fragment’s figure-ground labeling as described in Sect. 3.3, which is then used again to segment complete objects in the training images. (As the degree of cover becomes more accurate, we can also use individual pixels instead of bottom-up subregions to define the fragment labeling.) This iterative refinement improves the consistency between the figure-ground labeling of overlapping fragments since the degree of cover is determined by the segmentation of complete objects and the segmentation of complete objects is determined by the majority labeling of overlapping fragments. This iterative process was found to improve and converge to a stable state (within 3 iterations), since majority of fragment regions are already labeled correctly by the first stage (see results).

6 Results

We tested the algorithm using three types of object classes: horse heads, human faces and cars. The images were highly variable and difficult to segment, as indicated by the bottom-up segmentation results (see below). For the class of horse heads we ran three independent experiments. In each experiment, we constructed a fragment set as described in Sect. 2. The fragments were extracted from 15 images chosen randomly from a training set of 139 class images (size 32×36). The selected fragments all contained both figure and background pixels. The selection process may also produce fragments that are entirely interior to

the object, in which case the degree of cover will be high for all the figure regions. We tried two different sizes for the fragment set: in one, we used 100 fragments, which on average gave a cover area that is 7.2 times larger than the average area of an object; in the second, we used the 40 most informative fragments within each larger set of 100 fragments. These smaller sets gave a cover area that was 3.4 times the average area of an object. We initialized the figure-ground labels of the fragments using the method described in Sect. 3. We used the fragments to segment all these 139 images, as described in Sect. 4, and then used these segmentations to refine the figure-ground labels of the fragments, as described in Sect. 5. We repeated this refinement procedure until convergence, namely, when the updating of figure-ground labels stabilized. This was obtained rapidly, after only three iterations.

The fragments selected in these experiments all contained both figure and background pixels. The selection process may also produce fragments that are entirely interior to the object, in which case the degree of cover will be high for all the figure regions.

To evaluate the automatic figure-ground labeling in these experiments, we manually segmented 100 horse head images out of the 139 images, and used them as a labeling benchmark. The benchmark was used to evaluate the quality of the fragments' labeling as well as the relative contribution of the different stages in the learning process. We performed two types of tests: in the first (labeling consistency), we compared the automatic labeling with manual figure-ground labeling of individual fragments. For this comparison we evaluated the fraction of fragments' pixels labeled consistently by the learning process and by the manual labeling (derived from the manual benchmark).

In the second type of test (segmentation consistency), we compared the segmentation of complete objects as derived by the automatically labeled fragments; the manually labeled fragments; and a bottom-up segmentation. For this comparison we used the fraction of covered pixels whose labeling matched that given by the benchmark. In the case of bottom-up segmentation, segments were labeled such that their consistency with the benchmark is maximal. The output of the segmentation (given by using [9]) was chosen so that each image was segmented into a maximum of 4 regions. The average benchmark consistency rate was 92% for the case of automatically labeled fragments, 92.5% for the case of manually labeled fragments and 70% for the labeled bottom-up segments. More detailed results from these experiments are summarized in Table 1. The results indicate that the scheme is reliable and does not depend on the initial choice of fragments set. We also found that the smaller fragment sets (40th most informative within each bigger set) give somewhat better results. This indicates that the segmentation is improved by using the most informative fragments. The automatic labeling of the fragments is highly consistent with manual labeling, and its use gives segmentation results with the same level of accuracy as those obtained using fragments that are labeled manually. The results are significantly better than bottom-up segmentation algorithms.

Another type of experiment was aimed at verifying that the approach is general and that the same algorithm applies well to different classes. This was

Table 1. Results. This table summarizes the results of the first two type of tests we performed (labeling and segmentation consistency).

		Large set			Small set		
		Ex.1	Ex.2	Ex.3	Ex.1	Ex.2	Ex.3
Labeling consistency auto. vs. benchmark	Initial Labeling (sect. 3)	83%	88%	80%	86%	90%	93%
	Final Labeling (Sect. 5)	88%	91%	89%	93%	97%	95%
Segmentation consistency	fragments labeled automatically	90%	90%	92%	90%	90%	90%
	fragments labeled manually	92%	91%	94%	91%	95%	92%
	Bottom-up Segmentation	70%					

demonstrated using two additional classes: human faces and side view images of cars. For these classes we did not evaluate the results using a manual benchmark, but as can be seen in Fig. 6, our learning algorithm gives a similar level of segmentation accuracy as obtained with manually labeled fragments. Examples of the final segmentation results on the three classes are shown in Fig. 6. It is interesting to note that shadows, which appeared in almost all the training class images, were learned by the system as car parts.

The results demonstrate the relative merits of top-down and bottom-up segmentation. Using the top-down process, the objects are detected correctly as complete entities in all images, despite the high variability of the objects shape and cluttered background. Boundaries are sometimes slightly distorted and small features such as the ears may be missed. This is expected from pure top-down segmentation, especially when fragments are extracted from as few as 15 training images. In contrast, bottom-up processes can detect region boundaries with higher accuracy compared with top-down processes, but face difficulty in grouping together the relevant regions and identifying figure-ground boundaries – such as the boundaries of horse-heads, cars and human faces in our experiments.

7 Discussion and Conclusions

Our work demonstrates that it is possible to learn automatically how to segment class-specific objects, giving good results for both the figure-ground labeling of the image fragments themselves as well as the segmentation of novel class images. The approach can be successfully applied to a variety of classes. In contrast to previous class- and object-based approaches, our approach avoids the need for manual segmentation as well as minimizing the need for other forms of manual intervention. The initial input to the system is a training set of class and non-class images. These are raw unsegmented images, each having only one additional bit of information which indicates the image as class or non-class. The

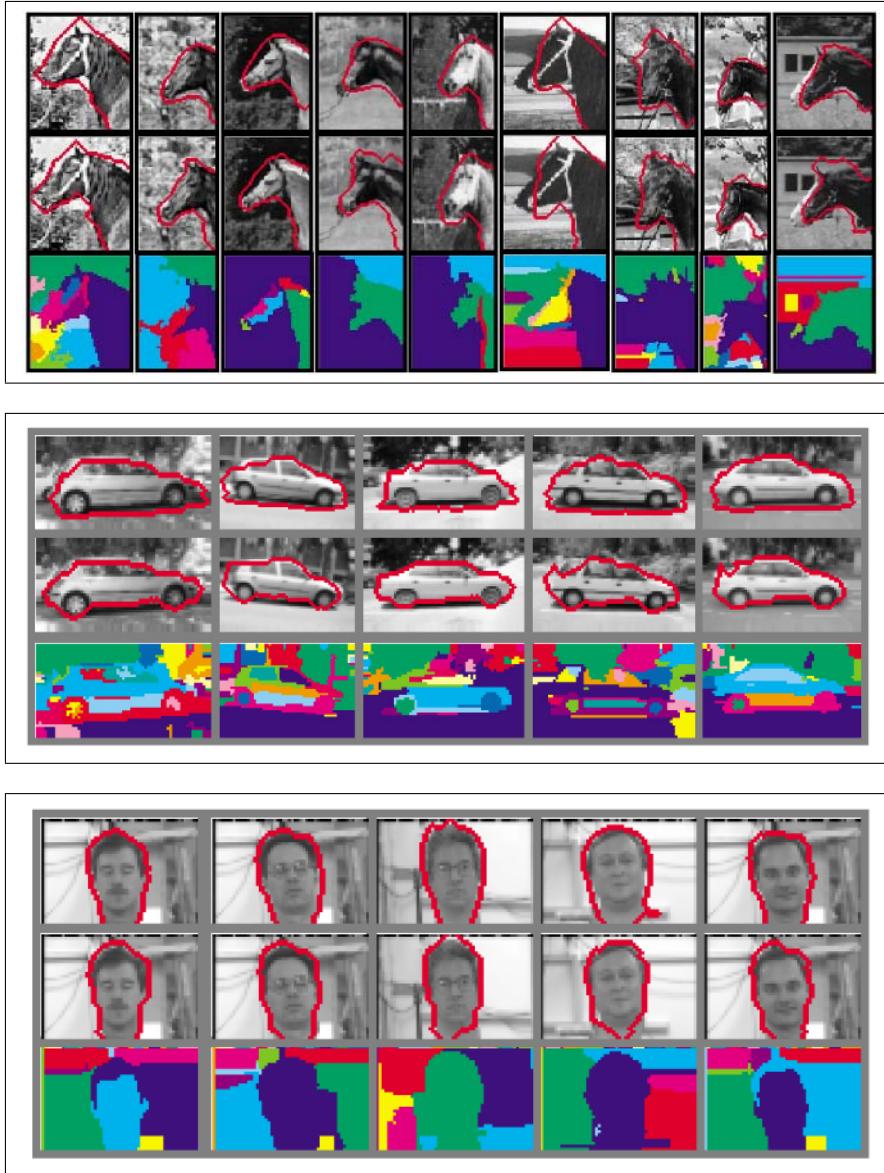


Fig. 6. Results. Rows 1-2,4-5,7-8 show figure-ground segmentation results, denoted by the red contour. The results in rows 1,4,7 are obtained using the automatic figure-ground labeling of the present method. The results in rows 2,5,8 are obtained using a manual figure-ground labeling of the fragments. Rows 3,6,9 demonstrate the difficulties faced in segmenting these images into their figure and background elements using a bottom-up approach [9]: segments are represented by different colors.

system uses this input to construct automatically an internal representation for the class that consists of image fragments representing shape primitives of the class. Each fragment is automatically segmented by our algorithm into figure and background parts. This representation can then be effectively used to segment novel class images.

The automatic labeling process relies on two main criteria: the degree of cover of fragment regions and the consistent edges within the fragments. Both rely on the high variability of background region compared with the consistency of the figure regions. We also evaluated another natural alternative criterion based on a direct measure of variability: the variability of a regions' properties (such as its grey level values) along the fragment's hit samples. Experimental evaluation showed that the degree of cover and border consistency were more reliable criteria for defining region variability – the main reason being that in some of the fragment hits, the figure part was also highly variable. This occurred in particular when the figure part was highly textured. In such cases, fragments were detected primarily based on the contour separating the figure from background region, and the figure region was about as variable as the background region. It therefore proved advantageous to use the consistency of the separating boundary rather than that of the figure part.

Another useful aspect is the use of inter-fragment consistency for iterative refinement: the figure-ground segmentation of individual fragments is used to segment images, and the complete resulting segmentation is in turn used to improve the segmentation of the individual fragments.

The figure-ground learning scheme combined bottom-up and top-down processes. The bottom-up process was used to detect homogenous fragment regions, likely to share the same figure-ground label. The top-down process was used to define the fragments and to determine for each fragment its degree of cover and consistent edges likely to separate its figure part from its background part. This combination of bottom-up and top-down processes could be further extended. In particular, in the present scheme, segmentation of the training images is based on the cover produced by the fragments. Incorporating similar bottom-up criteria at this stage as well could improve object segmentation in the training images and consequently improve the figure-ground labeling of fragments. As illustrated in Fig. 6, the top down process effectively identifies the figure region, and the bottom-up process can be used to obtain more accurate object boundaries.

Acknowledgment. The authors would like to thank Michel Vidal-Naquet for providing the fragment extraction software.

References

1. Needham, A., Baillargeon, R.: Effects of prior experience in 4.5-month-old infants' object segregation. *Infant Behaviour and Development* **21** (1998) 1–24
2. Peterson, M., Gibson, B.: Shape recognition contributions to figure-ground organization in three-dimensional displays. *Cognitive Psychology* **25** (1993) 383–429

3. Sali, E., Ullman, S.: Detecting object classes by the detection of overlapping 2-d fragments. In: BMVC, 10th. (1999) 203–213
4. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: ECCV. Volume I. (2000) 18–32
5. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: ECCV. Volume IV. (2002) 113–130
6. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. Volume II. (2003) 264–271
7. Borenstein, E., Ullman, S.: Class-specific, top-down segmentation. In: ECCV. Volume II. (2002) 109–124
8. Ullman, S., Sali, E., Vidal-Naquet, M.: A fragment based approach to object representation and classification. In: Proc. of 4th international workshop on visual form, Capri, Italy (2001) 85–100
9. Sharon, E., Brandt, A., Basri, R.: Segmentation and boundary detection using multiscale intensity measurements. In: CVPR. Volume I., Hawaii (2001) 469–476
10. Yuille, A., Hallinan, P.: Deformable templates. In: A. Blake and A. Yuille, editors, Active Vision, MIT press (1992) 21–38
11. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models — their training and application. CVIU **61** (1995) 38–59
12. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **1** (1987) 321–331

MCMC-Based Multiview Reconstruction of Piecewise Smooth Subdivision Curves with a Variable Number of Control Points

Michael Kaess, Rafal Zboinski, and Frank Dellaert

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
`{kaess,rafal,frank}@cc.gatech.edu`

Abstract. We investigate the automated reconstruction of piecewise smooth 3D curves, using subdivision curves as a simple but flexible curve representation. This representation allows tagging corners to model non-smooth features along otherwise smooth curves. We present a reversible jump Markov chain Monte Carlo approach which obtains an approximate posterior distribution over the number of control points and tags. In a Rao-Blackwellization scheme, we integrate out the control point locations, reducing the variance of the resulting sampler. We apply this general methodology to the reconstruction of piecewise smooth curves from multiple calibrated views, in which the object is segmented from the background using a Markov random field approach. Results are shown for multiple images of two pot shards as would be encountered in archaeological applications.

1 Introduction

In this paper we investigate the reconstruction of piecewise smooth 3D curves from multiple calibrated views. Among other applications, this is useful for the reconstruction of shards and other artifacts that are known to have “jagged edges”. A motivating example of a broken pot-shard is shown in Figure 1. Such objects frequently show up in large museum collections and archaeological digs, and hence solving this problem would have important implications in preserving our cultural heritage. One possible application of our work is the automatic reconstruction of archaeological artifacts [1,2]. Apart from these applied uses, the problem of representing and performing inference in the space of piecewise smooth curves is of interest in its own right, and the methods developed here have potential application for other types of objects that have both continuous and discrete parameters that need to be optimized over.

We focus on the reconstruction of curves rather than surfaces. Existing 3D surface reconstruction methods rely on automatically extracted points and/or lines [3]. In the case of textured objects or when using structured light, these methods can be used successfully to densely sample from the 3D surface of an



Fig. 1. Two automatically segmented pot-shard images. The boundaries derived from an MRF-based segmentation algorithm are shown in white.

object. However, these methods fail to use or capture the fact that an object like a shard is delineated by a closed boundary curve. In this paper we compliment traditional 3D reconstruction methods by explicitly recovering these curves.

To model piecewise smooth curves we use tagged subdivision curves as the representation. This is inspired by the work of Hoppe [4], who successfully used tagged subdivision surfaces for fitting piecewise smooth surfaces to 3D point clouds. The curve fitting literature includes use of algebraic curves [5,6], piecewise polynomials [7,8,9], point curves [10], and B-splines [11,12,13].

To the best of our knowledge, no prior work on fitting subdivision curves exists. Subdivision curves are simple to implement and provide a flexible way of representing curves of any type, including all kinds of B-splines and extending to functions without analytic representation [14]. In [4], Hoppe introduces piecewise smooth subdivision *surfaces*, allowing to model sharp features such as creases and corners by tagging the corresponding control points. We apply the tagging concept to subdivision *curves* to represent piecewise smooth curves. Earlier, we have presented this idea with some preliminary results in a workshop paper [15]. In this paper we significantly extend our approach to automatically determine the number of control points needed for a good fit. Furthermore, we replaced the manual segmentation process with an automatic MRF-based segmentation preprocessing step, obtaining a completely automated system.

To infer the parameters of these curves from the data, we propose Rao-Blackwellized sampling. In our approach, Markov chain Monte Carlo (MCMC) sampling [16] is used to obtain a posterior distribution over the discrete variables, while the continuous control point locations are integrated out after a non-linear optimization step. We also sample over the number of control points, using the framework of reversible jump (RJ) MCMC that was introduced by Green [17] and later described in a more easily accessible way as trans-dimensional MCMC [18]. In related work, Denison and Mallick [7,8] propose fitting piecewise polynomials with an unknown number of knots using RJMCMC sampling. Punskaya [9] extends this work to unknown models within each segment with applications in signal segmentation. DiMatteo [13] extends Denison's work for the special case of natural cubic B-splines, handling non-smooth curves by representing a

corner with multiple knots. However, multiple knots cannot be at the same location, and therefore only approximate the corner. With our method, corners can be represented exactly with a single control point. In addition, we are working with a much reduced sample space, as we directly solve for optimal control point locations and hence only sample over the boolean product space of corner tags.

We apply this general methodology to 3D reconstruction of piecewise smooth curves from multiple calibrated images. While much of the curve fitting literature is concerned with 1D curves for signal processing [8,7,13,9], in computer vision it is more common to fit curves in 2D or 3D. For example, 2D curves are often fit to scattered point data [6] and image contours [11]. For the special case of stereo cameras, [12] describes reconstruction and tracking of 3D curves represented by 2D B-spline curves that are either coupled through epipolar geometry constraints or are coupled to a canonical frame model through affine transformations. More general multiple view curve reconstruction methods are described in [5] using algebraic curves and in [10] for point curves with uncalibrated cameras.

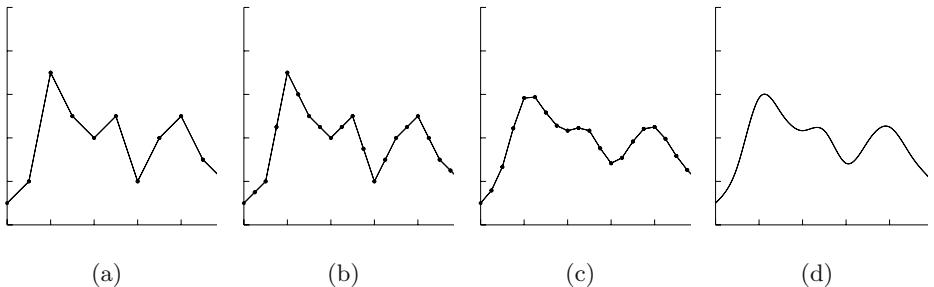


Fig. 2. The subdivision process: (a) initial control points, linearly interpolated; (b) the original mesh, subdivided by introducing average points; (c) the result after application of the averaging mask to the subdivided control points; (d) converged curve.

2 Subdivision Curves

Here we briefly review subdivision curves. See [14] for more details. A subdivision curve is defined by repeatedly refining a vector of control points $\Theta_t = (x_0^t, x_1^t, \dots, x_{n2^t-1}^t)$, where n is the initial number of control points and t the number of subdivisions performed. This refinement process can be separated into two steps as shown in Figure 2: a *splitting step* that introduces midpoints:

$$x_{2i}^{t+1} \triangleq x_i^t \quad x_{2i+1}^{t+1} \triangleq \frac{1}{2} (x_i^t + x_{i+1}^t)$$

and an *averaging step* that computes weighted averages:

$$x_i^{t+1} \triangleq \sum_k r_k x_{i+k}^t$$

The type of the resulting curve depends on the *averaging mask* r . Subdivision can be used to create a wide range of functions [14], including uniform and

non-uniform B-splines, and functions that have no analytic representation like Daubechies wavelets. For example, the mask for a cubic B-spline is $\frac{1}{4}(1, 2, 1)$.

As explained in [14], the splitting and averaging steps can be combined into multiplication of the local control points with a *local subdivision matrix* L , e.g.

$$L = \frac{1}{8} \begin{pmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{pmatrix}$$

for cubic B-splines. Repeated application of this matrix to a control point and its immediate neighbors results in a sequence of increasingly refined points that converges to the limit value of the center point. Eigenvector analysis on the matrix L leads to an *evaluation mask* u that can be applied to a control point and its neighbors, resulting in the limit position for that control point. For example, the evaluation mask for cubic B-splines is

$$u = \frac{1}{6} (1, 4, 1)$$

The curve can be refined to the desired resolution before this mask is applied.

It is convenient for the exposition below to view the *entire* subdivision process as a large matrix multiplication

$$C = S\Theta \tag{1}$$

where C is the final curve, the $n \times 1$ vector Θ represents the control points/polygon, and the *subdivision matrix* S combines all m subdivision steps and the application of the evaluation mask into one $n2^m \times n$ matrix. This can be done as both subdivision and evaluation steps are linear operations on the control points. The final curve C is a $n2^m \times 1$ vector that is obtained by multiplying S with the control point vector Θ as in (1).

The *derivative* of the final curve C with respect to a change in the control points Θ , needed below to optimize over them, is simply a constant matrix:

$$\frac{\partial C}{\partial \Theta} = \frac{\partial(S\Theta)}{\partial \Theta} = S \tag{2}$$

While the above holds for 1D functions only, an n -dimensional subdivision curve can easily be defined by using n -dimensional control points, effectively representing each coordinate by a 1D subdivision curve. Our implementation is done in the functional language ML and uses functors to remain independent of the dimensionality of the underlying space. Note that in this case, the derivative equation (2) holds for each dimension separately.

3 Piecewise Smooth Subdivision Curves

There are a number of ways to represent sharp corners in otherwise smooth curves. One solution is to place multiple control points at the location of a

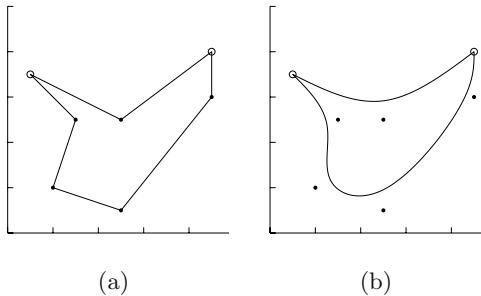


Fig. 3. A tagged 2D subdivision curve. Tagged points are drawn as circles. (a) the original control points, (b) the converged curve with non-smoothly interpolated points.

corner, but consecutive subdivision creates superfluous points at this location. Furthermore, it places unwanted constraints on the adjacent curve segments. A commonly used method in connection with B-splines is extra knot insertion.

We employ a more general method here based on work of Hoppe [4] for subdivision surfaces. It allows us to “tag” control points, allowing different averaging masks to be used at these points. E.g., using the mask $(0, 1, 0)$ forces the interpolation of the control point and at this point introduces a discontinuity in the derivative, while retaining the smooth curve properties for all other points of the curve (Figure 3). The number of tagged control points does not increase during the subdivision process, and so the non-smoothness of the curve is restricted to the tagged control points, which will always be interpolated.

Below we will use the following notation to describe tagged 3D subdivision curves. The locations of the 3D control points are given by $\Theta \triangleq \{x_0^0, x_1^0, \dots, x_{n-1}^0\}$, where n is the number of control points. For each original control point x_i^0 , a boolean *tag* b_i indicates whether it is non-smoothly interpolated, i.e. whether there is a “corner” at control point i . The collection of tags b_i is written as $T \triangleq \{b_0, b_1, \dots, b_{n-1}\}$.

4 Rao-Blackwellized Curve Fitting

In this section we describe how the parameters of a tagged subdivision curve can be estimated from noisy measurements Z , irrespective of how those measurements were obtained. In Section 5 this will be specialized to the problem of fitting from multiple, calibrated segmentations of an object.

Because the measurements are noisy, we take a probabilistic approach. Of interest is the posterior distribution

$$P(n, \Theta_n, T_n | Z) \propto P(Z | n, \Theta_n, T_n) P(\Theta_n | n, T_n) P(T_n | n) P(n) \quad (3)$$

over the possible number of control points n , the control point values Θ_n and the tag variables T_n . Here the control points $\Theta_n \in \mathbb{R}^{3n}$ are continuous and the tags $T_n \in \{0, 1\}^n$ are discrete. The *likelihood* $P(Z | n, \Theta_n, T_n)$ and *control polygon prior*

$P(\Theta_n|n, T_n)$ are application specific and will be specified in Section 5. Choosing the *complexity prior* $P(n)$ to be uniform over a range of valid values allows us to find an unbiased distribution over the number of control points. Similarly, we use an uninformative *tagging prior* $P(T_n|n)$, but a binomial distribution $P(T_n|n) \propto p^c(1-p)^{n-c}$ over the number of active tags c could also be used.

4.1 Trans-dimensional MCMC

Since the number of possible tag configurations is 2^n and hence exponential in n , we propose to use reversible jump Markov chain Monte Carlo (MCMC) sampling [18,17] to perform approximate inference. MCMC methods produce an approximate sample from a target distribution $\pi(X)$, by simulating a Markov chain whose equilibrium distribution is $\pi(X)$. The algorithm starts from a random initial state $X^{(0)}$ and proposes probabilistically generated moves in the state space, which is equivalent to running a Markov chain. The specific MCMC algorithm we use is the trans-dimensional MCMC algorithm from [18]:

1. Start with a random initial state $X^{(0)}$.
2. Propose a move type $m \in M$ with probability $j(m)$.
3. Generate a random sample u from the move-specific proposal density g_m . The move type m and random sample u determine how to move from the current state $X^{(r)}$ to the proposed state X' .
4. Calculate the corresponding reverse move (m', u') .
5. Compute the acceptance ratio

$$a = \frac{\pi(X')}{\pi(X^{(r)})} \frac{j(m')}{j(m)} \frac{g_{m'}(u')}{g_m(u)} \left| \frac{\partial(x', u')}{\partial(x, u)} \right| \quad (4)$$

where the Jacobian factor corrects for the change in variables (see below).

6. Accept $X^{(r+1)} \leftarrow X'$ with probability $\min(a, 1)$, otherwise $X^{(r+1)} \leftarrow X^{(r)}$.

The generated sequence of states $\{X^{(r)}\}$ will be a sample from $\pi(X)$ if the sampler is run sufficiently long, and one discards the samples in the initial “burn-in” period of the sampler to avoid dependence on the chosen start state.

For fitting tagged subdivision curves, one possible set of move types consists of “Up” and “Down” for inserting and deleting control points and “Modify” for flipping one or more of the tags. Care has to be taken to ensure that the move from (x, u) to (x', u') is reversible and therefore a diffeomorphism. One requirement is that the dimensions on both sides have to match. Note that in our case the Jacobian of the diffeomorphism $\left| \frac{\partial(x', u')}{\partial(x, u)} \right|$ is always 1 because we integrate out the continuous part of the space (see below).

4.2 Rao-Blackwellization

Sampling over the joint discrete-continuous space is expensive. A crucial element of our approach is to not sample from the joint posterior (3) but rather from the marginal distribution $P(n, T_n|Z)$ over the number of points n and the tags T :

$$\pi(X) \triangleq P(n, T_n | Z) = \int P(n, \Theta_n, T_n | Z) d\Theta_n \quad (5)$$

while performing the integration (5) above analytically.

From a sample over n and T of size N we can obtain a high quality approximation to the joint posterior as follows

$$P(n, \Theta_n, T_n | Z) \approx \sum_{r=1}^N P(\Theta_n | Z, (n, T_n)^{(r)}) \delta((n, T_n), (n, T_n)^{(r)}) \quad (6)$$

with $\delta(., .)$ being the Kronecker delta. Thus, (6) approximates the joint posterior $P(n, \Theta_n, T_n | Z)$ as a combination of discrete samples and continuous conditional densities $P(\Theta_n | Z, (n, T_n)^{(r)})$.

Integrating out the continuous part of the state space reduces the number of samples needed. The superiority of (6) over a density estimate based on joint samples is rooted in an application of the Rao-Blackwell theorem, which is why this technique is often referred to as *Rao-Blackwellization* [19,20,21]. Intuitively, the variance of (6) is lower because it uses exact conditional densities to approximate the continuous part of the state. As such, far fewer samples are needed to obtain a density estimate of similar quality.

Substituting the factorization of the posterior (3) in (5) we obtain

$$P(n, T_n | Z) = k P(n) P(T_n | n) \int P(Z | n, \Theta_n, T_n) P(\Theta_n | n, T_n) d\Theta_n$$

Assuming the conditional posterior $P(Z | n, \Theta_n, T_n) P(\Theta_n | n, T_n)$ is approximately normally distributed around the MAP estimate of the control points Θ_n^*

$$P(\Theta_n | Z, n, T_n) \approx \frac{1}{\sqrt{|2\pi\Sigma|}} e^{-\frac{1}{2}\|\Theta_n - \Theta_n^*\|_\Sigma^2}$$

the integral can be approximated via Laplace's method, and we obtain the following target distribution over the number of control points n and the tags T_n :

$$P(n, T_n | Z) = k P(n) P(T_n | n) \sqrt{|2\pi\Sigma|} P(Z | n, \Theta_n^*, T_n) P(\Theta_n^* | n, T_n)$$

The MAP estimate Θ_n^* can be found by non-linear optimization (see below).

5 Multiple View Fitting

In this section we specialize the general methodology of Section 4 to reconstructing tagged 3D subdivision curves from multiple 2D views of a “jagged” object. We assume here that (a) the images are calibrated, and (b) the measurements Z are the object boundaries in the 2D images, i.e. the object has been segmented out from the background. For the results presented below, the object boundaries are segmented automatically using a Markov random field approach.

To calculate an objective function, the 3D curve given by Θ_n and T_n is subdivided m times to the desired resolution, evaluated, and the resulting points $\{p_i\}_1^{n2^m}$ are projected into each view. We then use the following form for the likelihood:

$$P(Z|n, \Theta_n, T_n) \propto e^{-\frac{1}{2\sigma^2} E(Z, n, \Theta_n, T_n)}$$

where σ^2 is a variance of the noise in the measurements. The error function E above is obtained as a sum of squared errors

$$E(Z, n, \Theta_n, T_n) = \sum_{c=1}^C \sum_{i=1}^{n2^m} \Delta(\Pi_c(p_i), Z_c)^2 \quad (7)$$

with one term for each of the $n2^m$ final subdivision curve points in each of the C images, explained in more detail below. The prior $P(n)$ on the number of control points, the prior $P(T)$ on the tag configuration and the conditional control polygon prior are taken to be uniform in all the results reported below.

Each error term $\Delta(\Pi_c(p_i), Z_c)$ in (7) determines the distance from the projection $\Pi_c(p_i)$ of a point p_i into view c , to the nearest point on the object boundary Z_c . In order to speed up this common calculation, we pre-calculate a lookup table for Δ by means of the well known distance transform to obtain a Chamfer image [22]. Each pixel in a Chamfer image contains the distance from this pixel to the nearest point on the segmented curve Z_c in view c . Calculating the Chamfer images has to be done only once and runs in linear time. In this way, we trade memory usage for computational speed.

The outlines of the shards were automatically segmented as foreground and background classes using a Markov random field (MRF) approach. To offset the analytical intractability associated with MRFs, we employed Gibbs sampling to approximate the posterior probability $P(Z|I)$ of the outlines Z given the input images I . The sampling is initialized by selecting the class with highest likelihood for each pixel. Consequently, Gibbs sampling requires only a few samples to achieve accurate segmentations.

The reprojection of the 3D curve in the images is done in the standard way [3]. We assume the cameras are described using a single radial distortion parameter κ , focal lengths f_x and f_y , principal point (p_x, p_y) , and skew s . The pose of a camera is given by a rotation R and translation t . The projection of a 3D point X into an image is then given by

$$\Pi(X) = \mathcal{D}(K[R|t], \kappa, X)$$

where K is the 3×3 calibration matrix

$$K = \begin{pmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{pmatrix}$$

and D is a function that models the radial distortion.

To minimize (7) given a specific tag configuration T , we use Levenberg-Marquardt non-linear optimization. To obtain the derivative $\frac{\partial E}{\partial \Theta_0}$ of the error function E with respect to the original control points Θ_0 , we apply the chain rule to combine the derivatives of the camera projections and the derivative S from equation 2 on page 332 of the subdivision curve with respect to the original control points. Implementing the chain rule involves a pointwise multiplication of the projected curve derivatives with the Chamfer image gradients, which are estimated by convolution with a Gaussian derivative mask.

6 Results

We illustrate our approach on two sets of pot-shard images, shown in Figure 4. The shards were placed on a calibration pattern, which allowed us to easily optimize for the camera calibration parameters. Six images for the first pot shard and four images for the second were used for curve fitting, all of which are taken from about the same angle from different sides of the shard. Two of those views are shown in columns (a) and (b) of Figure 4. The images in column (c) were not used for fitting and serve as verification views. They are informative, since they are taken at a much lower angle than all other images. The shard images were automatically segmented by an MRF approach, as described in Section 5. Figure 1 shows two such segmentations overlayed on the corresponding shard images.

The fitting process starts with a circular configuration of a small number of control points in a plane parallel to the ground plane, as shown in the first row of Figure 4. Each proposal for the Markov Chain consists of a random move as described in Section 4.1, followed by non-linear optimization of the control point locations. The move can either introduce a new control point, delete an existing one, or change the tag configuration by inverting each tag with probability $\frac{1}{n}$. For evaluation of the error function, the curve is subdivided four times before the limit positions of the points are calculated using the evaluation mask.

After only five iterations, as shown in the second row of Figure 4, the subdivision curve adapted pretty well to the boundary, but the number of control points is still too small and the tagging is wrong. After 250 iterations the number of control points and the tagging configuration both adapted to fit the shard boundary well. Note that we *sample* from the posterior distribution, and the number of control points and the tagging configuration never converge to a single value.

The algorithm finds a suitable number of control points independent of the initial number, as can be seen in Figure 5(a), where this number is plotted over time. Independent of the initial number of control points, the algorithm converges quickly to a “optimal” number that is influenced by the curve itself and the variance of the measurements.

The diagram in Figure 5(a) shows nicely the burn-in phase of the Markov chain. In what follows, the posterior distribution is determined by throwing away the first 500 of the 1500 samples to make the result independent of the initialization. 1500 samples seem sufficient, an evaluation of 9000 samples did not show a significant change in results.



Fig. 4. Results are shown for two pot shards. Projections of the control points are drawn as yellow '+' for untagged and yellow 'x' for tagged ones, and the corresponding subdivision curve is drawn in white. Six views are used for the fitting process of the first shard, while only four are used for the second shard. In both cases, two of those are shown in columns (a) and (b). The third column (c) shows a view that is not used for fitting and is taken from a lower angle than all other images.

The first row shows the projections of the initial control points and the corresponding 3D subdivision curve on images of the first shard. The second row shows results after five iterations (error: $9.8 \cdot 10^2$). The third row is sample number 250 (error: $4.3 \cdot 10^2$). The last row is the result for the second shard after 250 samples.

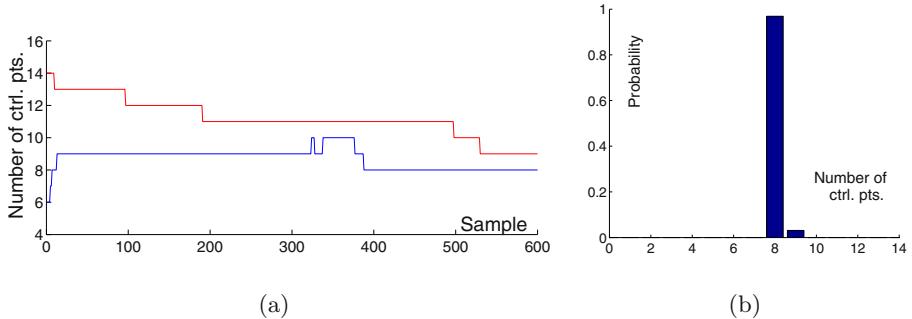


Fig. 5. (a) Number of control points during the burn-in phase. The lower curve starts with six control points, the upper one with 14. (b) Probability over different numbers of control points after burn-in phase.

One example of a marginal distribution that can be approximated from these samples is the number of control points that are needed to fit the curve well, as shown in Figure 5(b).

7 Conclusion

We investigated modeling piecewise smooth curves with tagged subdivision curves that provide a simple and flexible representation. The parameters of these curves were determined by a Rao-Blackwellized sampler, in which the optimal locations of the control points were integrated out and determined by non-linear optimization, and only the distribution over the number of control points and the tag configurations was sampled over.

This method was successfully applied to 3D curve reconstruction from multiple images, as illustrated in the results section. These results were obtained in automated fashion, using an MRF-based segmentation approach to automatically segment the images with a known calibration background. Then, starting from an initial circular distribution of the control points, the algorithm approximated the object boundary well within a small number of sampling steps.

It would be of interest to more closely examine the quality of the Gaussian assumption made in the Rao-Blackwellization step. One way to validate this assumption is by MCMC sampling over the control point locations for a given tag configuration. Also, on the image processing side, there are some problems with double shard boundaries and occluding contours that need to be resolved in order to create a robust, automated system.

We would like to connect our 3D curve reconstruction methods with a complete “broken-pot” reconstruction such as described in [2]. Comparing features of boundaries from different shards could be used for reconstruction of archaeological artifacts, possibly in connection with other features, like texture and surface curvature, as suggested in [1]. Finally, it is our hope that the general method-

ology described here can be successfully applied in other discrete-continuous reconstruction settings.

Acknowledgments. We would like to thank Peter Presti from IMTC for providing the shard images and Zia Khan for valuable discussions on reversible jump MCMC.

References

1. Kanaya, I., Chen, Q., Kanemoto, Y., Chihara, K.: Three-dimensional modeling for virtual relic restoration. In: MultiMedia IEEE. Volume 7. (2000) 42–44
2. Cooper, D., Willis, A., Andrews, S., Baker, J., Cao, Y., Han, D., Kang, K., Kong, W., Leymarie, F., Orriols, X., Velipasalar, S., Vote, E., Joukowsky, M., Kimia, B., Laidlaw, D., Mumford, D.: Bayesian virtual pot-assemble from fragments as problems in perceptual-grouping and geometric-learning. In: Intl. Conf. on Pattern Recognition (ICPR), IEEE Computer Society Press (2002) 11–15
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
4. Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W.: Piecewise smooth surface reconstruction. Computer Graphics **28** (1994) 295–302
5. Kaminski, J., Fryers, M., Shashua, A., Teicher, M.: Multiple view geometry of non-planar algebraic curves. In: ICCV. Volume 2. (2001) 181–186
6. Bajaj, C., Xu, G.: Data fitting with cubic A-splines. In: Comp. Graph. Int. (1994)
7. Denison, D.G.T., Mallick, B.K., Smith, A.F.M.: Automatic bayesian curve fitting. Journal of the Royal Statistical Society, Series B **60** (1998) 333–350
8. Mallick, B.K.: Bayesian curve estimation by polynomials of random order. J. Statist. Plan. Inform. **70** (1997) 91–109
9. Punsikaya, E., Andrieu, C., Doucet, A., Fitzgerald, W.: Bayesian curve fitting using MCMC with applications to signal segmentation. IEEE Transactions on Signal Processing **50** (2002) 747–758
10. Berthilsson, R., Åström, K., Heyden, A.: Reconstruction of curves in R^3 , using factorization and bundle adjustment. In: ICCV. Volume 1. (1999) 674–679
11. Cham, T.J., Cipolla, R.: Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies. PAMI **21** (1999) 49–53
12. Cham, T.J., Cipolla, R.: Stereo coupled active contours. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). (1997) 1094–1099
13. DiMatteo, I., Genovese, C.R., Kass, R.E.: Bayesian curve fitting with free-knot splines. In: Biometrika. (2001) 1055–1071
14. Stollnitz, E., DeRose, T., Salesin, D.: Wavelets for computer graphics: theory and applications. Morgan Kaufmann (1996)
15. Kaess, M., Dellaert, F.: Reconstruction of objects with jagged edges through rao-blackwellized fitting of piecewise smooth subdivision curves. In: Workshop on Higher Level Knowledge in Computer Vision at ICCV. (2003)
16. Gilks, W., Richardson, S., Spiegelhalter, D., eds.: Markov chain Monte Carlo in practice. Chapman and Hall (1996)
17. Green, P.: Reversible jump Markov chain Monte Carlo computation and bayesian model determination. Biometrika **82** (1995) 711–732

18. Green, P.: Trans-dimensional Markov chain Monte Carlo. Chapter in Highly Structured Stochastic Systems (2003)
19. Gelfand, A., Smith, A.: Sampling-based approaches to calculating marginal densities. *J. Am. Statistical Association* **85** (1990) 398–409
20. Casella, G., Robert, C.: Rao-Blackwellisation of sampling schemes. *Biometrika* **83** (1996) 81–94
21. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer (1999)
22. Thiel, E., Montanvert, A.: Chamfer masks: Discrete distance functions, geometrical properties and optimization. In: Pattern Recognition. (1992) 244–247

Bayesian Correction of Image Intensity with Spatial Consideration*

Jiaya Jia¹, Jian Sun², Chi-Keung Tang¹, and Heung-Yeung Shum²

¹ Computer Science Department,
Hong Kong University of Science and Technology,
`{leojia, cktang}@cs.ust.hk`
² Microsoft Research Asia,
`{t-jiansu, hshum}@microsoft.com`

Abstract. Under dimly lit condition, it is difficult to take a satisfactory image in long exposure time with a hand-held camera. Despite the use of a tripod, moving objects in the scene still generate ghosting and blurring effect. In this paper, we propose a novel approach to recover a high-quality image by exploiting the tradeoff between exposure time and motion blur, which considers color statistics and spatial constraints simultaneously, by using only two defective input images. A Bayesian framework is adopted to incorporate the factors to generate an optimal color mapping function. No estimation of PSF is performed. Our new approach can be readily extended to handle high contrast scenes to reveal fine details in saturated or highlight regions. An image acquisition system deploying off-the-shelf digital cameras and camera control softwares was built. We present our results on a variety of defective images: global and local motion blur due to camera shake or object movement, and saturation due to high contrast scenes.

1 Introduction

Taking satisfactory photos under weak lighting conditions using a hand-held camera is very difficult. In this paper, we propose a two-image approach to address the image recovery problem by performing intensity correction. In order to exploit the tradeoff between the exposure time and the blurring degree of the captured images, we take the two input images using the same camera with the following exposure settings:

- One image I_L is taken with exposure time around the safe shutter speed¹, producing an under-exposed image where motion blur is largely reduced. Since it is too dark, the colors in the image are not acceptable (Fig. 1(a)).

* This work is supported by the Research Grant Council of Hong Kong Special Administration Region, China: HKUST6193/02E.

¹ In photography, the safe shutter speed is assumed to be not slower than the reciprocal of the focal length of the lens, in the unit of seconds [1]. The longer the exposure time, the blurrier the image becomes.

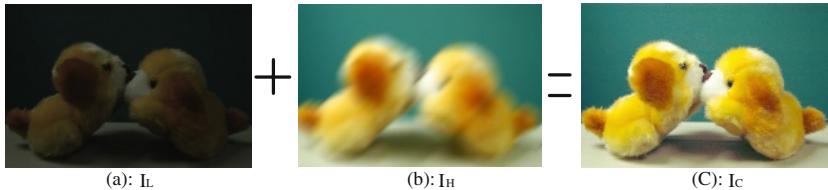


Fig. 1. We take two successive images with different exposure intervals to construct the high-quality image.

- The other image I_H is a normal image acquired under an extended exposure time. The color and brightness of this image is acceptable. However, it is motion blurred because of camera shaking or moving objects in the scene (Fig. 1(b)).

The images can be taken by a hand-held camera, and possibly in a dimly lit condition. Combining these two defective images I_L and I_H , our method automatically generates a clear and crisp image I_C , as shown in Fig. 1(c).

There are several related techniques to recover images from camera when exposure time is above the safe shutter speed. They can be roughly classified into *in-process* and *post-process* approaches, which eliminate motion blur due to long exposure and camera shake. In-process approaches are mainly hardware-based techniques, where lens stabilization is achieved by camera shake compensation [8,9]. Alternatively, CMOS cameras can perform high-speed frame captures within normal exposure time, which allows for multiple image-based motion blur restoration [11]. These methods are able to produce clear and crisp images, given a reasonable exposure time. However, they require specially designed hardware devices.

On the other hand, post-process methods are mostly motion deblurring techniques. Among them, blind deconvolution is widely adopted to enhance a single blurred image, under different assumptions on the PSF [6,15,10,14]. Alternatively, several images with different blurring directions [12] or an image sequence [2] is used, in more general situations, to estimate the PSF. In both cases, due to the discretization and quantization of images in both spatial and temporal coordinates, the PSF can not be reliably estimated, which produced a result inferior to the ground truth image if available (which is an image either taken with a camera on a tripod, or of a static scene).

Ezra and Nayar [3] proposed a hybrid imaging system consisting of a primary (high spatial resolution) detector and a secondary (high temporal resolution) detector. The secondary detector provides more accurate motion information to estimate the PSF, thus making deblurring possible even under long exposure. However, the method needs additional hardware support, and the deblurred image can still be distinguishable from the ground truth image.

Because of the weakness of the deblurring methods, we do not directly perform deblurring on I_H . Instead, an image color correction approach is adopted. By incorporating the color statistics and the spatial structures of I_H and I_L , we

propose a Bayesian framework, and maximize the *a posterior* (MAP) of the color mapping function $f(\cdot)$ from I_L to I_H in the color space so that the under-exposed I_L is enhanced to a normally exposed image I_C .

Our method can deal with camera shake and object movement at the same time, and in an unified framework. Moreover, change of object topology or object deformation can also be naturally handled, which is difficult for most deblurring methods, since different parts of the object have different PSFs. Besides, by slightly modifying one constraint, our method can be extended to deal with high contrast scenes, and automatically produce images which capture fine details in highlight or saturated area.

The rest of this paper is organized as follows: we describe our image acquisition system in Section 2. Section 3 defines the relationship between I_L and I_H . In Section 4, we state and define our problem, propose our probabilistic model, and infer the color mapping function in the Bayesian framework. Section 5 presents our results. Finally, we conclude our paper in Section 6.

2 Image Acquisition

To correctly relate two images, we require that I_L be taken almost immediately after I_H is taken. This is to minimize the difference between the two images and to maximize the regional match of the positions of each pixel if the time lapse is kept as short as possible, as illustrated in Fig. 2(a). In other words, the under-exposed image I_L can be regarded as a sensing component in the normally exposed image I_H in the temporal coordinates. This requirement makes it possible to reasonably model the camera movement during the exposure time, and constrain the mapping process.

Our image acquisition system and its configuration is in Fig. 2(b). The digital camera is connected to the computer. The two successive exposures with different shutter speeds are controlled by the corresponding camera software. This setup

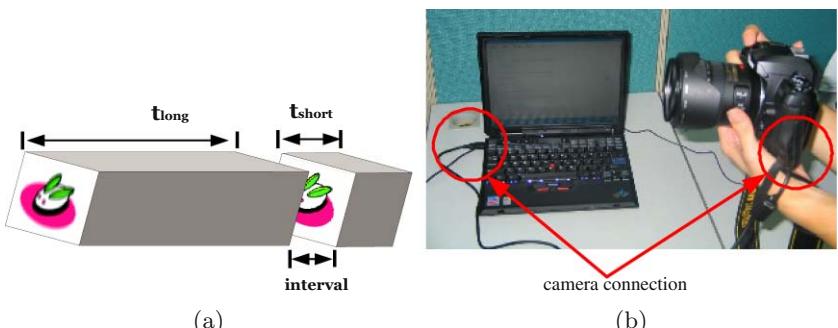


Fig. 2. (a) Two successive exposures guarantee that the center of the images do not vary by too much. (b) The configuration of our camera system.

frees the photographer from manually changing the camera parameters between shots. So that s/he can focus on shooting the best pictures.

A similar functionality, called *Exposure Bracketing*, has already been built in many digital cameras, e.g., Canon G-model and some Nikon Coolpix model digital cameras. With one shutter pressing, two or three successive images are taken with different shutter speeds under the same configurations. However, using the built-in camera functionality has some limitations: it does not operate in manual mode, and the difference of shutter speeds is limited.

In the next section, we analyze the relationship between I_L and I_H , and propose the constraints that relate these two images.

3 Relationship between I_L and I_H

I_L and I_H are two images of the same scene taken successively with different exposures. Therefore, they are related not only by the color statistics, but also by the corresponding spatial coherence. In this section, we describe their relationship, which are translated into constraints for inferring a color mapping function in our Bayesian framework, which will be described in the next section.

3.1 Color Statistics

In RGB color space, important color statistics can often be revealed through the shape of a color histogram. Thus, the histogram can be used to establish explicate connection between I_H and I_L . Moreover, since high irradiance always generates brighter pixels [7], the color statistics in I_L and I_H can be matched in order from lower to higher in pixel intensity values. Accordingly, we want to reshape the histogram of I_L , say, h_{I_L} , such that:

$$g(h_{I_L}) \doteq h_{I_H} \quad (1)$$

where $g(\cdot)$ is the transformation function performed on each color value in histogram, and h_{I_H} is the histogram of I_H . A common method to estimate $g(\cdot)$ is *adaptive histogram equalization*, which normally modifies the dynamic range and contrasts of a image according to a destination curve.

Unfortunately, this histogram equalization does not produce satisfactory results. The quantized 256 (single byte accuracy) colors in each channel are not sufficient to accurately model the variety of histogram shapes. Hence, we adopt the following method to optimally estimate the transformation function:

1. Convert the image from RGB space to a perception-based color space $l\alpha\beta$ [4], where the l is the achromatic channel and α and β contain the chromaticity value. In this way, the image is transformed to a more discrete space with known phosphor chromaticity.
2. Accordingly, we cluster the color distributions in the new color space into 65536 (double byte precision) bins, and perform histogram equalization.
3. Finally, we transform the result back to the RGB space.

By performing this transformed *histogram equalization*, we relate the two images entirely in their color space.

3.2 Color Statistics in High Contrast Scene

In situations that the images are taken in a high contrast scene, bright regions will become saturated in I_H . Histogram equalization can not faithfully transfer colors from I_L to I_H , especially in the saturated area, which not only degrades the structured detail in the highlight region, but also generates abrupt changes in the image color space. To solve this problem, the color mapping function $g(\cdot)$ described in section 3.1 needs to be modified to cover a larger range. In our experiments, we adopt the color transfer technique in [13] in this situation. It also operates on image histogram, which transfers the color from the source image to the target by matching the mean and standard deviation for each channel. It has no limit on the maximum value of the transferred color since the process is actually a Gaussian matching. In our method, all non-saturated pixels in I_H are used for color transfer to I_L . After applying [13], the mapping result of I_L exceeds the color depth (that is, above 255), and extends the saturated pixels to larger color values. Hence, we construct a higher intensity range² image to reveal details in both bright and dark regions.

3.3 Spatial Constraint

The statistics depicted above does not consider any temporal coherence between I_H and I_L . However, since the two images are taken successively, there is a strong spatial constraint between I_H and I_L .

Let us consider the situation that a region contains similar color pixels, Fig. 3(a) shows a region from the original image, while Fig. 3(b) shows the same region taken with motion blur. The yellow dots mark the region centers. The lower curves show pixel colors along one direction. From this figure, we can observe that the color toward the center of the region is less affected by blurring, given that the region area is sufficient large and homogeneous. Additionally, the consistency of colors in the region also guarantees that the color of central pixels can be matched. Therefore, we adopt the following region matching method to robustly select matching seeds in I_H and I_L :

1. Over-segment I_H such that each region $R_m(I_H)$ contains similar colors (Fig. 4(a)).
2. To sort all regions according to the homogeneity and size, we perform the same morphological eroding operation for each region $R_m(I_H)$, and record the number of iterations to completely erode it and the region centers which are the last few pixels in the eroding process for each region. Fig. 4(b) shows an intermediate image in the eroding process.
3. We sort all iteration numbers in descending order, and select the first M regions as the most possible candidates. As a result, the positions of these region centers are selected as matching positions. Finally, we pick out pixel pairs $\{c_L^m, c_H^m\}$ in I_H and I_L in the matching position and calculate the value for each c^m as a Gaussian average of the colors of neighboring pixels,

² We do not construct HDR since we do not perform radiometric calibration

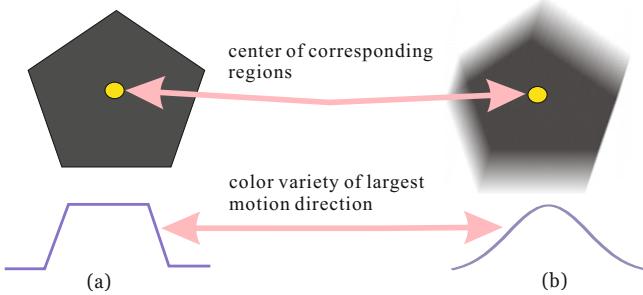


Fig. 3. Matching homogeneous region in blurred situation. (a) original homogeneous region. (b) blurred region. Color towards the center is less influenced by blurring.

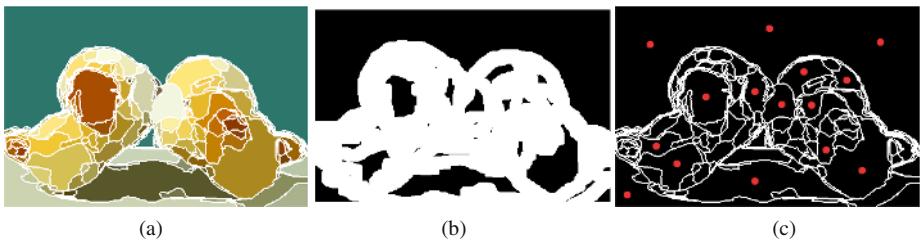


Fig. 4. Region matching process. (a) Initial segmentation. (b) In the eroding process, small regions are filled quickly. (c) The final selected regions, in which the red dots represent the selected region centers after eroding.

where the variance is proportional to the iteration numbers. We illustrate the selected region centers as red dots in Fig. 4(c), which are in the largest and most homogeneous M regions.

The matching process implies that an ideal color mapping function should robustly transform some matching seeds colors in I_L to those in I_H . In the next section, we propose our Bayesian framework which incorporates the two constraints, color and spatial, into consideration, so as to infer a constrained mapping function.

4 Constrained Mapping Function

We define the color mapping function $f(\ell_i) = \ell'_i$, where ℓ_i and ℓ'_i are color values in the two sets respectively. Accordingly, the resulting image I_C is built by applying $f(\cdot)$ to the under-exposed image I_L : $I_C(x, y) = f(I_L(x, y))$, where $I_k(x, y)$ is pixel values in image I_k . Note that the form of $f(\cdot)$ is constrained by both I_L and I_H .

In Bayesian framework, we maximize the *a posteriori* probability (MAP) to infer f^* given the observations from I_L and I_H :

$$f^* = \arg \max_f p(f | I_L, I_H) \quad (2)$$

In section 3, we observe two kinds of connections between I_L and I_H . One is color statistics which can be described by two histograms h_{I_L} and h_{I_H} of I_L and I_H respectively. The other is region matching constraint which can be represented by a number of M corresponding matching color seeds $\{c_L^m, c_H^m\}_{m=1}^M$ between I_L and I_H . In our formulation, we regard them as our constraints and rewrite (2) as:

$$\begin{aligned} f^* &= \arg \max_f p(f | h_{I_L}, h_{I_H}, \{c_L^m, c_H^m\}_{m=1}^M) \\ &= \arg \max_f p(h_{I_L}, h_{I_H}, \{c_L^m, c_H^m\}_{m=1}^M | f) p(f) \end{aligned} \quad (3)$$

Next, we define the likelihood $p(h_{I_L}, h_{I_H}, \{c_L^m, c_H^m\}_{m=1}^M | f)$, and the prior $p(f)$.

4.1 Likelihood

Since we perform global matching in discrete color space, f is approximated by a set of discrete values $f = \{f_1, f_2, \dots, f_i, \dots, f_N\}$, where N is the total number of bins in color space. Hence, the likelihood in Eqn. (3) can be factorized under the i.i.d. assumption:

$$p(h_{I_L}, h_{I_H}, \{c_L^m, c_H^m\}_{m=1}^M | f) = \prod_{i=1}^N p(g(\ell_i), \{\bar{c}_L^i, \bar{c}_H^i\} | f_i) \quad (4)$$

where $g(\ell_i)$ is a function to transform h_{I_L} to h_{I_H} at color value ℓ_i . The \bar{c}_L^i is the most similar color to ℓ_i in color seeds set $\{c_L^m\}_{m=1}^M$, and \bar{c}_H^i is the corresponding color of \bar{c}_L^i in color seed pairs.

According to the analysis in section 3, $g(\ell_i)$ and $\{\bar{c}_L^i, \bar{c}_H^i\}$ are two constraint factors for each f_i . Both of their properties should be maintained on the mapping function. As a consequence, we balance the two constraints and model the likelihood as follows:

$$p(g(\ell_i), \{\bar{c}_L^i, \bar{c}_H^i\} | f_i) \propto \exp\left(-\frac{\|f_i - (\alpha g(\ell_i) + (1 - \alpha)\bar{c}_L^i)\|^2}{2\sigma_I^2}\right) \quad (5)$$

where the scale α weights these two constraints, and σ_I^2 is a variance to model the uncertainty of two kinds of constraints. The larger the value of α is, the smaller the confidence of the matching seed pairs. We relate α to the following factors:

- The distance $\|\ell_i - \bar{c}_L^i\|$. Large distance indicates weak region matching constraint, which makes α approach to 1. Hence, the α is inversely proportional to this distance.
- The uncertainty of correspondence in matching color pair $\{\bar{c}_L^i, \bar{c}_H^i\}$. As depicted in section 3.3, the larger the matching region size is, the larger confidence we can get from the region center for the matching colors. Hence, we define uncertainty σ_c to be proportional to the region size for each matching color.

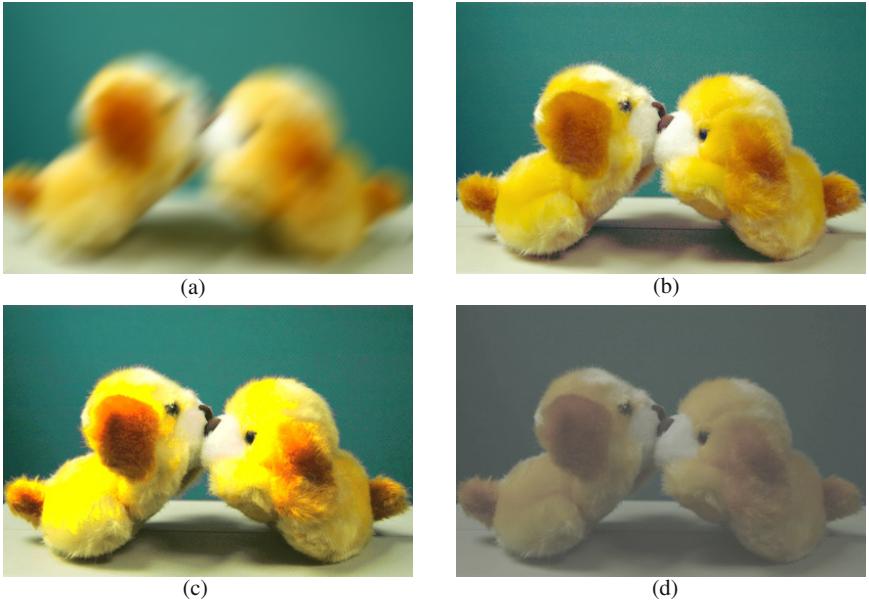


Fig. 5. Puppies. (a) Input blurred image. (b) Our result. (c) Color transfer result [13]. (d) Result of Gamma correction by 2.5. Better visual quality and more details are achieved by using spatial constraint in our framework.

Combining these two factors, we define α as:

$$\alpha = \exp\left(-\frac{\sigma_c^2 \|\ell_i - \bar{c}_L^i\|^2}{2\beta^2}\right) \quad (6)$$

where β is the scale parameter to control the influence of α .

4.2 Prior

As a prior, we enforce the monotonic constraint on $f(\cdot)$, which maintains the structural details in I_L . In addition, to avoid abrupt change of the color mapping for neighboring colors, we require that $f(\cdot)$ be smooth in its shape. In this paper, we minimize the second derivative of f :

$$\begin{aligned} p(f) &\propto \exp\left(-\frac{1}{2\sigma_f^2} \int (f'')^2\right) \\ &\propto \exp\left(-\frac{1}{2\sigma_f^2} \sum_i (f_{i-1} - 2f_i + f_{i+1})^2\right) \end{aligned} \quad (7)$$

where σ_f^2 is the variance to control the smoothness of f .

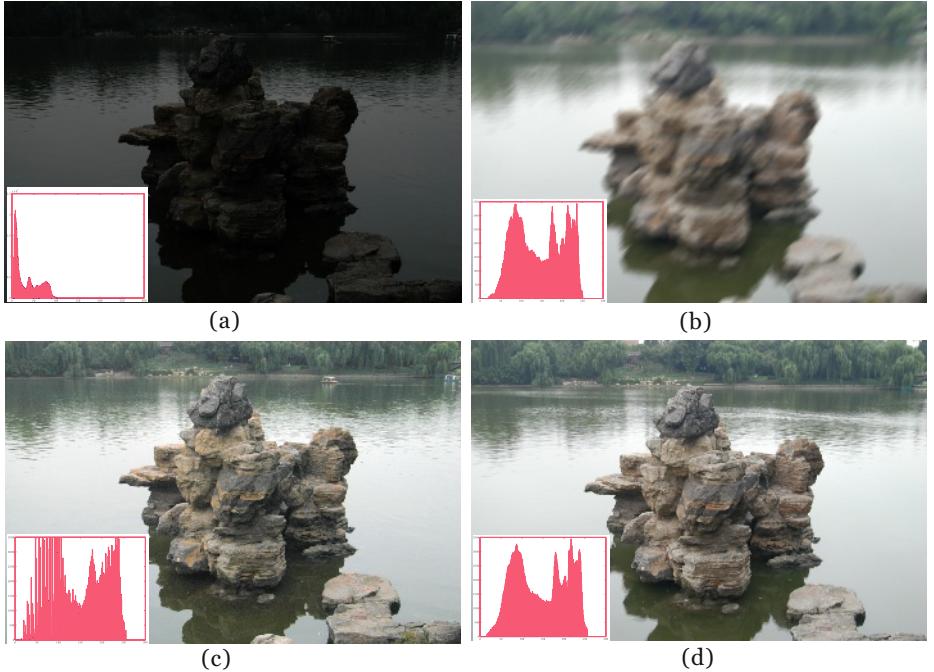


Fig. 6. Rock example of image correction. The upper two images are input defective images. (c) is our result. (d) is the ground truth. Note the histograms in (c) and (d) are much closer than those in (a) and (b). However, because of the quantization error and large exposure difference between I_L and I_H , they can not be identical in shapes.

4.3 MAP Solution

Combining the log likelihood of Eqn. (4) and the log prior in Eqn. (7), we solve the optimization problem by minimizing the following log posterior function:

$$E(f) = - \sum_i \log p(g(\ell_i), \{\bar{c}_L^i, \bar{c}_H^i\} | f_i) - \log p(f) \quad (8)$$

where $E(f)$ is a quadratic objective function. Therefore, the global optimal mapping function $f(\cdot)$ can be obtained by the singular value decomposition (SVD).

Although the monotonic constraint is not enforced explicitly in Eqn. (7), we find the smoothness constraint is sufficient to construct the final monotonic f in our experiments.

5 Results

We evaluate our method in difficult scenarios to show the efficacy of our approach. The results are classified into 4 different groups as follows, all of them are illustrated in color:

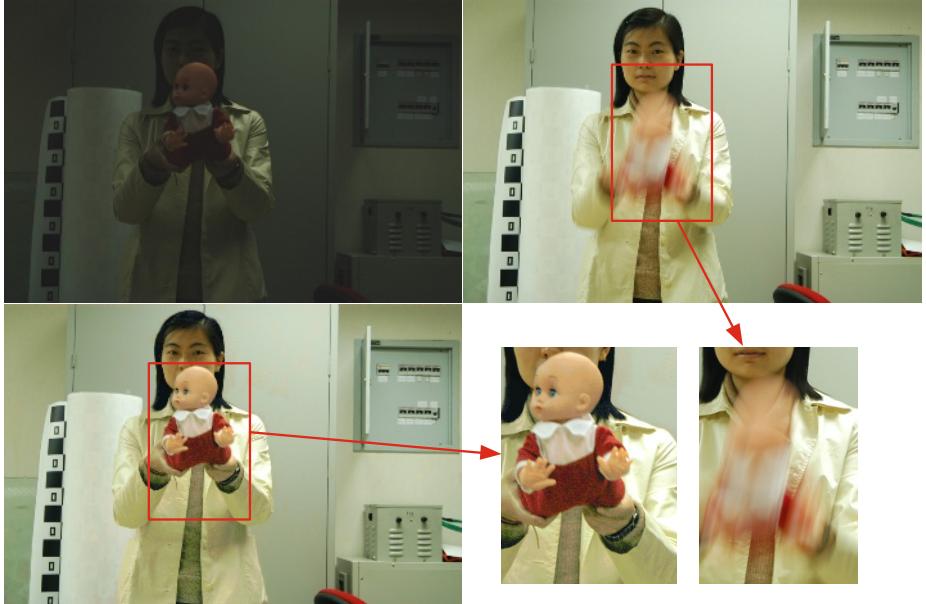


Fig. 7. Doll example. The upper two images are our input. Our result is the left bottom image, which indicates that local blurring in images can be naturally handled.

5.1 Bayesian Color Mapping versus Other Adjustment Techniques

The two constraints described in section 3 are both essential in our method. They optimize the solution in two different aspects cooperatively. Therefore, the combination and balance of these constraints guarantee the visual correctness of our method. Fig. 5 compare our result with that from pure color transfer method [13] and adaptive histogram equalization. We take the first two images in Fig. 1 as input. They are taken with shutter speed $\frac{1}{30}$ s and $\frac{1}{1.6}$ s respectively. Fig. 5(b) is generated with our method. Fig. 5(c) and (d) are the results of pure color transfer and gamma correction. Clearly, Fig. 5(b) has higher visual quality, and the colors are closest to the input image in Fig. 5(a).

5.2 Motion Blur Caused by Hand-Held Camera

The rock example in Fig. 6 shows the ability of our method to optimally combine the color information of the two input images. Unlike other deblurring methods, the resulting edges are very crisp and clear. The two input images (a) and (b) are taken with shutter speeds $\frac{1}{40}$ s and $\frac{1}{3}$ s respectively. (c) and (d) are our color mapped image I_C and ground truth with their corresponding histograms. The ground truth is taken by using a tripod. Note that colors are visually and statistically close.

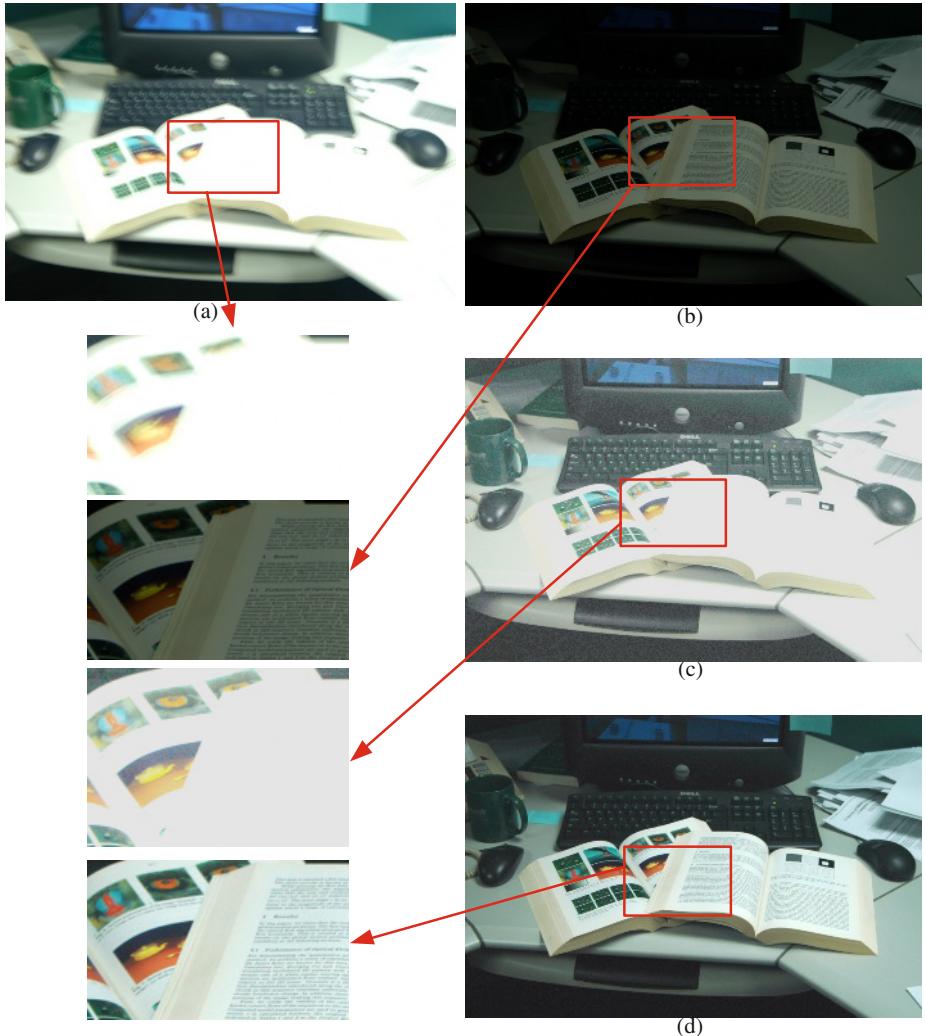


Fig. 8. Image correction for high contrast scene. (a) I_H , which has a large saturated area. (b) I_L has clear structure information. (c) The result produced by applying original histogram equalization. (d) Our final result I_C where pixel intensity values are enhanced and fine details are maintained. The bottom left images are selected enlarged portions of I_C .

5.3 Motion Blur Caused by Objects Movement

Another strength of our method is that we can easily solve the object movement or deformation problem if the object movement is too fast in normal exposure interval. Fig. 7 illustrates one experiment. The input normal exposure image is locally blurred, i.e., PSF has no uniform representation in the whole image,

which easily makes deconvolving methods fail. In our method, by reducing the camera shutter speed by 4 stops, we produce I_C with largely reduced blurring effect.

5.4 High Contrast Scene

As described in section 3.2, for high contrast scene, we modify the statistical color mapping function from adaptive histogram equalization to the color transfer function [13] in the framework. We present our results in Fig. 8. (a) and (b) are input I_H and I_L , respectively, and (c) is reconstructed by setting $g(\cdot)$ as the original histogram equalization function. (d) is our final result with enhanced colors and details by modifying $g(\cdot)$ to use the color transfer method in [13]. Tone mapping [5] is performed to display the image we constructed in (d).

6 Conclusion

In this paper, we propose a Bayesian approach to combine two defective images to construct a high quality image of the scene, which may contain moving objects. No special hardware is built to compensate camera shake. Instead, a color mapping approach is adopted. Yet our color mapping is constrained by spatial details given by the under-exposed image, and thus differs from and improves on previous pure color transfer techniques. By properly formulating color statistics and spatial constraints, and incorporating them into our Bayesian framework, the MAP solution produces an optimal color mapping function that preserves structural details while enhancing pixel colors simultaneously. Using only two images in all our experiments, we produce a high quality image, and largely reduce the shutter speed by 3 to 4 stops to enhance the image quality in dim light.

However, the color statistics is largely dependent the image quality of the camera. If the dark image contains a large amount of noise, the contaminated information needs to be treated first. One solution is taking more under-exposed images to reduce noise level. Another issue is the search for spatial correspondence in the presence of fast movement of camera or objects. These issues will be investigated in future work.

References

1. *Complete Digital Photography (2nd Edition)*. Charles River Media Press, 2002.
2. B. Basile, Andrew Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, pages 573–582, 1996.
3. Moshe Ben-Ezra and Shree K. Nayar. Motion deblurring using hybrid imaging. *Proceedings of CVPR*, 2003.
4. T.W.Cornin D.L.Rudeman and C.C.Chiao. Statistics of cone responses to natural images: Implications for visual coding. In *J. Optical Soc. of America*, number 8, pages 2036–2045, 1998.

5. Peter Shirley Erik Reinhard, Mike Stark and Jim Ferwerda. Photographic tone reproduction for digital images. In *Siggraph 2002*, pages 267–276, 2002.
6. R. Fabian and D. Malah. Robust identification of motion and out-of-focus blur parameters from blurred and noisy images. *CVGIP: Graphical Models and Image Processing.*, 1991.
7. M. D. Grossberg and S. K. Nayar. What can be known about the radiometric response function from images? In *ECCV*, May 2002.
8. Canon Inc. http://www.canon.com.my/techno/optical/optical_b.htm.
9. Nikon Inc. http://www.nikon.co.jp/main/eng/society/tec-rep/tr8-vr_e.htm.
10. D. Kundur and D. Hatzinakos. A novel blind deconvolution scheme for image restoration using recursive filtering. *IEEE Transactions on Signal Processing.*, pages 46(2):375–390, February 1998.
11. X. Liu and A. Gamal. Simultaneous image formation and motion blur restoration via multiple capture. *Proc. Int. Conf. Acoustics, Speech, Signal Processing.*, 2001.
12. A. Rav-Acha and S. Peleg. Restoration of multiple images with motion blur in different directions. *IEEE Workshop on Applications of Computer Vision (WACV)*, 2000.
13. M. Gooch B. Reinhard, E. Ashikhmin and P. Shirley. Color transfer between images. In *IEEE Computer Graphics and Applications*, pages 34–40, 2001.
14. A. Lantzman Y. Yitzhaky, I. Mor and N. S. Kopeika. Direct method for restoration of motion-blurred images. *J. Opt. Soc. Am. A.*, pages 15(6):1512–1519, June 1998.
15. Y. Levy Y. Yitzhaky, G. Boshusha and N.S. Kopeika. Restoration of an image degraded by vibrations using only a single frame. *Optical Engineering*, 2002.

Stretching Bayesian Learning in the Relevance Feedback of Image Retrieval

Ruofei Zhang and Zhongfei (Mark) Zhang

Department of Computer Science
State University of New York at Binghamton, Binghamton, NY 13902, USA
`{rzhang,zhongfei}@cs.binghamton.edu`

Abstract. This paper is about the work on user relevance feedback in image retrieval. We take this problem as a standard two-class pattern classification problem aiming at refining the retrieval precision by learning through the user relevance feedback data. However, we have investigated the problem by noting two important unique characteristics of the problem: small sample collection and asymmetric sample distributions between positive and negative samples. We have developed a novel approach to stretching Bayesian learning to solve for this problem by explicitly exploiting the two unique characteristics, which is the methodology of **B**Ayesian **L**earning in **A**symmetric and **S**mall sample collections, thus called **BALAS**. Different learning strategies are used for positive and negative sample collections in **BALAS**, respectively, based on the two unique characteristics. By defining the relevancy confidence as the relevant posterior probability, we have developed an integrated ranking scheme in **BALAS** which complementarily combines the subjective relevancy confidence and the objective feature-based distance measure to capture the overall retrieval semantics. The experimental evaluations have confirmed the rationale of the proposed ranking scheme, and have also demonstrated that **BALAS** is superior to an existing relevance feedback method in the current literature in capturing the overall retrieval semantics.

1 Introduction

This paper is on Content-Based Image Retrieval (CBIR). Since 1990's, CBIR has attracted significant research attention [8]. Early research focused on finding the "best" representation for image features. The similarity between two images is typically determined by the distances of individual low-level features and the retrieval process is performed by a k - nn search in the feature space [1]. In this context, high level concepts and user's perception subjectivity cannot be well modelled. Recent approaches introduce human-computer interaction (HCI) into CBIR. The interaction mechanism allows a user to submit a coarse initial query and continuously refine his(her) searching via relevance feedback. This approach greatly reduces the labor required to precisely compose a query and easily captures the user's subjective retrieval preference.

However, most approaches to relevance feedback were based on heuristic formulation of empirical parameter adjustment and/or feature component reweighting, which is typically *ad hoc* and not systematic, and thus cannot be substantiated well. Some of the recent work [16,14,4] investigated the problem from a more systematic point of view by formulating the relevance feedback problem as a general classification or learning problem and used optimization methods to address it. These learning methods are all based on the assumption that both positive and negative samples confirm either implicitly or explicitly a well formed distribution. We note that without further exploiting the unique characteristics of training samples in the relevance feedback of image retrieval, it is difficult to map the image retrieval problem to a general two-class (i.e., relevance vs. irrelevance) classification problem in realistic applications. Consequently before we design a specific relevance feedback methodology, two unique characteristics of the relevance feedback problem in image retrieval must be noted and addressed. The first is the small sample collection issue. In relevance feedback of image retrieval, the number of training samples is usually small (typically < 20 in each round of interaction) relative to the dimensionality of the feature space (from dozens to hundreds, or even more), whereas the number of image classes or categories is usually large for many real-world image databases. The second characteristic is the asymmetric training sample issue. Most classification or learning techniques proposed in the literature of pattern recognition and computer vision, such as discriminant analysis [6] and Support Vector Machine(SVM) [15], regard the positive and negative examples interchangeably and assume that both sets are distributed approximately equally. However, in relevance feedback, while it is reasonable to assume that all the positive samples confirm to a specific class distribution, it is typically not valid to make the same assumption for the negative samples, as there may be an arbitrary number of semantic classes for the negative samples to a given query; thus, the small, limited number of negative examples is unlikely to be representative for all the irrelevant classes, and this asymmetric characteristic must be taken into account in the relevance feedback learning.

In this paper, we investigate the relevance feedback problem in image retrieval using Bayesian learning. Specifically, we stretch Bayesian learning by explicitly exploiting the two unique characteristics through developing a novel user relevance feedback methodology in image retrieval — **B**AYesian Learning in **A**symmetric and **S**mall sample collections, called **BALAS**. In **BALAS**, we introduce specific strategies to estimate the probabilistic density functions for the positive and negative sample collections, respectively. It is shown that an optimal classification can be achieved when a scheme for measuring the relevancy confidence is developed to reflect the *subjective* relevancy degree of an image w.r.t. a query image. The relevancy confidence is integrated with the measure of feature-based distance, which reflects the *objective* proximity degree between feature vectors, to order the ranking of the retrieved images from a database.

2 BALAS Methodology

Given a query image, a “good” relevance feedback method would, after learning, allow as many as relevant images to be retrieved and reject as many as irrelevant images from being retrieved. Given a feature space in which each image is represented as a feature vector, we apply Bayesian theory to determine the degree in which an image in the database is classified as a relevant or an irrelevant one to the query image. It is proven that Bayesian rule is optimal in the expectation of misclassification aspect [6].

We define the notations as follows. We always use boldface symbols to represent vectors or matrices, and non-boldface symbols to represent scalar variables. Given a query image, Let R and I be the events of the relevancy and irrelevancy for all the images in the image database to a query image, respectively, and let Img_i be the i th image in the image database. We use $P()$ to denote a probability, and use $p()$ to denote a probability density function (pdf). Thus, $P(R)$ and $P(I)$ are the prior probabilities of relevancy and irrelevancy for all the images in the image database to the query image, respectively; and $p(Img_i)$ is the pdf of the i th image in the image database. Based on the Bayes’ rule the following equations hold:

$$P(R|Img_i) = \frac{p(Img_i|R)P(R)}{p(Img_i)}, \quad P(I|Img_i) = \frac{p(Img_i|I)P(I)}{p(Img_i)} \quad (1)$$

where $i = 1, \dots, M$ and M is the number of images in the database.

Definition 1. *Given a specific image Img_i in the image database, for any query image, the relevancy confidence of this image to the query image is defined as the posterior probability $P(R|Img_i)$. Similarly, the irrelevancy confidence of this image to the query image is defined as the posterior probability $P(I|Img_i)$. Obviously, the two confidences are related as $P(R|Img_i) + P(I|Img_i) = 1$.*

The relevancy confidence and irrelevancy confidence of an image are used to indicate the *subjective* relevance and irrelevance degree quantitatively to the query image, respectively. From Eq. 1, the problem of determining whether an image Img_i is (ir)relevant to the query image and the corresponding (ir)relevancy confidence is reduced to estimating the conditional pdfs $p(Img_i|R)$ and $p(Img_i|I)$, respectively, the prior probabilities $P(R)$ and $P(I)$, respectively, and the pdf $p(Img_i)$ in the continuous feature space. These probabilities and pdfs may be estimated from the positive and negative samples provided by the user relevance feedback, as we shall show below.

Since in CBIR, each image is always represented as a feature vector or a group of feature vectors (when each feature vector is used to represent a region or an object in the image) in a feature space, to facilitate the discussion we use a feature vector to represent an image in this paper. Consequently, in the rest of this paper, we use the terminologies vector and image interchangeably. Due to the typical high dimensionality of feature vectors, it is safe and desirable to perform vector quantization before the pdf estimations to ease the computation intensity.

As a preprocessing, we apply uniform quantization to every dimension of feature vectors and each interval is represented by its corresponding representative value.

It is straightforward to estimate the pdf $p(Img_i)$ by statistically counting the percentage of the quantized feature vectors in the feature space of the whole image database. Note that this estimation is performed offline and for each image it is only required to be computed once, resulting in no complexity for online retrieval. For image databases updated with batch manner(most practical databases are updated in this way), the content of databases does not change during the online search session, and periodically updating $p(Img_i)$ along with the database updating is feasible.

Since it is well observed that all the positive (i.e., the relevant) samples “are alike in a way” [19]. In other words some features of the class-of-interest usually have compact support in reality. We assume that the pdf of each feature dimension of all the relevant images to a given query image satisfies the Gaussian distribution.

$$p(x_k|R) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left[-\frac{(x_k - m_k)^2}{2\sigma_k^2}\right] \quad (2)$$

where x_k is the k^{th} dimension of the feature vector of an image, m_k is the mean value of the x_k of all relevant images to the query image, and σ_k is the variance of the corresponding x_k .

To verify this model for positive samples, we tested on images of several predefined semantic categories. The experiment confirms that the model is practically acceptable. Fig. 1(a) shows a quantile-quantile test [9] of the standardized *hue* feature of 100 images in one predefined semantic category. It is shown that the quantile of the standardized feature dimension and the quantile of the standard Gaussian distribution are similar, which means that the feature dimension of the 100 images in this semantic category can be approximated as a Gaussian.

Assume that $L = \{l^1, l^2, \dots, l^N\}$ is the labelled relevant sample set. Applying the maximum-likelihood method [2], we obtain the following unbiased estimations of the mean vector m_k and the variance σ_k :

$$\widehat{m}_k = \frac{1}{N} \sum_{i=1}^N l_k^i, \quad \widehat{\sigma}_k = \frac{1}{N-1} \sum_{i=1}^N (l_k^i - \widehat{m}_k)^2 \quad (3)$$

In order to ensure that these estimations are close to the true values of the parameters, we must have sufficient relevant samples. However, the number of relevant samples in each relevance feedback iteration is typically limited. Hence, we develop a cumulative strategy to increase the number of relevant samples. Specifically, the relevant samples in each iterations in a query session are recorded over the iterations; when we estimate the parameters using Eq. 3, we not only use the relevant samples labelled by the user in the current iteration, but also include all the relevant samples recorded in the previous iterations to improve the estimation accuracy.

It is notable that not every feature dimension of relevant images conforms to a Gaussian distribution equally well. It is possible that, for one semantic category,

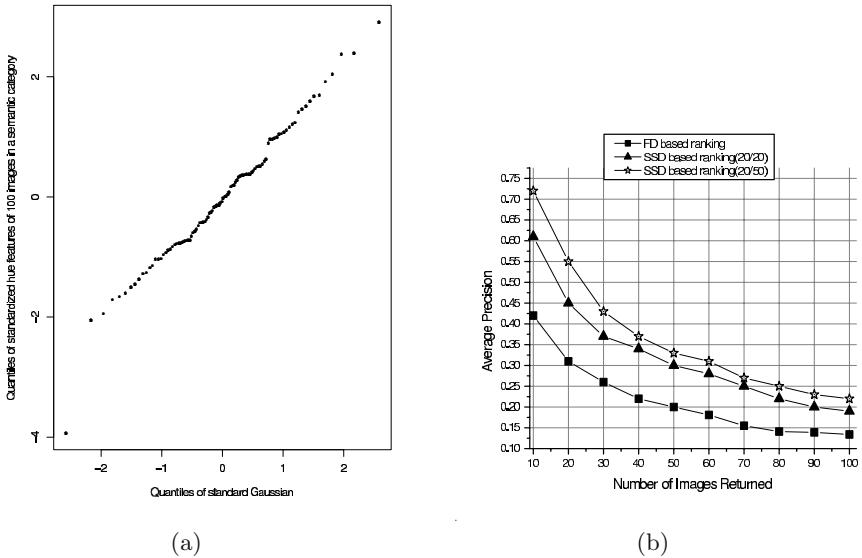


Fig. 1. (a): Quantile-quantile test of a standardized feature dimension for images in one semantic category. (b): Average precisions vs. the numbers of the returned images with and without **BALAS** enabled.

some feature dimensions are more semantically related than other dimensions such that these dimensions appear to conform to a Gaussian model better, while other dimensions' distributions in the feature space are jumbled, and thus do not conform to Gaussian well. To describe the difference of conformity degrees and compensate the corresponding effect we introduce a measure, called *trustworthy degree*, for every feature dimension. The trustworthy degree depicts the importance weight for every feature dimension. It is defined as $w_k = \frac{\sigma_k^{-1}}{\max_{k=1}^T \sigma_k^{-1}}$, where T is the number of dimensions in one image feature. If the variance of the relevant samples is high along a dimension k , we deduce that the values on this dimension are not very relevant to the query image and thus the Gaussian distribution might not be a good model for this dimension because the features are not centered with a prominent mean. Consequently a low trustworthy degree w_k is assigned. Otherwise, a high trustworthy degree w_k is assigned. Note that the $\max w_k = 1$ for $k = 1 \dots T$.

It is reasonable to assume all dimensions of one feature are independent (raw features *per se* are independent, e. g., color and texture features, or we can always apply K-L transform [5] to generate uncorrelated features from raw features; in this way the support to independency is strengthened), thus the pdf of positive samples is determined as a trustworthy degree pruned joint pdf:

$$p(\mathbf{x}|R) = \prod_{\substack{k=1 \\ w_k \geq \delta}}^T p(x_k|R) \quad (4)$$

where δ is a threshold for incorporating only high trustworthy dimensions (conforming to the Gaussian model well) to determine $p(\mathbf{x}|R)$. Those dimensions that do not conform to the Gaussian distribution well would result in inaccurate pdf estimations, and consequently are filtered out.

In order to correctly and accurately estimate the conditional pdf distribution for the negative samples, we assume that each negative sample represents a unique potential semantic class, and we apply the kernel density estimator [12] to determining the statistical distribution function of this irrelevance class. In case two negative samples happen to come from the same semantic class, it is supposed that they would exhibit the same distribution function, and thus this assumption is still valid. Consequently, the overall pdf for the negative samples is the agglomeration of all the kernel functions.

We choose the kernel function in the estimator as an isotropic Gaussian function (assuming all the feature vectors have been normalized). The window of the estimation is a hyper-sphere centered at each negative sample $\mathbf{x}_j, j = 1, 2, \dots, N$, assuming that there are N negative samples in total. Let the radius of the j th hyper-sphere be r_j , which is called the *bandwidth* of the kernel density estimation in the literature [3]. Typically it is practical to assume that $r_j = r$ for all the different j , where r is a constant bandwidth. Hence, the conditional pdf to be estimated for the sample \mathbf{x}_i in the feature space is given by

$$p(\mathbf{x}_i|I) = \sum_{j=1}^N \text{kernel}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{j=1}^N \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2r_j^2}\right\} \quad (5)$$

where $\|\mathbf{x}_i - \mathbf{x}_j\|_2$ is the Euclidian distance between the neighboring sample \mathbf{x}_j and the center feature vector \mathbf{x}_i .

The choice of the bandwidth r has an important effect in the estimated pdfs. If the bandwidth is too large, the estimation would suffer from low resolution. On the other hand, if the bandwidth is too small, the estimation may be locally overfitted, hurting the generalization of the estimation. In this consideration, the optimal Parzen window size has been studied extensively in the literature [13]. In practice, the optimal bandwidth may be determined by minimizing the *integrated squared error* (ISE), or the *mean integrated squared error* (MISE). Adaptive bandwidth is also proposed in the literature [13]. For simplicity, we choose a constant bandwidth r based on the maximum distance from all the negative samples to their closest neighbor D defined as $r = \lambda D = \lambda \max_{\mathbf{x}_k} [\min_{\mathbf{x}_l} (\|\mathbf{x}_k - \mathbf{x}_l\|_2)]$, where λ is a scalar. We find in our experiments that with well-normalized feature vectors, a λ between 1 and 10 often gives good results.

The computational overhead in estimating conditional pdf with Eq. 5 is tractable due to the limited number of negative samples and utilization of dimensionality reduction techniques, such as PCA [5], on the low-level features, while the estimation accuracy is acceptable.

Since negative samples may potentially belong to different semantic classes, and since each such semantic class only has a very limited number of samples thus far in one typical relevance feedback iteration, we must “generate” a sufficient number of samples to ensure that the estimated pdf for the negative

samples is accurate. To solve for this “scarce sample collection” problem, we actually generate additional negative samples based on the kernel distributions for each semantic classes defined in Eq. 5. These generated additional samples are the hypothetical images. For the sake of discussion, we call the original negative samples provided by the user in the relevance feedback iterations as the *labelled* samples, and the generated samples as the *unlabelled* samples. To ensure that the number of generated samples is sufficiently large, for each labelled negative sample in one relevance feedback iteration, we generate q additional unlabelled negative samples based on Eq. 5, where q is a parameter. To ensure a “fair sampling” to the kernel function in Eq. 5, the generation of the unlabelled samples follows a probability function defined by the following Gaussian pdf function $p(\mathbf{y}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{\|\mathbf{y}-\mathbf{x}_i\|_2^2}{2\sigma^2}\right\}$, where $d = \|\mathbf{y}-\mathbf{x}_i\|_2$ is the Euclidian distance between the unlabelled sample \mathbf{y} and each labelled sample \mathbf{x}_i , and σ is the standard deviation, which is set to the average distance between two feature vectors in the labelled negative feature space defined as $\sigma = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^N \|\mathbf{x}_i - \mathbf{x}_j\|_2$.

Hence, an unlabelled sample is more likely to be selected if it is close to a labelled negative sample. The probability density defined in above decays when the Euclidian distance to the labelled sample increases.

Consequently, an algorithm, called SAMPLING, is designed to perform the unlabelled sample selection based on roulette wheel selection strategy [2]. SAMPLING implements a roulette wheel sampling strategy to select unlabelled samples. The unlabelled samples with smaller distances to a labelled sample have larger probabilities to be selected as the additional samples. On the other hand, those potential unlabelled samples farther away from a labelled sample are not completely eliminated from being selected, though their chances of being selected are small. With the extended number of the negative samples, the accuracy of the pdf estimation defined in Eq. 5 is significantly improved. Similarly, the cumulative learning principle adopted in the estimation of the conditional pdf for the positive samples described above is also applied in the estimation of the conditional pdf for the negative samples to further improve the estimation accuracy.

In order to determine the relevancy and irrelevancy confidences defined as the posterior probabilities in Eq. 1, we must solve for the prior probabilities $P(R)$ and $P(I)$ first. Unlike the typical approach in the classical pattern classification problems in which a prior probability is usually estimated from the supervised training samples, in the problem of the relevance feedback in image retrieval the relevancy or irrelevancy of an image is subject to different query images and different users’ subjective preferences. Thus, the relevancy and irrelevancy of an image vary in different queries and in different query sessions. Consequently, it is impossible to estimate the prior probabilities in advance. In other words, these prior probabilities must be estimated online also in solving for the relevance feedback problem.

Given a query image, for each image Img_i in the image database, we have

$$p(Img_i) = p(Img_i|R)P(R) + p(Img_i|I)P(I) \quad (6)$$

and for the query image we also have

$$P(R) + P(I) = 1 \quad (7)$$

Combining Eqs. 6 and 7, we immediately have:

$$P(R) = \frac{p(Img_i) - p(Img_i|I)}{p(Img_i|R) - p(Img_i|I)} \quad (8)$$

From Eq. 8, it is clear that since we have already developed methods to determine $p(Img_i|R)$, $p(Img_i|I)$, and $p(Img_i)$, the prior probability $P(R)$ may be uniquely determined immediately. Thus, $P(I)$ may also be immediately determined from Eq. 7. This reveals that for each query image given, the *overall* relevancy and irrelevancy of *all* the images in the image database may be uniquely determined by *any individual* image Img_i in the image database. In other words, any individual image Img_i in the image database may be used to determine the prior probabilities, and given a query image, the prior probabilities are independent of the selection of any of the images in the database. The experimental results have verified this conclusion. Nevertheless, due to the noise in the data, in practice, the estimated prior probabilities based on different individual images in the database may exhibit slight variations. In order to give an accurate estimation of the prior probabilities that are not subject to the bias towards a specific image in the database, we denote $P_i(R)$ as the prior probability determined in Eq. 8 using the individual image Img_i , and $P(R)$ is the average from all the images in the database, i.e., $P(R) = \frac{1}{M} \sum_{i=1}^M P_i(R)$. The prior probability $P(I)$ is thus determined accordingly.

Given a query image in a query session, for each image Img_i in the database, there is a corresponding relevancy confidence $P(R|Img_i)$, which represents the relevancy degree of this image to the query image learned from the user's subjective preference through the relevance feedback. Hence, this relevancy confidence captures the *subjective* relevancy degree of each image in the database to a query. On the other hand, for any CBIR system, there is always a feature-based distance measure used for image retrieval. The feature-based distance measure typically does not incorporate the user relevance preferences, and thus, only captures the *objective* proximity degree in the feature space of each image in the database to a query. Consequently, in order to design a ranking scheme in image retrieval that "makes best sense", it is natural to consider to integrate the subjective relevancy confidence and the objective distance measure together through taking advantage of labelled sample image set to define an comprehensive ranking scheme.

Note that the relevancy confidence and the feature-based distance measure are complementary to each other. Exploiting this property explicitly, we define a unified ranking scheme, called *Session Semantic Distance* (SSD), to measure the relevance of any image Img_i within the image database in terms of both relevancy confidence $P(R|Img_i)$, irrelevancy confidence $P(I|Img_j)$, and feature-based distance measure $FD(Img_i)$.

The SSD for any image $SSD(Img_i)$ is defined using a modified form of the Rocchio's formula [10] as follows:

$$\begin{aligned} SSD(Img_i) &= \log(1 + P(R|Img_i))FD(Img_i) \\ &\quad + \beta\left\{\frac{1}{N_R} \sum_{k \in D_R} [(1 + P(R|Img_k))D_{ik}]\right\} \\ &\quad - \gamma\left\{\frac{1}{N_I} \sum_{k \in D_I} [(1 + P(I|Img_k))D_{ik}]\right\} \end{aligned} \quad (9)$$

where N_R and N_I are the sizes of the positive and negative labelled sample set D_R and D_I , respectively, in the feedback. D_{ik} is the feature-based distance between the image Img_i and Img_k . We have replaced the first parameter α in Rocchio's formula with the logarithm of the relevancy confidence of the image Img_i . The other two parameters β and γ are assigned a value of 1.0 in our current implementation of the system for the sake of simplicity. However, other values can be given to emphasize the different weights between the last two terms.

With this definition of the $SSD(Img_i)$, the relevancy confidence of Img_i , the relevancy confidence of images in the labelled relevant set, the irrelevancy confidence of images in the labelled irrelevant set, and the objective feature distance measure are integrated in a unified way. The (ir)relevancy confidences of images in the labelled sample set act adaptively as weights to correct the feature-based distance measure. In the ranking scheme, an image is ranked high in the returned list if it is similar, in relevancy confidence measure and/or feature-based distance measure, to the query image and images in the labelled relevant image set and it is dissimilar to images in the labelled irrelevant image set in both relevancy confidence and feature-based distance measure; otherwise, its rank is low. Thus, the robustness and accuracy of the semantic distance measure is improved, resulting in lower false-positives, by using both subjective and objective similarity measures to form a more accurate measure for semantic similarity.

3 Experiments and Discussions

The focus of this paper is on user relevance feedback in image retrieval rather than on a specific image indexing and retrieval method. The relevance feedback methodology we have developed in this paper, **BALAS**, is independent of any specific image indexing and retrieval methods, and in principle, may be applied to any such image indexing and retrieval methods. The objective of this section is to demonstrate that **BALAS** can effectively improve the image retrieval relevancy through the user relevance feedback using any specific CBIR system.

For the evaluation purpose, we implemented an image indexing and retrieval prototype system. Many types of low-level features may be used to describe the content of images. In the current implementation, we use color moment. We extract the first two moments from each channel of CIE-LUV color space, and the simple yet effective $L2$ distance is used to be the feature-based ranking metric. Since the objective is to test the relevance feedback learning method

rather than to evaluate features, the feature we use is not as sophisticated as those used in some existing CBIR systems [17,18].

The following evaluations are performed on a general-purpose color image database containing 10,000 images from the COREL collection with 96 categories. 1,500 images were randomly selected from all the categories of this image database to be the query set. A retrieved image is considered semantics-relevant if it is in the same category of the query image. We note that the category information in the COREL collection is only used to ground-truth the evaluation, and we do not make use of this information in the indexing and retrieval procedures.

In order to evaluate the semantics learning capability of **BALAS**, we implemented the **BALAS** methodology on the prototype CBIR system, which we also call **BALAS** for the purpose of the discussion in this paper. The threshold δ in Eq. 4 was empirically set as 0.7 in the prototype. Since user relevance feedback requires subjective feedback, we invite a group of 5 users to participate the evaluations. The participants consist of CS graduate students as well as lay-people outside the CS Department. We ask different users to run **BALAS** initially without the relevance feedback interaction, and then to place their relevance feedbacks after the initial retrievals. For the evaluation purpose, we define the retrieval precision as the ratio of the number of relevant images retrieved to the total number of retrieved images in each round of the retrieval in a query session. For the comparison purpose, we have recorded the retrieval precisions in the initial retrieval, i.e., without the **BALAS** relevance feedback capability and purely based on the similarity measure, the retrieval precisions after every rounds of relevance feedback using **BALAS** *only* based on the relevancy confidence, and the retrieval precisions after every rounds of relevance feedback using **BALAS** based on the session semantic distance, respectively. All the reported data are the averages of the whole group of users. The average time for each round of retrieval after the relevance input is about 5 seconds on a *PentiumIV* 2GHz computer with 512MB memory.

We ran the implemented CBIR system with **BALAS** for the 1,500 query image set with varied number of truncated top retrieved images and plotted the curves of the average retrieval precision vs. the number of truncated top retrieved images. Fig. 1(b) shows the average precision-scope plot for the system with and without **BALAS** enabled. In other words, for ranking scheme one test is based solely on feature-based distance *FD* and another test is based on session semantic distance *SSD* with different numbers of provided sample images. The notation (m/n) in the figure legend denotes the number of positive sample images vs. number of negative sample images for the learning. It is clear that the **BALAS** relevance feedback learning capability enhances the retrieval effectiveness substantially.

For performance comparison, we used the same image database and the query set to compare **BALAS** with the relevance feedback method developed by Yong and Huang [11], which is a combination and improvement of its early version and MindReader [7] and represents the state-of-the-art relevance feedback research in the literature. Two versions of [11] are implemented. The first uses the color moments (CM) computed in the same way as described above and the other

uses the correlogram (and thus is called CG here). The overall comparison evaluations are documented in Fig. 2(a). The average precision in this evaluation is determined based on the top 100 returned images for each query out of the 1,500 query image set. From the figure, it appears that during the first two iterations, the CG version of [11] performs noticeably better than **BALAS** while the CM version of [11] performs comparably with **BALAS**. After the second iteration, **BALAS** exhibits a significant improvement in performance over that of [11] in either of the two versions, and as the number of iterations increases, the improvement of the performance of **BALAS** over [11] appears to increase also. This also confirms with the cumulative learning strategy employed in **BALAS** and the fact that when more iterations of relevance feedback are conducted, more learning samples are given, and thus more accurate density estimation may be expected from **BALAS**.

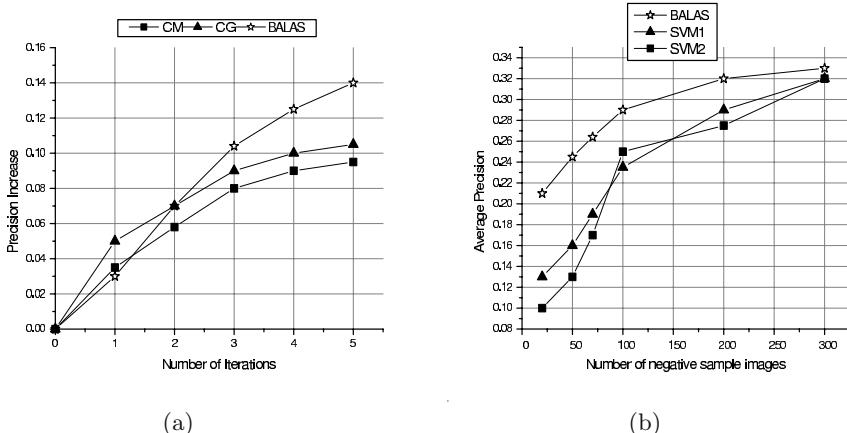


Fig. 2. (a): Retrieval precision comparison using relevance feedback between **BALAS** and CM and CG. (b): Average precision in top 100 images returned. Number of positive sample images =20. SVM1 denotes SVM classifier with $\sigma = 50$ and SVM2 denotes SVM classifier with $\sigma = 100$.

To evaluate the effectiveness of explicitly addressing the asymmetry property of CBIR, we compared **BALAS** with SVM [15] classification method. SVM classifier adopts the two-class assumption and treats positive and negative samples equally, which is not valid in CBIR as is discussed above. In addition, there is no satisfactory method to optimally select kernel function and its parameters other than empirically testing yet. In the comparison experiment, the RBF kernel $K(x, y) = \exp^{-\|x-y\|^2/2\sigma^2}$ with different σ s were tested for SVM classifier. The original SVM classifier only gives a decision boundary without providing confidence of each object belonging to each class. To utilize SVM classifiers in image retrieval, a ranking scheme is needed. In the comparison, *Larger margin first* retrieval scheme [15] is adopted for SVM to determine the rank of the

retrieved images. A query set was composed of randomly selected 100 images, which was applied to **BALAS** and SVM, respectively; the average precisions in the top 100 images were recorded for different number of negative sample images with the number of positive samples images fixed. SVM with two different σ s were tested; $\sigma = 50$ and $\sigma = 100$. Fig. 2(b) shows the result. We see that the performance of SVM is affected by σ in some degree but **BALAS** outperforms SVM consistently. The unsatisfactory performance of SVM is partially due to the false assumption that the two classes are equivalent and the negative samples are representative of the true distributions. With this invalid assumption, we found that the positive part “spills over” freely into the part of the unlabelled areas of the feature space by the SVM classification. The result of this “spillover” effect is that after the user’s feedback, the machine returns a totally different set of images, with most of them likely to be negative. In **BALAS**, this phenomenon did not occur due to the asymmetric density estimations.

4 Conclusions

This paper focuses work on user relevance feedback in image retrieval. We take this problem as a standard two-class pattern classification problem aiming at refining the retrieval precision by learning through the user relevance feedback data. However, we have investigated the problem by noting two important unique characteristics: small sample collection and asymmetric sample distributions between positive and negative samples. We have developed a novel approach to stretching Bayesian learning to solve for this problem by explicitly exploiting the two unique characteristics, which is the methodology of BAyesian Learning in Asymmetric and Small sample collections, thus called **BALAS**. Different learning strategies are used for positive and negative sample collections in **BALAS**, respectively, based on the two unique characteristics. By defining the relevancy confidence as the relevant posterior probability, we have developed an integrated ranking scheme in **BALAS** which complementarily combines the subjective relevancy confidence and the objective similarity measure to capture the overall retrieval semantics. The experimental evaluations have confirmed the rationale of the proposed ranking scheme, and have also demonstrated that **BALAS** is superior to an existing relevance feedback method in the literature in capturing the overall retrieval semantics.

References

1. A. D. Bimbo. *Visual Information Retrieval*. Morgan kaufmann Pub., San Francisco, CA, 1999.
2. G. Blom. *Probability and Statistics: Theory and Applications*. Springer Verlag, London, U. K., 1989.
3. S.-T. Chiu. A comparative review of bandwidth selection for kernel density estimation. *Statistica Sinica*, 16:129–145, 1996.

4. I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The bayesian image retrieval system, pichunter: Theory, implementation and psychophysical experiments. *IEEE Trans. on Image Processing*, 9(1):20–37, 2000.
5. W. R. Dillon and M. Goldstein. *Multivariate Analysis, Methods and Applications*. John Wiley and Sons, New York, 1984.
6. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.
7. Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Query databases through multiple examples. In *the 24th VLDB Conference Proceedings*, New York, 1998.
8. M. D. Marsicoi, L. Cinque, and S. Levialdi. Indexing pictorial documents by their content: a survey of current techniques. *Image and Vision Computing*, 15:119–141, 1997.
9. B. D. Ripley and W. N. Venables. *Modern Applied Statistics with S*. Springer Verlag, New York, New York, 2002.
10. Rocchio and J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System — Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Inc, Englewood Cliffs, NJ, 1971.
11. Y. Rui and T. S. Huang. Optimizing learning in image retrieval. In *IEEE Conf. Computer Vision and Pattern Recognition*, South Carolina, June 2000.
12. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.
13. G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20:1236–1265, 1992.
14. K. Tieu and P. Viola. Boosting image retrieval. In *IEEE Conf. Computer Vision and Pattern Recognition Proceedings*, South Carolina, June 2000.
15. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
16. Y. Wu, Q. Tian, and T. S. Huang. Discriminant em algorithm with application to image retrieval. In *IEEE Conf. Computer Vision and Pattern Recognition Proceedings*, South Carolina, June 2000.
17. R. Zhang and Z. Zhang. Addressing cbir efficiency, effectiveness, and retrieval subjectivity simultaneously. In *ACM Multimedia 2003 Multimedia Information Retrieval Workshop*, Berkeley, CA, November 2003.
18. R. Zhang and Z. Zhang. A robust color object analysis approach to efficient image retrieval. *EURASIP Journal on Applied Signal Processing*, 2004.
19. X. S. Zhou and T. S. Huang. Small sample learning during multimedia retrieval using biasmap. In *IEEE Conf. Computer Vision and Pattern Recognition Proceedings*, Hawaii, December 2001.

Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera

Antonis A. Argyros and Manolis I.A. Lourakis

Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
Vassiliaka Vouton, P.O.Box 1385, GR 711 10
Heraklion, Crete, GREECE
`{argyros,lourakis}@ics.forth.gr`

Abstract. This paper presents a method for tracking multiple skin-colored objects in images acquired by a possibly moving camera. The proposed method encompasses a collection of techniques that enable the modeling and detection of skin-colored objects as well as their temporal association in image sequences. Skin-colored objects are detected with a Bayesian classifier which is bootstrapped with a small set of training data. Then, an off-line iterative training procedure is employed to refine the classifier using additional training images. On-line adaptation of skin-color probabilities is used to enable the classifier to cope with illumination changes. Tracking over time is realized through a novel technique which can handle multiple skin-colored objects. Such objects may move in complex trajectories and occlude each other in the field of view of a possibly moving camera. Moreover, the number of tracked objects may vary in time. A prototype implementation of the developed system operates on 320x240 live video in real time (28Hz) on a conventional Pentium 4 processor. Representative experimental results from the application of this prototype to image sequences are also provided.

1 Introduction

An essential building block of many vision systems is one that permits tracking objects of interest in a temporal sequence of images. For example, this is the case of systems able to interpret the activities of humans, in which we are particularly interested. Such systems depend on effective and efficient tracking of a human operator or parts of his body (e.g. hands, face), as he performs a certain task. In this context, vision-based tracking needs to provide answers to the following fundamental questions. First, how is a human modeled and how are instances of the employed model detected in an image? Second, how are instances of the detected model associated temporally in sequences of images?

The human body is a complex, non-rigid structure with many degrees of freedom. Therefore, the type and complexity of the models employed for tracking vary dramatically [4,3], depending heavily on the requirements of the application domain under consideration. For example, tracking people in an indoors

environment in the context of a surveillance application has completely different modeling requirements compared to tracking the fingers of a hand for sign language interpretation. Many visual cues like color, texture, motion and structure have been employed as the basis for the modeling of human body parts. Among those, skin color is very effective towards detecting the presence of humans in a scene. Color offers many advantages over geometric models, such as robustness under occlusions, scale and resolution changes, as well as geometric transformations. Additionally, the computational requirements of color processing are considerably lower compared to those associated with the processing of complex geometric models.

In the remainder of this section, we review existing approaches based on the answers they provide to the two fundamental questions stated above and, then, we outline the proposed method for addressing the tracking problem.

1.1 Color Modeling and Detection

A recent survey [22] includes a very interesting overview of the use of color for face (and, therefore skin-color) detection. A major decision towards providing a model of skin color is the selection of the color space to be employed. Several color spaces have been proposed including RGB [8], normalized RGB [12,10], HSV [15], YCrCb [2], YUV [20], etc. Color spaces efficiently separating the chrominance from the luminance components of color are typically considered preferable. This is due to the fact that by employing chrominance-dependent components of color only, some degree of robustness to illumination changes can be achieved. Terrillon et al [18] review different skin chrominance models and evaluate their performance.

Having selected a suitable color space, the simplest approach for defining what constitutes skin color is to employ bounds on the coordinates of the selected space [2]. These bounds are typically selected empirically, i.e. by examining the distribution of skin colors in a preselected set of images. Another approach is to assume that the probabilities of skin colors follow a distribution that can be learned either off-line or by employing an on-line iterative method [15]. In the case of non-parametric approaches, the learnt distribution is represented by means of a color probabilities histogram. Other, so-called parametric approaches, are based either on a unimodal Gaussian probability density function [12,20] or multimodal Gaussian mixtures [9,14] that model the probability distribution of skin color. The parameters of a unimodal Gaussian density function are estimated by maximum likelihood estimation techniques. Multi-modal Gaussian mixtures require the Expectation-Maximization (EM) algorithm to be employed. According to Yang et al [21], a mixture of Gaussians is preferable compared to a single Gaussian distribution. Still, [10] argues that histogram models provide better accuracy and incur lower computational costs compared to mixture models for the detection of skin-colored areas in an image. A few of the proposed methods perform some sort of adaptation to become insensitive to changes in the illumination conditions. For example in [14] it has been suggested to adapt a

Gaussian mixture model that approximates the multi-modal distribution of the object's colors, based on a recent history of detected skin-colored regions.

1.2 Temporal Data Association

Assuming that skin-colored regions have been modeled and can be reliably detected in an image, another major problem relates to the temporal association of these observations in an image sequence. The traditional approach to solving this problem has been based on the original work of Kalman [11] and its extensions. If the observations and object dynamics are of Gaussian nature, Kalman filtering suffices to solve the tracking problem. However, in many cases the involved distributions are non-Gaussian and, therefore, the underlying assumptions of Kalman filtering are violated. As reported in [17], recent research efforts that deal with object tracking can be classified into two categories, the ones that solve the tracking problem in a non-Bayesian framework (e.g. [7,16,19]) and the ones that tackle it in a Bayesian one (e.g. [6,13,5]). In some cases [6], the problem of single-object tracking is investigated. These single-object approaches usually rely upon sophisticated, powerful object models. In other cases [13,5] the problem of tracking several objects in parallel is addressed. Some of these methods solve the multi-object tracking problem by employing configurations of individual objects, thus reducing the multi-object tracking problem to several instances of the less difficult single-object tracking problem. Other methods employ particle filtering based algorithms, which track multiple objects simultaneously. Despite the considerable amount of research devoted to tracking, an efficient and robust solution to the general formulation of the problem is still lacking, especially for the case of simultaneous tracking of multiple targets.

1.3 Proposed Approach

With respect to the two fundamental questions that have been posed, the proposed approach relies on a non-parametric method for skin-color detection and performs tracking in a non-Bayesian framework. Compared to existing approaches, the proposed method has several attractive properties. A skin-color representation is learned through an off-line procedure. A technique is proposed that permits the avoidance of much of the burden involved in the process of generating training data. Moreover, the proposed method adapts the employed skin-color model based on the recent history of tracked skin-colored objects. Thus, without relying on complex models, it is able to robustly and efficiently detect skin-colored objects even in the case of changing illumination conditions. Tracking over time is performed by employing a novel technique that can cope with multiple skin-colored objects, moving in complex patterns in the field of view of a possibly moving camera. Furthermore, the employed method is very efficient, computationally. The developed tracker operates on live video at a rate of 28 Hz on a Pentium 4 processor running under MS Windows.

The rest of the paper is organized as follows. Section 2 presents the proposed tracker. Section 3 provides sample results from the operation of the tracker in

long image sequences as well as issues related to its computational performance. Finally, section 4 provides the main conclusions of this work as well as extensions that are under investigation.

2 Method Description

The proposed method for tracking multiple skin-colored objects operates as follows. At each time instance, the camera acquires an image on which skin-colored blobs (i.e. connected sets of skin-colored pixels) are detected. The method also maintains a set of object hypotheses that have been tracked up to this instance in time. The detected blobs, together with the object hypotheses are then associated in time. The goal of this association is (a) to assign a new, unique label to each new object that enters the camera's field of view for the first time, and (b) to propagate in time the labels of already detected objects. What follows, is a more detailed description of the approach adopted to solve the aforementioned subproblems.

2.1 Skin Color Detection

Skin color detection involves (a) estimation of the probability of a pixel being skin-colored, (b) hysteresis thresholding on the derived probabilities map, (c) connected components labeling to yield skin-colored blobs and, (d) computation of statistical information for each blob. Skin color detection adopts a Bayesian approach, involving an iterative training phase and an adaptive detection phase.

Basic training and detection mechanisms. A small set of training input images is selected on which a human operator manually delineates skin-colored regions. The color representation used in this process is YUV 4:2:2. However, the Y-component of this representation is not employed for two reasons. First, the Y-component corresponds to the illumination of an image point and therefore, by omitting it, the developed classifier becomes less sensitive to illumination changes. Second, by employing a 2D color representation (UV), as opposed to a 3D one (YUV), the dimensionality of the problem is reduced, as are the computational requirements of the overall system.

Assuming that image points $I(x, y)$ have a color $c = c(x, y)$, the training set is used to compute (a) the prior probability $P(s)$ of skin color, (b) the prior probability $P(c)$ of the occurrence of each color c in the training set and (c) the prior probability $P(c|s)$ of a color c being a skin color. Following the training phase, the probability $P(s|c)$ of a color c being a skin color can be computed by employing the Bayes rule:

$$P(s|c) = P(c|s)P(s)/P(c) \quad (1)$$

Then, the probability of each image point being skin-colored can be determined and all image points with probability $P(s|c) > T_{max}$ are considered as being

skin-colored. These points constitute the seeds of potential blobs. More specifically, image points with probability $P(s|c) > T_{min}$ where $T_{min} < T_{max}$, that are immediate neighbors of skin-colored image points are recursively added to each blob. The rationale behind this region growing operation is that an image point with relatively low probability of being skin-colored should be considered as such in the case that it is a neighbor of an image point with high probability of being skin-colored. This hysteresis thresholding type of operation has been very successfully applied to edge detection [1] and also proves extremely useful in the context of robust detection of skin-colored blobs. Indicative values for the thresholds T_{max} and T_{min} are 0.5 and 0.15, respectively. A connected components labeling algorithm is then responsible for assigning different labels to the image points of different blobs. Size filtering on the derived connected components is also performed to eliminate small, isolated blobs that are attributed to noise and do not correspond to interesting skin-colored regions. Each of the remaining connected components corresponds to a skin-colored blob. The final step in skin color detection is the computation of up to second order moments for each blob that will be used in the tracking process.

Simplifying off-line training. Training is an off-line procedure that does not affect the on-line performance of the tracker. Nevertheless, the compilation of a sufficiently representative training set is a time-consuming and labor-intensive process. To cope with this problem, an adaptive training procedure has been developed. Training is performed on a small set of seed images for which a human provides ground truth by defining skin-colored regions. Alternatively, already existing, publicly available training sets can be employed. Following this, detection together with hysteresis thresholding is used to continuously update the prior probabilities $P(s)$, $P(c)$ and $P(c|s)$ based on a larger image data set. The updated prior probabilities are used to classify points of these images into skin-colored and non-skin-colored ones. In cases where the classifier produces wrong results (false positives / false negatives), manual user intervention for correcting these errors is necessary; still, up to this point, the classifier has automatically completed much of the required work. The final training of the classifier is then performed based on the training set that results after user editing. This process for adapting the prior probabilities $P(s)$, $P(c)$ and $P(c|s)$ can either be disabled as soon as it is decided that the achieved training is sufficient for the purposes of the tracker or continue as more input images are fed to the system.

Adaptive detection. In the case of varying illumination conditions, skin color detection may produce poor results, even if the employed color representation has certain illumination-independent characteristics. Hence, a mechanism that adapts the representation of skin-colored image points according to the recent history of detected skin-colored points is required. To solve this problem, skin color detection maintains two sets of prior probabilities $P(s)$, $P(c)$, $P(c|s)$, corresponding to the off-line training set and $P_w(s)$, $P_w(c)$, $P_w(c|s)$, corresponding to

the evidence that the system gathers during the w most recent frames. Clearly, the second set better reflects the “recent” appearance of skin-colored objects and is better adapted to the current illumination conditions. Skin color detection is then performed based on:

$$P(s|c) = \gamma P(s|c) + (1 - \gamma)P_w(s|c), \quad (2)$$

where $P(s|c)$ and $P_w(s|c)$ are both given by eq. (1) but involve prior probabilities that have been computed from the whole training set and from the detection results in the last w frames, respectively. In eq. (2), γ is a sensitivity parameter that controls the influence of the training set in the detection process. Setting $w = 5$ and $\gamma = 0.8$ gave rise to very good results in a series of experiments involving gradual variations of illumination.

2.2 Tracking Multiple Objects over Time

We assume that at time t , M blobs have been detected as described in section 2.1. Each blob b_j , $1 \leq j \leq M$, corresponds to a set of connected skin-colored image points. Note that the correspondence among blobs and objects is not necessarily one-to-one. As an example, two crossing hands are two different skin-colored objects that appear as one blob at the time one occludes the other. In this work we assume that an object may correspond to either one blob or part of a blob. Symmetrically, one blob may correspond to one or many objects.

We also assume that the spatial distribution of the pixels depicting a skin-colored object can be coarsely approximated by an ellipse. This assumption is valid for skin-colored objects like hand palms and faces. Let N be the number of skin-colored objects present in the viewed scene at time t and o_i , $1 \leq i \leq N$, be the set of skin pixels that image the i -th object. We also denote with $h_i = h_i(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i)$ the ellipse model of this object where (c_{x_i}, c_{y_i}) is its centroid, α_i and β_i are, respectively, the lengths of its major and minor axis, and θ_i is its orientation on the image plane. Finally, we use capital letters $B = \cup_{j=1}^M b_j$, $O = \cup_{i=1}^N o_i$, and $H = \cup_{i=1}^N h_i$ to denote the union of skin-colored pixels, object pixels and ellipses, respectively. Tracking amounts to determining the relation between object models (h_i) and observations (b_j) in time. Figure 1 exemplifies the problem. In this particular example there are three blobs (b_1 , b_2 and b_3) while there are four object hypotheses (h_1 , h_2 , h_3 and h_4) from the previous frame.

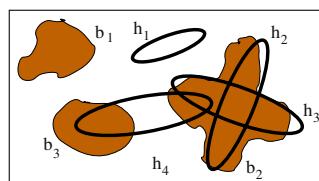


Fig. 1. Various cases of the relation between skin-colored blobs and object hypotheses.

What follows is an algorithm that can cope effectively with the data association problem. The proposed algorithm needs to address three different subproblems: (a) object hypothesis generation (i.e. an object appears in the field of view for the first time) (b) object hypothesis tracking in the presence of multiple, potential occluding objects (i.e. previously detected objects move arbitrarily in the field of view) and (c) object model hypothesis removal (i.e. a tracked object disappears from the field of view).

Object hypothesis generation. We define the distance $D(p, h)$ of a point $p = p(x, y)$ from an ellipse $h(c_x, c_y, \alpha, \beta, \theta)$ as follows:

$$D(p, h) = \sqrt{\vec{v} \cdot \vec{v}} \quad (3)$$

where

$$\vec{v} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \left(\frac{x - x_c}{\alpha}, \frac{y - y_c}{\beta} \right)$$

From the definition of $D(p, h)$ it turns out that the value of this metric is less than 1.0, equal to 1.0 or greater than 1.0 depending on whether point p is inside, on, or outside ellipse h , respectively. Consider now a model ellipse h and a point p belonging to a blob b . In the case that $D(p, h) < 1.0$, we conclude that the point p and the blob b support the existence of the object hypothesis h and that object hypothesis h predicts blob b . Consider now a blob b such that:

$$\forall p \in b, \min_{h \in H} \{D(p, h)\} > 1.0. \quad (4)$$

Equation (4) describes a blob with empty intersection with all ellipses of the existing object hypotheses. Blob b_1 in Fig. 1 is such a case. This, implies that none of the existing object hypotheses accounts for the existence of this blob. For each such blob, a new object hypothesis is generated. The parameters of the generated object hypothesis can be derived directly from the statistics of the distribution of points belonging to the blob. The center of the ellipse of the object hypothesis becomes equal to the centroid of the blob and the rest of the ellipse parameters can be computed from the covariance matrix of the bivariate distribution of the location of blob points. More specifically, it can be shown that if the covariance matrix Σ of the blob's points distribution is $\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$, then an ellipse can be defined with parameters:

$$\alpha = \sqrt{\lambda_1}, \quad \beta = \sqrt{\lambda_2}, \quad \theta = \tan^{-1} \left(\frac{-\sigma_{xy}}{\lambda_1 - \sigma_{yy}} \right) \quad (5)$$

where $\lambda_1 = \frac{\sigma_{xx} + \sigma_{yy} + \Lambda}{2}$, $\lambda_2 = \frac{\sigma_{xx} + \sigma_{yy} - \Lambda}{2}$, and $\Lambda = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 - 4\sigma_{xy}^2}$.

Algorithmically, at each time t , all detected blobs are tested against the criterion of eq. (4). For all qualifying blobs, an object hypothesis is formed and the corresponding ellipse parameters are determined based on eqs. (5). Moreover, all such blobs are excluded from further consideration in the subsequent steps of object tracking.

Object hypothesis tracking. After new object hypotheses have been formed as described in the previous section, all the remaining blobs must support the existence of past object hypotheses. The main task of the tracking algorithm amounts to associating blob pixels to object hypotheses. There are two rules governing this association:

- **Rule 1:** If a skin-colored pixel of a blob is located within the ellipse of some object hypothesis (i.e. supports the existence of the hypothesis) then this pixel is considered as belonging to this hypothesis.
- **Rule 2:** If a skin-colored pixel is outside all ellipses corresponding to the object hypotheses, then it is assigned to the object hypothesis that is closer to it, using the distance metric of eq. (3).

Formally, the set o of skin-colored pixels that are associated with an object hypothesis h is given by $o = R_1 \cup R_2$ where $R_1 = \{p \in B \mid D(p, h) < 1.0\}$ and $R_2 = \{p \in B \mid D(p, h) = \min_{k \in H} \{D(p, k)\}\}$.

In the example of Fig. 1, two different object hypotheses (h_2 and h_3) are “competing” for the skin-colored area of blob b_2 . According to the rule 1 above, all skin pixels within the ellipse of h_2 will be assigned to it. According to the same rule, the same will happen for skin pixels under the ellipse of h_3 . Note that pixels in the intersection of these ellipses will be assigned to both hypotheses h_2 and h_3 . According to rule 2, pixels of blob b_2 that are not within any of the ellipses, will be assigned to their closest ellipse which is determined by eq. (3).

Another interesting case is that of a hypothesis that is supported by more than one blobs (see for example hypothesis h_4 in Fig. 1). Such cases may arise when, for example, two objects are connected at the time they first appear in the scene and later split. To cope with situations where a hypothesis h receives support from several blobs, the following strategy is adopted. If there exists only one blob b that is predicted by h and, at the same time, not predicted by any other hypothesis, then h is assigned to b . Otherwise, h is assigned to the blob with which it shares the largest number of skin-colored points. In the example of Fig. 1, hypothesis h_4 gets support from blobs b_2 and b_3 . Based on the above rule, it will be finally assigned to blob b_3 .

After having assigned skin pixels to object hypotheses, the parameters of the object hypotheses h_i are re-estimated based on the statistics of pixels o_i that have been assigned to them.

Object hypothesis removal. An object hypothesis should be removed either when the object moves out of the camera’s field of view, or when the object is occluded by another (non-skin colored) object in the scene. Thus, an object hypothesis h should be removed from further consideration whenever

$$\forall p \in B, D(p, h) > 1.0. \quad (6)$$

Equation (6) essentially describes hypotheses that are not supported by any skin-colored image points. Hypothesis h_1 in Fig. 1 is such a case. In practice, we permit an object hypothesis to “survive” for a certain amount of time, even

in the absence of any support, so that we account for the case of possibly poor skin-color detection. In our implementation, this time interval has been set to half a second, which approximately amounts to fourteen image frames.

Prediction. In the processes of object hypothesis generation, tracking and removal that have been considered so far, data association is based on object hypotheses that have been formed at the previous time step. Therefore, there is a time lag between the definition of models and the acquisition of data these models need to represent. Assuming that the immediate past is a good prediction for the immediate future, a simple linear rule can be used to predict the location of object hypotheses at time t , based on their locations at time $t - 2$ and $t - 1$. Therefore, instead of employing $h_i = h_i(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i)$, as the ellipses describing the object hypothesis i , we actually employ $\widehat{h}_i = h_i(\widehat{c}_{x_i}, \widehat{c}_{y_i}, \alpha_i, \beta_i, \theta_i)$, where $(\widehat{c}_{x_i}(t), \widehat{c}_{y_i}(t)) = C_i(t - 1) + \Delta C_i(t)$. In the last equation, $C_i(t)$ denotes $(c_{x_i}(t), c_{y_i}(t))$ and $\Delta C_i(t) = C_i(t - 1) - C_i(t - 2)$.

The above equations postulate that an object hypothesis will maintain the same direction and magnitude of translation on the image plane, without changing any of its other parameters. Experimental results have shown that this simple prediction mechanism performs surprisingly well in complex object motions, provided that processing is performed close to real-time.

3 Experimental Results

In this section, representative results from a prototype implementation of the proposed tracker are provided. The reported experiment consists of a long (3825 frames) sequence that has been acquired and processed on-line and in real-time on a Pentium 4 laptop computer running MS Windows at 2.56 GHz. A web camera with an IEEE 1394 (Firewire) interface has been used for this experiment.

For the reported experiment, the initial, “seed” training set contained 20 images and was later refined in a semi-automatic manner using 80 additional images. The training set contains images of four different persons that have been acquired under various lighting conditions.

Figure 2 provides a few characteristic snapshots of the experiment. For visualization purposes, the contour of each tracked object hypothesis is shown. Different contour colors correspond to different object hypotheses.

When the experiment starts, the camera is still and the tracker correctly asserts that there are no skin-colored objects in the scene (Fig. 2(a)). Later, the hand of a person enters the field of view of the camera and starts moving at various depths, directions and speeds in front of it. At some point in time, the camera also starts moving in a very jerky way; the camera is mounted on the laptop’s monitor which is being moved back and forth. The person’s second hand enters the field of view; hands now move in overlapping trajectories. Then, the person’s face enters the field of view. Hands disappear and then reappear in the scene. All three objects move independently in disjoint trajectories and in varying speeds ((b)-(d)), ranging from slow to fast; at some point in time the



Fig. 2. Characteristic snapshots from the on-line tracking experiment.

person starts dancing, jumping and moving his hands very fast. The experiment proceeds with hands moving in crossing trajectories. Initially hands cross each other slowly and then very fast ((e)-(g)). Later on, the person starts applauding which results in his hands touching but not crossing each other ((h)-(j)). Right after, the person starts crossing his hands like tying in knots ((k)-(o)). Next,

the hands cross each other and stay like this for a considerable amount of time; then the person starts moving, still keeping his hands crossed ((p)-(r)). Then, the person waves and crosses his hands in front of his face ((s)-(u)). The experiments concludes with the person turning the light on and off ((v)-(x)), while greeting towards the camera (Fig.2(x)).

As it can be verified from the snapshots, the labeling of the object hypotheses is consistent throughout the whole sequence, which indicates that they are correctly tracked. Thus, the proposed tracker performs very well in all the above cases, some of which are challenging. Note also that no images of the person depicted in this experiment were contained in the training set. With respect to computational performance, the 3825 frames sequence presented previously has been acquired and processed at an average frame rate of 28.45 fps (320x240 images). The time required for grabbing a single video frame from the IEEE 1394 interface dominates the tracker's cycle time. When prerecorded image sequences are loaded from disk, considerably higher tracking frame rates can be achieved.

Besides the reported example, the proposed tracker has also been extensively tested with different cameras and in different settings involving different scenes and humans. Demonstration videos including the reported experiment can be found at <http://www.ics.forth.gr/cvrl/demos>.

4 Discussion

In this paper, a new method for tracking multiple skin-colored objects has been presented. The proposed method can cope successfully with multiple objects moving in complex patterns as they dynamically enter and exit the field of view of a camera. Since the tracker is not based on explicit background modeling and subtraction, it may operate even with images acquired by a moving camera. Ongoing research efforts are currently focused on (1) combining the proposed method with binocular stereo processing in order to derive 3D information regarding the tracked objects, (2) providing means for discriminating various types of skin-colored areas (e.g. hands, faces, etc) and (3) developing methods that build upon the proposed tracker in order to be able to track interesting parts of skin-colored areas (e.g. eyes for faces, fingertips for hands, etc).

Acknowledgements. This work was partially supported by EU IST-2001-32184 project ActIPret.

References

1. J.F. Canny. A computational approach to edge detection. *IEEE Trans. on PAMI*, 8(11):769–798, 1986.
2. D. Chai and K.N. Ngan. Locating facial region of a head-and-shoulders color image. In *Proc. of FG'98*, pages 124–129, 1998.
3. Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81:328–357, 2001.

4. D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
5. C. Hue, J.-P. Le Cadre, and P. Perez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Proc.*, 50(2):309–325, 2002.
6. M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. of ECCV'98*, pages 893–908, 1998.
7. O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *Proc. of ECCV'02*, pages 343–357, 2002.
8. T.S. Jebara and A. Pentland. Parameterized structure from motion for 3d adaptive feedback tracking of faces. In *Proc. of CVPR'97*, pages 144–150, 1997.
9. T.S. Jebara, K. Russel, and A. Pentland. Mixture of eigenfeatures for real-time structure from texture. In *Proc. of ICCV'98*, pages 128–135, 1998.
10. M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. In *Proc. of CVPR'99*, volume 1, pages 274–280, 1999.
11. R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ACME-Journal of Basic Engineering*, pages 35–45, 1960.
12. S.H. Kim, N.K. Kim, S.C. Ahn, and H.G. Kim. Object oriented face detection using range and color information. In *Proc. of FG'98*, pages 76–81, 1998.
13. E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34(2-3):93–105, 2001.
14. Y. Raja S. McKenna and S. Gong. Tracking color objects using adaptive mixture models. *IVC journal*, 17(3-4):225–231, 1999.
15. D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Proc. of FG'96*, pages 379–384, 1996.
16. N.T. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *Proc. of ECCV'02*, pages 373–387, 2002.
17. M. Spengler and B. Schiele. Multi object tracking based on a modular knowledge hierarchy. In *Proc. of International Conference on Computer Vision Systems*, pages 373–387, 2003.
18. J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proc. of FG'00*, pages 54–61, 2000.
19. J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001.
20. M.H. Yang and N. Ahuja. Detecting human faces in color images. In *Proc. of ICIP'98*, volume 1, pages 127–130, 1998.
21. M.H. Yang and N. Ahuja. *Face Detection and Gesture Recognition for Human-computer Interaction*. Kluwer Academic Publishers, New York, 2001.
22. M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. on PAMI*, 24(1):34–58, 2002.

Evaluation of Image Fusion Performance with Visible Differences

Vladimir Petrović¹ and Costas Xydeas²

¹ Imaging Science Biomedical Engineering, University of Manchester
Oxford Road, Manchester, M13 9PT, UK
v.petrovic@man.ac.uk

² Dept. Communication Systems, University of Lancaster,
Bailrigg, Lancaster, LA1 4YR, UK
c.xydeas@lancaster.ac.uk

Abstract. Multisensor signal-level image fusion has attracted considerable research attention recently. Whereas it is relatively straightforward to obtain a fused image, e.g. a simple but crude method is to average the input signals, assessing the performance of fusion algorithms is much harder in practice. This is particularly true in widespread “fusion for display” applications where multisensor images are fused and the resulting image is presented to a human operator. As recent studies have shown, the most direct and reliable image fusion evaluation method, subjective tests with a representative sample of potential users are expensive in terms of both time/effort and equipment required. This paper presents an investigation into the application of the Visible signal Differences Prediction modelling, to the objective evaluation of the performance of fusion algorithms. Thus given a pair of input images and a resulting fused image, the Visual Difference Prediction process evaluates the probability that a signal difference between each of the inputs and the fused image can be detected by the human visual system. The resulting probability maps are used to form objective fusion performance metrics and are also integrated with more complex fusion performance measures. Experimental results indicate that the inclusion of visible differences information in fusion assessment yields metrics whose accuracy, with reference to subjective results, is superior to that obtained from the state of the art objective fusion performance measures.

1 Introduction

Multisensor imaging arrays have become reliable sources of information in a growing range of applications. However, in order to fully exploit additional information in scene representations of different sensors, considerable processing effort is required. Furthermore, displaying multiple image modalities to a human operator simultaneously leads to confusion and overload, while integrating information across a group of users is almost impossible [6]. Signal-level image fusion deals with this problem by reducing the physical amount of multisensor (image) data while preserving its information content value [1-5]. Whereas it is relatively straightforward to obtain a fused image, e.g. simply average the inputs, assessing the performance of

fusion algorithms, particularly those intended to produce a visual display is much harder in practice. The most reliable and direct method of evaluating fusion for display are subjective tests in which audiences of intended users evaluate fused images under tightly controlled conditions either by comparing them to each other or by performing specific visually oriented tasks. Subject responses or task performance are logged and need to be processed further to obtain a meaningful performance evaluation making the whole process expensive both in terms of time, effort and equipment required [5-8]. Alternatively objective fusion evaluation metrics require no display equipment or complex organisation of an audience and their advantage is obvious in terms of effort and time expended on the evaluation process. Implementation of such algorithms in computer code and simulation experiments reduces the assessment time from days or weeks to often a few minutes. A significant advantage is also the ability to use objective metrics within the process of fusion system development. Hence a fully computational evaluation metric is able to provide irreplaceable performance change information and thus drive, for example, the process of fusion system parameter optimization in a way that is impossible to achieve using human subjects and complex visual testing procedures.

So far, only a limited number of relatively application dependent objective image fusion performance metrics has been published in the literature [3,4,9,10,12]. Target signature consistency as an evaluation criterion for detection/recognition applications is proposed in [10]. The idea of comparing the fused image to an “ideally” fused image and estimating performance from their difference was used in [3,4]. In both cases images with different focus points were fused manually (cut and paste) to produce the ideal fusion reference. In general however, this method is not generally applicable as in the majority of fusion applications the ideal fused image is “ill” defined and cannot be obtained manually [3,4]. Meanwhile, [2,12] proposed metrics based on mutual information for image sequence and still image fusion performance, respectively.

This paper addresses the issue of objectively predicting the results of subjective image fusion performance tests using Visible Differences (VD). VD is a concept widely used in image fidelity assessment that determines the probability of an observer noticing a difference between the appearances of two signals [14]. In the context of objective fusion assessment VD probability maps are derived from input and fused images and used to form several image fusion metrics. Importantly, VD information is also used to gauge the behaviour and the internal processes involved in subjective fusion evaluation. In the following, detailed description of the approach is provided as well as experimental results of the proposed metrics compared with data obtained from real subjective fusion evaluation tests that demonstrate its subjective relevance and relative advantage compared to the current state of the art.

2 Visible Differences Predictor

The Visual Difference Predictor (VDP) used in the proposed metrics is based on a scheme by Daly [14] that describes the visibility near the visual threshold of the differences between two versions A and B of an image. Figure 1 shows a block diagram of the VDP system that employs a model of the *human visual system* (HVS) and operates the input images A and B. The output of the VDP is a 2-dimensional

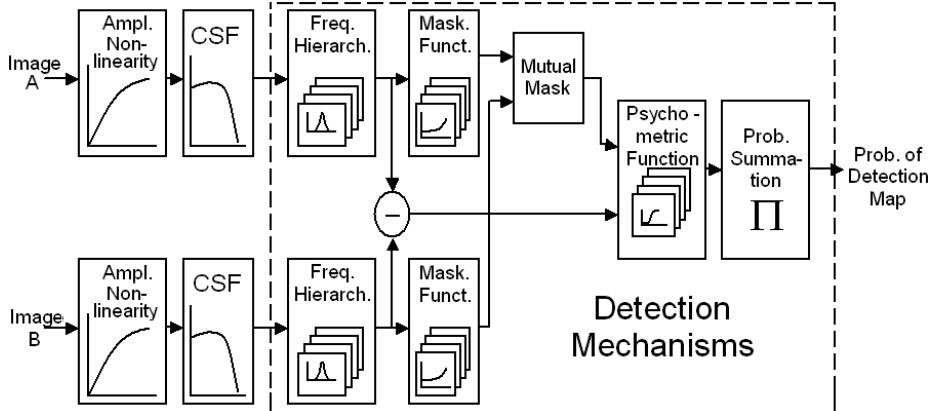


Fig. 1. Structure of the VDP

map P , whose elements $0 \leq P_{m,n} \leq 1$ indicate the probability of detecting visual differences between the two images at every pixel location (m,n) . $P_{m,n}=1$ indicates that differences are *suprathreshold* and completely detectable whereas $P_{m,n}=0$ indicates that the difference between the two images at this location can not be detected. Notice that the VDP output map does not discriminate between different suprathreshold visual distortions, which can be “fully” detected. According to [14], VDP has been successfully tested for many types of image distortion including compression artefacts, blurring, noise, banding, blocking, etc.

The HVS model accounts for three main visual sensitivity variations. These are modelled by the three sequentially cascaded components shown in Fig. 1, i.e. the *amplitude nonlinearity*, the *contrast sensitivity function* (CSF) and the *detection mechanisms*. These variations are functions of light intensity, spatial frequency and image content respectively. The sensitivity variation with respect to light intensity is primarily due to the light adaptive properties of the retina, and it is referred as the *amplitude nonlinearity* of the HVS. The variation in sensitivity with respect to spatial frequency is due to the combined effect of the eye optics and neural circuitry, and is referred as the *contrast sensitivity function* (CSF). Finally, the variation in sensitivity with image content is due to the post-receptor neural circuitry. This effect is referred as *masking* and is modelled by the *detection mechanisms*. A detailed description of an efficient implementation of the VDP system can be found in [13,14].

3 VD Image Fusion Performance Evaluation

The theoretic goal of signal-level image fusion process is to represent all the visual information from a number of images of a scene taken by disparate sensors into a single fused image without distortion or loss of information. In practice however, displaying *all* the information in one image is almost impossible and most algorithms concentrate on faithfully representing only the most important input information. In fusion for display [1-9] systems, this is *perceptually* important information and the fused image should therefore contain all, important input information detectable by an

observer. Additionally, the fusion must not result in distortions or other "false" information appearing in the fused image. The concept of visible differences can be used effectively in fusion performance evaluation through: i) detection of changes in the appearance of visual information transferred (fused) from the input into the fused image and ii) detection of fusion *artefacts* appearing in the fused image but not being present in the inputs.

In this context, VD relationships are defined between each of the input images and the output fused image (differences that exist between inputs are caused by the physical processes of image acquisition and not by the fusion process). Fig. 2 illustrates these relationships on a real image fusion example where two multisensor images (A and B) are fused to produce image F. Probability of visible differences maps between A and B, and F: P^{AF} and P^{BF} (middle column) show white where the probability of a difference being detected is high. They clearly highlight the effects of the information fusion process: areas taken from A are light in P^{AF} and dark in P^{BF} and vice versa.

3.1 Visible Difference Fusion Performance Measures

The VDP-maps P^{AF} and P^{BF} can be used directly to evaluate fusion. A simple measure of fusion performance is the average probability of noticing a difference between the inputs and the fused image. If the fusion process is successful, input image information will be faithfully represented in the fused image resulting in a small probability that observers will notice differences. Fusion systems that produce low $P_{m,n}^{AF}$ and $P_{m,n}^{BF}$ values therefore perform well. Thus a VD fusion metric $VDF \in [0,1]$ with 1 being ideal fusion and 0 being the "worst" possible fusion performance is defined for an $M \times N$ image in equation (1):

$$VDF = 1 - \frac{1}{2MN} \sum_m \sum_n P_{m,n}^{AF} + P_{m,n}^{BF}. \quad (1)$$

Now, image fusion is a data reduction process and lossless overlay of visual information is almost impossible in practice. In recognition of this fact many fusion algorithms rely on some form of spatial feature selection to resolve the problem of superposition by choosing to faithfully represent, at every location only one of the input images. In terms of visible differences, one would expect that at any one location a successfully fused image is similar to at least one of the inputs, resulting in one of the $P_{m,n}^{AF}$, $P_{m,n}^{BF}$ values being small. Accordingly, a $VD_{min} \in [0,1]$ metric is formulated as:

$$VD_{min} = 1 - \frac{1}{MN} \sum_m \sum_n \min(P_{m,n}^{AF}, P_{m,n}^{BF}). \quad (2)$$

Finally, if one is to consider the fact that during subjective fusion evaluation trials the observers' attention is captured by only the most significant differences between the input and fused images, there is a rationale to restrict the measurement to such areas only. Applying a visual detection threshold T_d to $P_{m,n}^{AF}$ and $P_{m,n}^{BF}$ simulates such

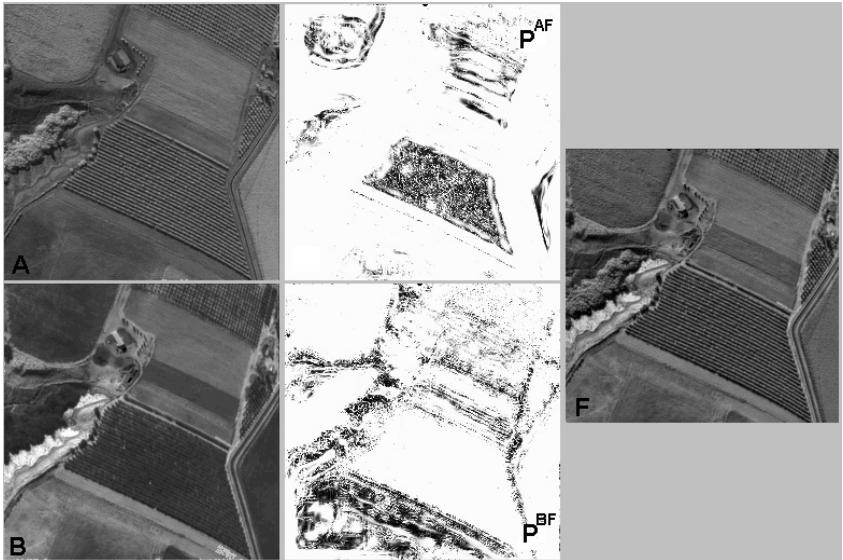


Fig. 2. Visible differences between two input images and the fused image

behaviour and a Visible Differences Area (VDA) metric can be defined which employs only those input image locations that exhibit significant changes between the input and fused images, equation (3). T_d effectively determines the probability level at which image differences are likely to be noticed and thus effect perceived fusion performance. Again, $VDA \in [0,1]$, while all three metric scores for the fusion example in Fig.2 are $VF = 0.12$, $VD_{min} = 0.23$ and $VDA = 0.31$.

$$VDA = 1 - \frac{1}{2MN} \sum_m \sum_n (P_{m,n}^{AF} + P_{m,n}^{BF}) \Big|_{\forall m,n, P_{m,n}^{AF,BF} > T_d} . \quad (3)$$

3.2 VD Information in Edge Based Fusion Assessment

In addition to the above metrics, the concept of visible differences can be applied to enrich an existing fusion performance evaluation framework. The $Q^{AB/F}$ metric [15] evaluates fusion performance by measuring success in preserving input image gradient information in the fused image (see Appendix A). The rationale for combining the VD and $Q^{AB/F}$ evaluation approaches lies in the disparate nature of their approaches promising a more comprehensive assessment. Visual differences information is combined with $Q^{AB/F}$ in two useful ways: *i*) VDP maps are used within the $Q^{AB/F}$ framework in a hybrid Q_{VD} metric and *ii*) both types of metric are independently evaluated and combined *a-posteriori*.

VDP takes into account a whole host of factors that influence attention distribution and the HVS and is introduced into a hybrid Q_{VD} metric by defining the relative importance of each pixel in the input images as $P_{w^A}^{AF}$ and $P_{w^B}^{BF}$ instead of simply w^A and w^B , equation (8). In this manner more importance is assigned to locations that are

generally salient (high w^A) and exhibit significant changes between the fused and input images (high P^{AF}). Consequently fusion performance at these locations has more influence on the final fusion performance score. The other, more *ad hoc*, alternative is a simple linear combination (VD+Q) of independently evaluated VD and $Q^{AB/F}$ metrics through a weighted sum, equation (4) $M_{VD} \in \{VDF, VD_{min}, VDA\}$. Coefficient $b \in [0,1]$ determines the relative emphasis of the two metrics on the final score. A higher value of b indicates that gradient information ($Q^{AB/F}$) dominates the visual information fusion assessment process.

$$Q = bQ^{AB/F} + (1 - b)M_{VD}. \quad (4)$$

4 Results

The aim of the proposed objective fusion performance metrics is subjective relevance, the highest possible level of correspondence with subjective performance scores. This is examined using results of eight different subjective fusion assessment tests [5,15]. In the tests, groups of images (two inputs and two different fused images), Fig. 3, were shown to test participants (observers) who scrutinised them and decided which of the two fused images, if any, better represents the input scene. Subject votes were aggregated and a preference score $\in \{0,1\}$ evaluated for each of the fused images (or equal preference). In all, subjective preferences for 120 different fused image pairs of various scenes (Figs 2,3 and 6), fused using 9 different schemes were recorded. Audiences of both expert and non-expert viewers performed the tests in controlled standard conditions [6,7-11,16]. A perceptually (subjectively) relevant fusion performance metric should be able to predict, with reasonable accuracy, the subjects' preference.

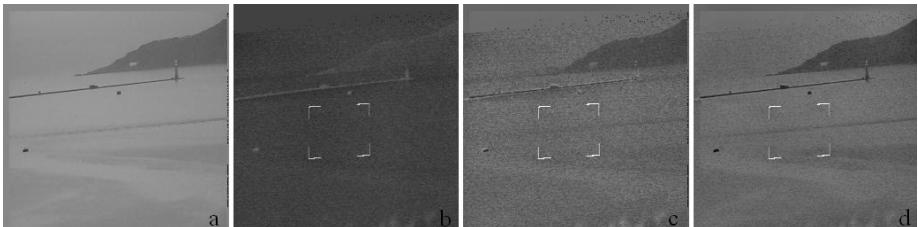


Fig. 3. Input images, a and b, and two different fused images c and d

4.1 Subjective-Objective Correspondence Evaluation

In order to measure the subjective relevance of the proposed metrics, fusion performance was evaluated for each fused image used in the tests with each metric. An objective preference was then recorded for the fused image with the higher score in each pair or, if the scores were within 1.5% of each other, an equal preference was recorded. Although similarity of 1.5% seems small, it was found to be sufficient due to the limited practical range of the values produced by the metrics [15].

From subjective and objective preference scores two distinct correspondence measures can be evaluated. The Correct Ranking (**CR**) measure is the proportion of all image pairs in which the subjective and objective ranking of offered fused images correspond. A value close to 1 (or 100%) is desirable while the minimum is the random guess rate for 3 options of 33%. The *relevance* measure (**r**) takes into account the relative certainty of the subjective scores. When the subjects are unanimous in their choice, subjective preference is 1 and so is the certainty. Alternatively, when each option receives an equal number of votes, the subjective preferences and certainty are 0.33. Relevance **r** is thus, the sum of subjective preferences of all the images with a higher objective metric score in each pair. The sum is further normalised to a range [0,1] between the smallest and largest possible relevance scores given by the subjective test results. An **r** of 1 therefore means that the metric predicted the subject's preferred choice in all image pairs. Globally, compared to **CR**, relevance **r** places more emphasis on cases where subjects are more unanimous.

4.2 Objective Metric Optimisation

Prior to testing, the VDA and VD+Q metrics were optimised with respect to the visual detection threshold T_d and the linear combination factor b respectively. The disparity of the types of images (120 input pairs) used further ensure that the optimisation process does not reach a local minima fit for a particular image type or content but a truly global optimum that is applicable to any type of input data. This optimisation process, besides determining an optimal operating point for each metric, provides a useful insight into the nature and robustness of the proposed metrics but also the behaviour and criteria used by subjects during fusion evaluation trials. The performance of VDA (in terms of **r** and **CR**) for various values of T_d is illustrated on Figure 4a. There is a clear trend in performance that improves as the detection threshold increases. For $T_d > 0.9$, performance is relatively robust while overall $\mathbf{r} > 0.73$ and $\mathbf{CR} > 0.61$. For higher values of T_d the metric considers progressively smaller areas and at $T_d = 0.95$ approximately 43% of the image area is affected by visible changes considered in the evaluation. Practically, this means that subjects form their decisions by considering relatively small proportions of the displayed images. These areas are illustrated on the fusion example from Fig. 3 in Fig. 5. The input images 3a and 3b provide two views of an outdoor scene through a visible and an infrared range sensor and are fused into two further different versions of the scene 3c and 3d. Image 3c suffers from reduced contrast and significant fusion artifacts while image 3d provides a much clearer view of the scene. Appropriately, image 3d results in fewer visible differences (white pixels) compared to both inputs (Fig. 5c and d). A high level of relevance (**r**) for this value of T_d (0.95) also indicates that the highlighted areas correspond to the areas of importance when subjects evaluate fusion.

Optimisation of the linear combination metrics, VDF+Q, $VD_{min}+Q$ and VDA+Q, with respect to coefficient b is illustrated in Fig. 4b. All three metrics have a peak in performance at $b \approx 0.7$. This indicates a greater influence of the $Q^{AB/F}$ metric and thus the appearance of gradient information in the fused image makes over general differences in the signal values they may detect.

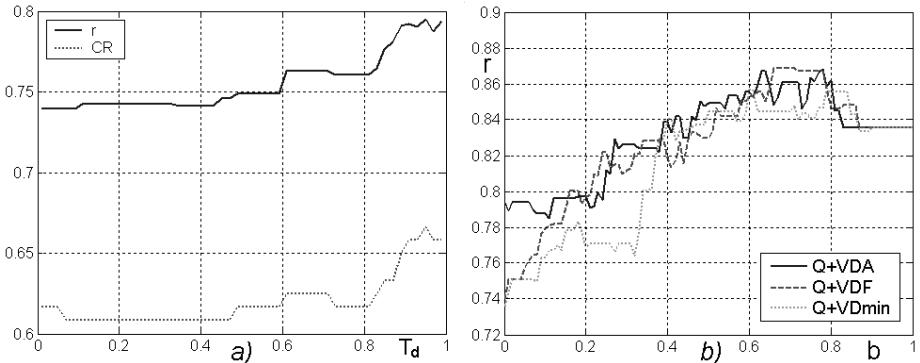


Fig. 4. Optimisation performance VDA a) and Q+VD metrics b)

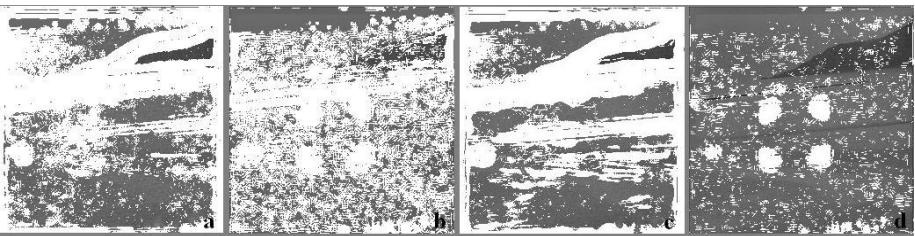


Fig. 5. VDPs between images 3a, 3b and 3c (a and b) and 3a, 3b and 3d (c and d)

4.3 Objective Metric Results

The proposed objective metrics, as well as reference metrics $Q^{\text{AB/F}}$ [15] and mutual information metric (QZY) of Qu *et. al.* [12] were tested/compared against the described set of subjective test results and yield results quoted in Table 1. The VDF and VD_{\min} metrics achieve respectable levels of success with r of 0.74 and 0.736 and CR of 61.6% and 60.8% respectively, on a par with the QZY mutual information metric. The VDA metric on the other hand performs better ($r=0.795$ and $\text{CR}=66.7\%$) proving the hypothesis that the subjects consider only sections, and not the whole of the displayed images (as in VDF and VD_{\min}) in deciding which of the offered fused images represents better the input information.

From Table 1 it is also obvious that a purely VD evaluation approach in these three metrics gives no improvement over the gradient based $Q^{\text{AB/F}}$ method. At best (VDA) VD metrics correctly rank in 80 out of the 120 fused image pairs ($\text{CR}=66.7\%$) while $Q^{\text{AB/F}}$ achieves 87. Performance is improved however, when the two approaches are combined. The hybrid Q_{VD} metric performs significantly better than the purely VD metrics and as well as $Q^{\text{AB/F}}$ ($r=0.829$, $\text{CR}=72.5\%$). The best performance however, is achieved using a linear combination of the two evaluation approaches. All three linearly combined metrics, VDF+Q, VD_{\min} +Q and VDA+Q, improve on the $Q^{\text{AB/F}}$ and Q_{VD} . VDF+Q is the best, correctly ranking 91 of the 120 pairs ($\text{CR}=75.8\%$) and achieving a relevance of $r=0.869$.

Table 1. Subjective correspondence of different image fusion performance metrics

Metric	VDP	VD _{min}	VDA	Q _{VD}	VD _{min} +Q	VDP+Q	VDA+Q	Q ^{AB/F}	QZY
r	0.740	0.736	0.795	0.829	0.855	0.869	0.868	0.833	0.742
CR	61.6%	60.8%	66.7%	72.5%	75.0%	75.8%	75.8%	72.5%	62.5%

Table 2. Fusion performance with different metrics for fused images in Figs 3 and 6

Fused Image	VDF	VD _{min}	VDA	Q _{VD}	VD _{min} +Q	VDF+Q	VDA+Q	QZY	Q ^{AB/F}
3c	0.16	0.27	0.40	0.53	0.42	0.39	0.49	1.03	0.49
3d	0.31	0.50	0.64	0.68	0.57	0.51	0.63	0.76	0.60
6c	0.43	0.82	0.57	0.77	0.78	0.66	0.69	1.79	0.76
6d	0.36	0.68	0.48	0.73	0.71	0.61	0.65	1.27	0.72

Example metric scores of the fusion example in Fig.3 are given in Table 2. Individual metric scores for the two fused images are given in Table 2. For this particular image pair, 27 out of the 28 subjects that took part in the tests opted for the image 3d. All the VD and combined metrics successfully predict this result by scoring higher for the image 3d. The exception is the QZY metric [12], based on a mutual information approach that considers histograms rather than actual image features and scores higher for image 3c. Another example, showing the case where fused images do not differ so significantly is in Figure 6. Subjects opted 9 to 4 for the fused image 6c as opposed to 6d with 2 indicating equal preference. The reason is fusion artefacts that appear as shadowing effects in image 6d and generally lower contrast compared to 6c. These effects are especially noticeable in the channel between the island and the peninsula. While VDPs between the fused images and input 6b, 6g and 6h, lend no useful information to differentiate between them, the effects of fusion artifacts are clearly visible on the VDP image between the fused image 6d and input 6a, see 6f, as white blotches indicating areas where differences are visible. In comparison the VDPs between fused image 6c and input 6a see 6e, exhibit no such effects. In terms of the numerical fusion performance scores all the metrics agree with the subjects in this case, see Table 2.

The results shown in Table 1 and Table 2 indicate that the VD approach to image fusion assessment has validity. Although the accuracy of the VD metrics is only reasonable, good scores of the hybrid Q_{VD} metric prove the usefulness of the VD information in highlighting the areas of interest to subjects during fusion assessment. The success of its compound importance evaluation algorithm indicates that subjects' attention distribution is guided by saliency in the displayed signal as well as the perceived differences in visual features that should be identical (in input and fused images). Equally, the improved performance of Q_{VD} with respect to the purely VD metrics indicates that absolute probabilities of detection of visible differences present a poorer measure of perceived information loss to the gradient approach used in Q^{AB/F} and Q_{VD}. Further justification for this view is the optimal operating point of the linearly combined metrics at $b \approx 0.7$, which indicates that the underlying Q^{AB/F} framework still provides for the majority of discrimination. At the same time, the VD

information provides an essential ingredient to making the overall fusion evaluation more accurate. Furthermore, linear combination measures are more robust than the individual VD or $Q^{AB/F}$ metrics in terms of the sensitivity to parameter values and input data types.

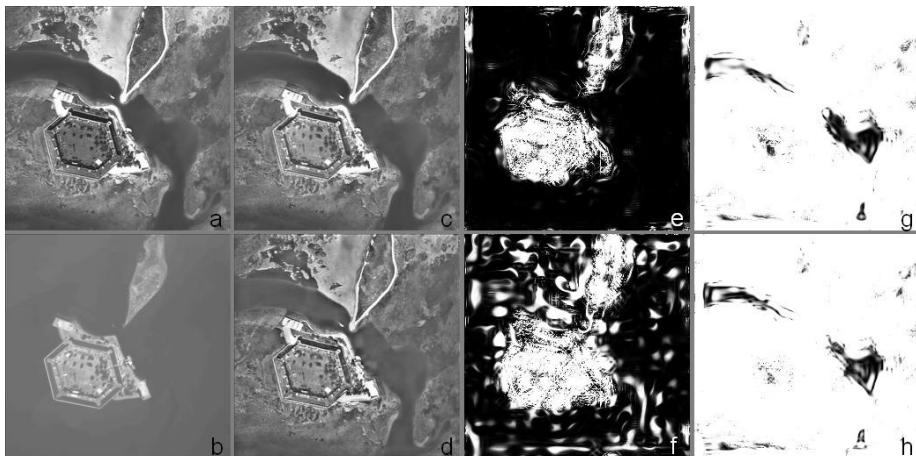


Fig. 6. Input images a and b, fused images c and d, and visible differences between c, d and a; e and f; c, d and b; g and f

5 Conclusions

This paper presented an investigation into the use of Visible Differences information in the objective assessment of image fusion performance. A Visible Differences Predictor was employed, on a pixel-by-pixel basis, to determine the probability that a human observer would notice a difference between the fused and each of the input images. Such probability maps were then integrated into an existing fusion evaluation framework or were used to form independent fusion performance metrics. Finally, the proposed metrics were tested against results obtained from real subjective fusion evaluation trials. It was found that pure VD metrics achieve respectable performance but do not improve on the best existing fusion evaluation algorithm. Hybrid measures that use both the VD and gradient evaluation approaches simultaneously however outperform all existing metrics. Such results clearly warrant further development of this ‘hybrid’ evaluation technology not only in the field of image fusion evaluation but also the more general field of image quality assessment. In the context of image fusion, the proven usefulness and dispersed nature of the VD information provides an exciting opportunity for exploration of the next step of fusion evaluation, which would itself provide a much deeper insight into different aspects of the image information fusion process, that of fusion performance characterisation.

References

1. Z Zhang, R Blum, "A Categorization of Multiscale-Decomposition-Based Image Fusion Schemes with a Performance Study for a Digital Camera Application", *Proceedings of the IEEE*, Vol. 87(8), 1999, pp1315-1326
2. O Rockinger, T Fechner, "Pixel-Level Image Fusion: The Case of Image Sequences", *Proc. SPIE*, Vol. 3374, 1998, pp 378-388
3. H Li, B Munjanath, S Mitra, "Multisensor Image Fusion Using the Wavelet Transform", *Graphical Models and Image Proc.*, Vol. 57(3), 1995, pp 235-245
4. Y Chibani, A Houacine, "Multiscale versus Multiresolution Analysis for Multisensor Image Fusion", *Proc. Eusipco98*, Rhodes, 1998
5. V Petrović, C Xydeas, "Computationally Efficient Pixel-level Image Fusion", *Proceedings of Eurofusion99*, Stratford-upon-Avon, October 1999, pp 177-184
6. A Toet, N Schoumans, J Ijspeert, "Perceptual Evaluation of Different Nighttime Imaging Modalities", *Proc. Fusion2000*, Paris, 2000, pp TuD3-17 – TuD3-23
7. A Toet, JK Ijspeert, "Perceptual evaluation of different image fusion schemes", *Proc. SPIE*, 2001, pp 436-441
8. P Steele, P Perconti, "Part task investigation of multispectral image fusion using gray scale and synthetic color night vision sensor imagery for helicopter pilotage", *Proc. SPIE*, Vol. 3062, 1997, pp 88-100
9. R Sims, M Phillips, "Target signature consistency of image data fusion alternatives", *Optical Engineering*, Vol. 36(3), 1997, pp 743-754
10. M Ulug, C McCullough, "A quantitative metric for comparison of night vision fusion algorithms", *Proc. SPIE*, Vol. 4051, 2000, pp 80-88
11. W Handee, P Wells, "The Perception of Visual Information", *Springer*, New York, 1997
12. G Qu, D Zhang, P Yan, "Information measure for performance of image fusion", *Electronic Letters*, Vol. 38(7), pp 313-315
13. F Chin, "Objective Image Quality Assessment and Spatial Adaptive JPEG Image Coding", MPhil thesis, University of Lancaster, April, 2002
14. S Daly, "The visible differences predictor: an algorithm for the assessment of image fidelity", in *Digital images and human vision*, MIT Press, Cambridge, MA, 1993
15. C Xydeas, V Petrović, "Objective Image Fusion Performance Measure", *Electronics Letters*, Vol. 36, No.4, February 2000, pp 308-309
16. ITU-R BT.500-10: "Methodology for the subjective assessment of the quality of television pictures", 1974-2000, ITU Radiocommunication Assembly

Appendix: Gradient Based Fusion Performance

Objective image fusion performance metric $Q^{AB/F}$ [15] associates important visual information with gradient information and assesses fusion by evaluating the success of gradient information transfer from the inputs to the fused image. Fusion algorithms that transfer more input gradient information into the fused image more accurately are said to perform better. Specifically, assuming two input images A and B and a resulting fused image F , a Sobel edge operator is applied to yield the strength $g(n,m)$ and orientation $a(n,m)$ ($\in [0,\pi]$) information for each input and fused image pixel. Using these parameters, relative strength and orientation "change" factors G and A , between each input and the fused image, are derived, e.g.:

$$G_{m,n}^{AF} = \begin{cases} \frac{g_F(m,n)}{g_A(m,n)}, & \text{if } g_A(m,n) > g_F(m,n) \\ \frac{g_A(m,n)}{g_F(m,n)}, & \text{otherwise} \end{cases}. \quad (5)$$

$$A_{m,n}^{AF} = \frac{\|\alpha_A(m,n) - \alpha_F(m,n)\| - \pi/2}{\pi/2}. \quad (6)$$

These factors are the basis of the edge information preservation measure Q^{AF} obtained by sigmoidal mapping of strength and orientation change factors. This quantity models the perceptual loss of input information in the fused image and constants Γ_g , κ_g , σ_g and Γ_α , κ_α , σ_α determine the exact shape of the sigmoid mappings:

$$Q_{m,n}^{AF} = \frac{\Gamma_g \Gamma_\alpha}{(1 + e^{\kappa_g(G^{AF} - \sigma_g)}) (1 + e^{\kappa_\alpha(G^{AF} - \sigma_\alpha)})}. \quad (7)$$

Total fusion performance $Q^{AB/F}$ is evaluated as a weighted sum of edge information preservation values for both input images Q^{AF} and Q^{BF} where the weights factors w^A and w^B represent perceptual importance of each input image pixel. The range is $0 \leq Q^{AB/F} \leq 1$, where 0 means complete loss of input information has occurred and $Q^{AB/F}=1$ indicates “ideal fusion” with no loss of input information. In their simplest form, the perceptual weights w^A and w^B take the values of the corresponding gradient strength parameters g_A and g_B .

$$Q^{AB/F} = \frac{\sum_{m=1}^M \sum_{n=1}^N Q_{m,n}^{AF} w_{m,n}^A + Q_{m,n}^{AF} w_{m,n}^B}{\sum_{m=1}^M \sum_{n=1}^N w_{m,n}^A + w_{m,n}^B}. \quad (8)$$

An Information-Based Measure for Grouping Quality

Erik A. Engbers¹, Michael Lindenbaum², and Arnold W.M. Smeulders¹

¹ University of Amsterdam, Amsterdam, The Netherlands,

{engbers,smeulders}@science.uva.nl

² Technion — I.I.T., Haifa, Israel,

mic@cs.technion.ac.il

Abstract. We propose a method for measuring the quality of a grouping result, based on the following observation: a better grouping result provides more information about the true, unknown grouping. The amount of information is evaluated using an automatic procedure, relying on the given hypothesized grouping, which generates (homogeneity) queries about the true grouping and answers them using an oracle. The process terminates once the queries suffice to specify the true grouping. The number of queries is a measure of the hypothesis non-informativeness. A relation between the query count and the (probabilistically characterized) uncertainty of the true grouping, is established and experimentally supported. The proposed information-based quality measure is free from arbitrary choices, uniformly treats different types of grouping errors, and does not favor any algorithm. We also found that it approximates human judgment better than other methods and gives better results when used to optimize a segmentation algorithm.

1 Introduction

The performance of vision algorithms may be considered a tradeoff between their computational cost and the quality of their results. Therefore, quality measures are essential tools in the design and tuning of algorithms, in the comparison between different algorithms and in matching algorithms to tasks. As measurement tools, quality measures should be independent of the algorithms they test. They should be free of arbitrarily set parameters, provide meaningful and useful evaluations, and preferably be consistent with human judgment.

The quality of grouping algorithms, on which we focus here, may be evaluated by either task-dependent or task-independent (generic) measures. Task-dependent advocates argue that the only way to evaluate grouping quality is by considering it in the context of some application and by using the application performance as a gauge for the grouping performance. This approach is best when working on a specific application, but it does not support modular design and does not guarantee a suitable performance for other tasks [8]. In contrast, as we know, humans can consistently discriminate between good and bad segmentations. This implies that, at least in principle, task-independent measures exist [11].

Our work is done in the context of generic empirical quality evaluation, depending on a reference grouping. (This is opposed to alternatives such as *Analytic* performance evaluation [20,1,2] or empirical evaluation without a reference [12].)

Existing generic grouping quality measures rely on some kind of set difference measure which specifies a (dis-)similarity between the evaluated grouping and a reference true grouping (see e.g. [10,18,11]). Quality may be evaluated by, say, counting the number of incorrectly assigned pixels (additions and/or deletions), by counting the number of true groups which split or merge, or by measuring Hausdorff distances between the segments. Such measures are indeed indicative of the segmentation correctness, but the preference of one similarity measure over the other is arbitrary. One approach to addressing this confusion is to consider the tradeoff between several different measures of quality [7]. Another problem is that such similarity measures are not in complete agreement with intuitive judgment [4]. A hierarchical ground truth with multiple options substantially increases the agreement with intuition. Still, for every one of these options, the measure is still arbitrarily selected [11]. The lack of a suitable generic quality measure seems to be the main reason that subjective judgment is still the most common way for evaluating grouping results.

Unlike similarity-based approaches, we consider the grouping hypothesis as an initial guess which provides information on the unknown true grouping and reduces its uncertainty. A better grouping result provides more information about the true grouping. The amount of information may be measured in two alternative but related approaches:

Uncertainty view - Without a grouping hypothesis, the uncertainty about the true grouping is high and the number of possible true groupings is large.

Knowing the grouping hypothesis reduces the number of correct grouping possibilities, or at least reduces the likelihood of some of them. Quantification of the uncertainty reduction is thus a measure of quality.

Effort view - Suppose the true group is specified by a sequence of basic measurements, such as the cues used by grouping algorithms. The length of the minimal sequence is the effort required for specifying the correct grouping. A given grouping reveals information about the unknown correct grouping and reduces the effort. Quantification of effort reduction is a measure of the hypothesis quality.

These two ways for evaluating the quality are related. When the uncertainty is large, more effort is needed to find the correct hypothesis. In information theory, a complete probabilistic characterization allows us to specify a tight relation between the code length (effort) and the entropy (uncertainty). Here, the relation exists but is not as tight. Therefore, we emphasize the effort based approach which provides a practical and intuitive procedure for grouping quality evaluation.

Similar relations between ‘effort’ and ‘quality’ form the basis of quality measures in other domains such as the earth movers distance [15] and the string edit distance [13]. In the context of image segmentation, this relation is explicit in

[17], where segmentation results are evaluated by measuring the effort it takes to manually edit the hypothesis into the correct solution. The resulting evaluation measure is the weighted sum of the performed actions in the editing phase. Our proposed method may be considered to be an automatic version of this subjective, manual approach.

We also propose a probabilistic characterization of the grouping process and show that considering the given grouping result to be a random variable gives a precise meaning to the uncertainty of the true grouping, using terms such as surprise and entropy, as defined in information theory [5]. We show how these terms are related to the quantification of quality using effort.

The proposed approach has the following advantages:

General, uniform and fair - The measure uniformly deals with various types of grouping mistakes, does not involve ad-hoc decisions or parameters and is not biased towards any particular method.

Consistent with HVS - The measure is more consistent with human judgment than other methods.

Meaningful - The measure may be interpreted by known statistical terms (from information theory).

Useful - The quality measure is practical, useful, and allows, for example the optimization of the parameters of a grouping algorithm, in a better way.

After some necessary definitions (Section 2), the proposed quality measure is presented in Section 3. A link to the uncertainty view is made in Section 4. Section 5 reports on some experiments, including psychophysics. The paper is concluded with some observations and directions for future work in Section 6.

2 Preliminaries

Let $S = \{s_1, s_2, \dots, s_n\}$ be a set of elements, and consider a grouping C of S as the partition of S into disjoint subsets, $C = \{X_1, X_2, \dots, X_m\}$, with $X_i \subseteq S$, $\cup_i X_i = S$, $X_i \cap X_j = \emptyset$ for $i \neq j$. The set of all possible groupings of set S is defined as $\mathcal{C} = \{C | C \text{ is a grouping of } S\}$. A useful grouping may not be disjoint or even unique. Here, however, we take a simplified approach and further assume that there is only one correct grouping, denoted $C_T \in \mathcal{C}$. It is straightforward to generalize the proposed measure to handle non-unique correct groupings.

A grouping algorithm provides a grouping hypothesis C_H , which is a partition as well. Usually it will not be identical to the true grouping C_T . The goal of this paper is to propose a measure, denoted $Q_T(C_H)$, for the quality of the grouping hypothesis C_H relative to a known true grouping C_T .

3 Judging Grouping Quality from a Questions Game

To quantify the effort required to get the correct grouping from the grouping hypothesis, we consider a generic procedure that asks questions about the true

unknown grouping. These questions are answered by an oracle, which relies on the true grouping and provides only “Yes/No” type answers. After a sufficient number of questions are asked, the true grouping may be inferred. The questioning procedure builds on the grouping hypothesis, and the grouping hypothesis is considered better if the number of questions, or the *effort*, is lower.

Note that this view of quality is closely related to the parlor game of “Twenty Questions” [16], where one needs to ask a minimal number of questions in order to identify a secret object. (See also [14] where such a query mechanism is used for visual object recognition.) In this context, the value of a ‘hint’ may be measured by the number of questions it can save. Correspondingly, the grouping hypothesis is considered a hint of the true grouping, and its quality is the number of queries it saves. To evaluate the quality of C_H , the following needs to be specified:

- an oracle that knows the correct grouping,
- a systematic questioning strategy, and
- a set of questions from which a subset is drawn according to the strategy.

3.1 Homogeneity Queries

The type of questions, or queries, we allow are *homogeneity queries* [19,10]. That is, every query specifies a subset of the image and asks whether all picture elements in it belong to the same group. The oracle, knowing the correct grouping, can easily answer such questions. Adopting another query type (e.g. a Boolean function over several homogeneity queries) could lower the number of queries, but we conjecture that it would not change the relative number substantially, implying that it would not be better for *comparing* grouping hypotheses. Our experiments (Section 5.3) support this argument.

3.2 Questioning Strategies

A questioning strategy suitable for the proposed quality evaluation should have two main properties. It should not be biased toward specific types of grouping results and should be efficient in the sense of not asking more questions than necessary. An optimal strategy, asking the minimal (average) number of questions would be best, and could be designed, at least in principle, from a probabilistic model (described in the next section).

Here, however, we have chosen a non-optimal strategy which is based on a split-and-merge search for the true grouping (see Section 5). We conjecture that such a strategy provides query counts which are proportional to those achieved with optimal strategies, and thus is good enough for estimating relative qualities. The experimental results, described in Section 5.3, support this conjecture.

3.3 A Normalized Quality Measure

Let $N_q(C_H, C_T)$ be the number of queries required to know C_T from the hypothesis C_H . One possible normalized quality measure is

$$Q_T(C_H) = \frac{N_q(C_W, C_T) - N_q(C_H, C_T)}{N_q(C_W, C_T) - N_q(C_T, C_T)}, \quad (1)$$

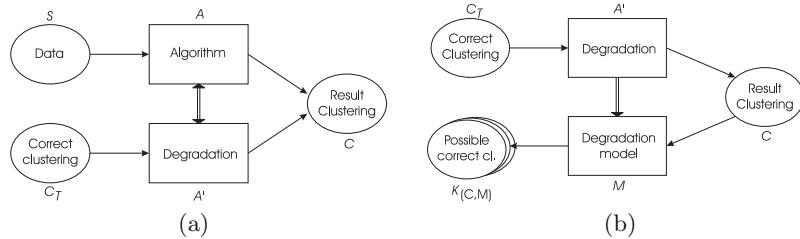


Fig. 1. Two alternative views of a grouping (clustering) process: as an algorithm that labels the data elements or as a degradation process from the true grouping to the grouping result (a), and construction of the set of all possible correct grouping (clusterings) given C and M (b).

which is maximal and 1 for the best (true) hypothesis ($C_H = C_T$) and is minimal and zero for the worst hypothesis C_W , specified as the one requiring the maximal number of questions. While this normalization is intuitive, other normalizations are possible, and may be preferable; see [6]. Here we focus on the raw quantity $N_q(C_H, C_T)$.

4 A Statistical Notion of Grouping Quality

4.1 A Statistical Model for Grouping

Grouping algorithms significantly differ in the processes they use. A quality measure, as any objective measurement tool, needs to consider only the results of the algorithms in a uniform way, using a common language, unrelated to the process carried out by the algorithm. Therefore, we consider the grouping hypothesis, provided by an algorithm A , to *also* be a result of an equivalent process, called *degradation*, denoted by $A' : \mathcal{C} \rightarrow \mathcal{C}$. This process, operating on groupings (and not on images) receives the correct grouping C_T as an input and provides that same hypothesis $C_H = A'(C_T)$ that the algorithm A delivers (see Figure 1).

The degradation process takes into account both the algorithm and the image given to it, as both of them influence the grouping hypothesis. It may be modeled stochastically as follows:

Stochastic Degradation Model:

A degradation process A' from C_T to $C = A'(C_T)$ is an instance of a random variable M drawn using some probability distribution $P_M(A')$.

The random variable M is denoted a *degradation model*. An instance of this random variable is a particular *degradation process*, which is equivalent to the action of a particular grouping algorithm on a particular image; see a more detailed modeling in [6].

4.2 The Posterior Distribution of Groupings

For a given grouping hypothesis C_H , true grouping C_T , degradation model M , and some prior distribution $P^*(C)$ over the set of true groupings, the set of all possible correct groupings, $K_{(C_H, M)}$, which are consistent with the grouping hypothesis C_H , can be constructed as (see Figure 1):

$$K_{(C_H, M)} = \{C | C \in \mathcal{C}, P_M(A') > 0, A'(C) = C_H\}. \quad (2)$$

Taking a Bayesian approach, the posterior probability that a particular grouping $C \in K_{(C_H, M)}$ is the true grouping, is

$$P_{(C_H, M)}(C) = \frac{\sum_{\{A' | A'(C) = C_H\}} P_M(A') P^*(C)}{\sum_{C' \in K_{(C_H, M)}} \sum_{\{A' | A'(C') = C_H\}} P_M(A') P^*(C')}. \quad (3)$$

4.3 The Quality of a Grouping Hypothesis as a Surprise

Given the posterior probability distribution on the possible true groupings, one can construct an optimal questioning strategy, following the corresponding Huffman code [5]. It is well known that the average length of the Huffman code converges (from above) to the entropy and is thus minimal¹.

The following related result is even more useful here: the number of bits required to code a message, associated with probability p , using a Huffman code, is never larger than $\lceil -\log_2 p \rceil$ [5]. The quantity $-\log_2 p$ is sometimes called *surprise* [9], in accordance with intuition: a rare message, associated with a small probability p , makes a large surprise when it appears.

If the query strategy is designed according to the Huffman code, it follows that the number of queries is not higher than the surprise associated with the event (of probability $P_{(C_H, M)}(C = C_T)$) that, given C_H , C_T is the true grouping. The quality measure Q_T (1) is monotonic in the query count. Therefore,

Claim 1 (Hypothesis Quality and Surprise) *Under the probabilistic degradation model, and with an optimal questioning strategy and unlimited queries, the proposed hypothesis quality measure Q_T is a monotonically non-increasing function of $-\log_2 P_{(C_H, M)}(C = C_T)$ - the surprise associated with this grouping.*

This relation attaches a new meaning to the proposed quality measure. A larger number of queries implies that the true grouping is more “surprising” (in the information theory sense), meaning that the hypothesis is less informative and worse.

A subtle issue is the choice of the degradation model and the corresponding questioning strategy. A model adapted to a particular algorithm assigns a smaller surprise to grouping errors which are typical to it and thus introduces bias. Therefore, when comparing different hypotheses associated with different algorithm, a degradation characterizing the ensemble of grouping algorithms is a better choice.

¹ $H(X) = -\sum_{x \in X} P(x) \log_2 P(x)$ is the entropy of the random variable X .

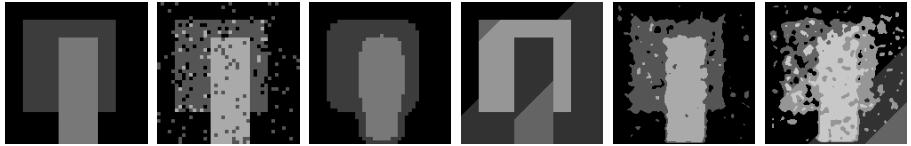


Fig. 2. The original true grouping, the three types of basic degradation: noise type, attached type, and split/merge type, and two examples of mixture degradation: noise+attached, and noise+attached+split (left to right).

4.4 The Quality of a Grouping Algorithm as (Average) Entropy

So far, we considered the quality of a single grouping hypothesis C_H . To evaluate an algorithm, we propose to average the number of queries required for a large number of grouping tasks. In our model, this average number converges to the average entropy

$$\overline{H(A)} = \sum_{C \in \mathcal{C}} H(P_{(C,M)}) \text{Prob}(C_H = C), \quad (4)$$

which depends only on the degradation model M and on the prior distribution of true groupings. For a detailed discussion see [6].

The elegant relation between effort and surprise (and between average effort and average entropy) holds rigorously only in the ideal case. When examining a grouping hypothesis we usually do not know the degradation model, and consequently, cannot design the optimal Huffman strategy. Moreover, the *homogeneity queries* we propose are weaker than the arbitrary queries required to specify the Huffman tree. While we do not have a provable relation, we conjecture that the query count is monotonic in the surprise and use the experimental effort as a quality measure. Some experiments, described in Section 5.3, show that this conjecture is a justified approximation.

5 Experiments

The experiments described below illustrate how the measure quantifies common types of grouping errors, show its improved agreement with human judgment and a relation between measured quality and entropy estimates, and demonstrate the proposed measure utility for optimizing a simple grouping algorithm.

5.1 Some Illustrative Quality Measurements

Data. The grouping errors considered in the first set of experiments are mixtures of three basic degradation types, illustrated in Figure 2: *noise type (independent errors)*, characterized by isolated “islands” of incorrect grouping and rough group boundaries, *attached errors*, where the errors are near the real groups boundaries and have relatively smooth shapes, and *split-and-merge errors* where large

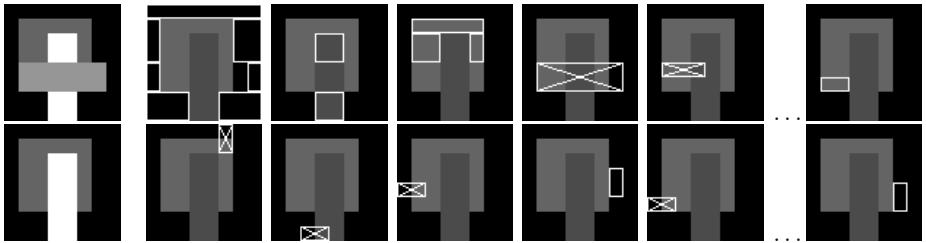


Fig. 3. An illustration of a query sequence: the two images on the left show the hypothesis (top) and the true grouping (bottom). The rest of the images describe typical queries starting from the first (top - second left). The top row shows queries from the split stage. The first three queries are done on a homogeneous region. The fourth is done on a non-homogeneous region and consequently this region is split. Queries associated with the merge stage are described in the bottom line. In every image, the white boundary rectangle (or union of such rectangles) marks the region tested for homogeneity, and a cross over this region implies that the homogeneity test failed.

parts of the original groups are split and merged. Using synthetic grouping error generators allows us to examine the quality measures for the different types of grouping errors. Moreover, this way, the measure is not developed in the context of a particular algorithm and is not biased towards its properties.

The questioning strategy. A sequence of homogeneity queries, based on the *given grouping hypothesis*, is used to find the *true grouping*. The strategy recursively splits the groups specified by the hypothesis until the subparts are homogeneous. This is done hierarchically, according to a binary regions tree, which is built for every hypothesized group. Then, the unions of the subparts are tested, so that no two (adjacent) parts which belong to the same group remain separate. See [6] for a more detailed and formal description and Figure 3 for an illustration.

The quality calculator procedure is available to the community via a web site www.cs.technion.ac.il/Labs/Isl/Research/grouping-quality/gq.htm allowing the uploading of the user's images.

The query count measure for a variety of grouping degradations. For every tested grouping, we consider two quality measures: the one we propose, and a reference one, denoted the difference measure, which is simply the minimal number of hypothesis pixels that should be corrected to get the true grouping.

First we considered hypotheses, degraded by different amounts of noise-based errors. Naturally, more erroneous hypotheses required more queries and have higher difference measures. Next we addressed a more complex degradation, which is a mixture of attached type and noise type errors. Here the number of queries significantly decreases when the attached type error is more dominant,

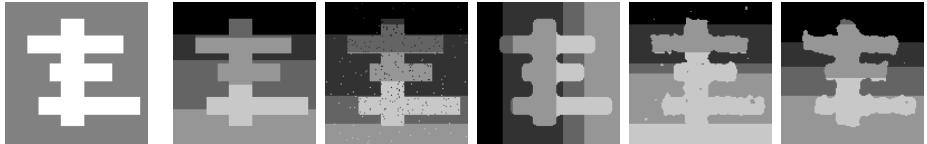


Fig. 4. The dependency of the query count on the error type. The five segmentations above differ in the source of the errors. The left most image is the true segmentation. The second left image is an hypothesis getting all its errors from splits and merges, while as we move to the right, the influence of attached errors and noise is stronger. The number of queries required for these hypotheses are 51, 623, 593, 914 and 1368. (Corresponding quality values (Q_T) are 0.987, 0.87, 0.84, 0.76, and 0.64.) The corresponding difference measures are 10601, 10243, 10001, 9946 and 9916. Note that even though the difference measure stays the same or even decreases, the number of queries significantly increases.



Fig. 5. The query count measures for some hand segmented images. The left-most image is assumed to be “the true one” and the rest are the other hand segmentations ordered (left to right) by increasing query count (317, 406, 548, 566 and 678).

even though the difference measure stays the same. The measures are markedly different when split/merge grouping mistakes are dominant. Then, in agreement with intuition, the proposed measure does not penalize every pixel of the split part as if the grouping of these pixels were completely incorrect. The number of queries required to restore a grouping dominated by this type of grouping errors is much lower than the number required for noise or attached type grouping errors; see Figure 4 for an example, and [6] for details regarding all experiments.

5.2 Comparison of Hand Segmented Images

While we still think that the characteristics of the proposed measure are best revealed using controllable grouping errors, we tested the measure using examples from the Berkeley database. We took several segmentations of the same image, chose one of them as the “true one”, and examined the information revealed on it from the others. The query count may be sensitive to small scale errors which, for some applications, should not make a difference. To reduce this dependence, we can treat thin regions of non-homogeneity as “don’t care” in the homogeneity query. See Figure 5 for an example demonstrating the good agreement with human judgment.

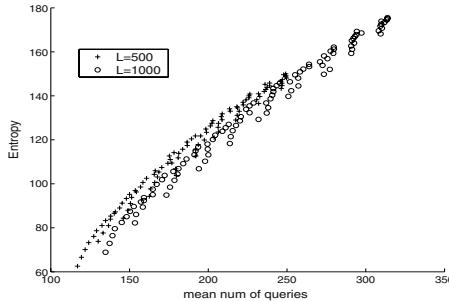


Fig. 6. The plot describes the combinatorially calculated entropy against the average query count, for two image sizes ($L = 500, 1000$), $k = 30$ true groups, and a variety of degradations. (All combinations of s splits and m merges where $s, m \in [6, 8, 10, \dots, 24]$).

5.3 Average Query Count versus Entropy Experiment

The validity of the probabilistic interpretation is tested by relating the average query count to the entropy. Picking a relatively simple example allows us to analytically approximate the entropy. We consider 1D “images” of length L . These images are truly divided into several (k) true groups, but the given hypotheses results from a degradation process in the form of m merges and s splits. (The splits do not split between true groups and the merges do not merge parts which were split in the degradation). For a given hypothesis, the number of feasible true groupings is $\binom{k+s-m-1}{s} \binom{L-k-s+m}{m}$. The logarithm of this number is the entropy (under uniform probabilistic assumptions).

A 1D variant of the questioning strategy, unaware of the parameters k, m and s , is used to find the query count measure. The average number of queries is estimated for every parameter set L, k, m, s , from several randomizations of the true groupings and degradations corresponding to this parameter set.

The combinatorially calculated entropy is plotted, in Figure 6, against the average query count. The different selections for s and m specify a wide variety of different grouping errors. Still, the relation between the query count and the entropy is almost linear, and independent of the error type. This relation is consistent with our interpretation, and supports the claim that ideally, the average query count is the (average) entropy, and (more weakly), the claim that the query count associated with one hypothesis is proportional to its surprise. More than one query is required for every bit of uncertainty, which is expected from the weaker homogeneity queries. An important practical implication is that the number of queries required by our heuristic strategy, being almost linear in the number of queries required by the ideal strategy for a variety of grouping errors, is an unbiased measure for comparing grouping results.

5.4 An Application – Algorithm Tuning

To show its utility, we used the query count to tune a particular segmentation (binarization) algorithm. This algorithm smooths a given image, extracts edge

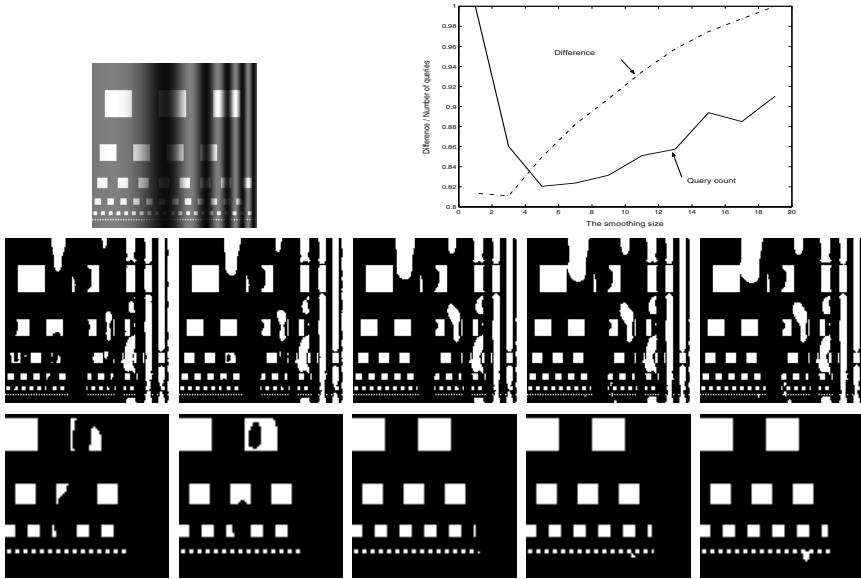


Fig. 7. The given image (top left), 5 binarized images (grouping hypotheses), corresponding to smoothing widths of 1, 3, 5, 7 and 9 (left to right, middle and bottom (zoomed in view)), and the query count and the difference measures plotted against the smoothing kernel width (top right, the measures are normalized). The binarization specified by the minimal query count is indeed visually optimal.

points and uses them to set a binarization threshold surface [3]. The degree of smoothing is a free parameter of the algorithm which we optimized; see Figure 7 for the results. Note that minimizing the query count (minimal at smoothing width = 5) leads to the visually most pleasing result, while relying on the difference measure (minimal at smoothing width = 3) does not.

5.5 Psychophysics

We tested the consistency of the proposed measure with human judgment in a simple psychophysical experiment. A subject was shown a sequence of slides, each containing two pairs of grouping images, and was asked to tell which pair shows a more similar presence of objects. The pairs were of very similar groupings, one different from the other by a mixture of attached type and noise type errors. The difference measure was 1-2% for all pairs.

An answer was considered correct if it was in agreement with the query count based preference. The number of incorrect answers (an average over five subjects) is plotted against the difference in quantized query counts (Figure 8). (The query counts of all pairs were quantized into 7, equally populated levels. For every difference value, 10 slides were presented.) The error rate sharply decreases for a higher difference in query count, showing an excellent agreement between

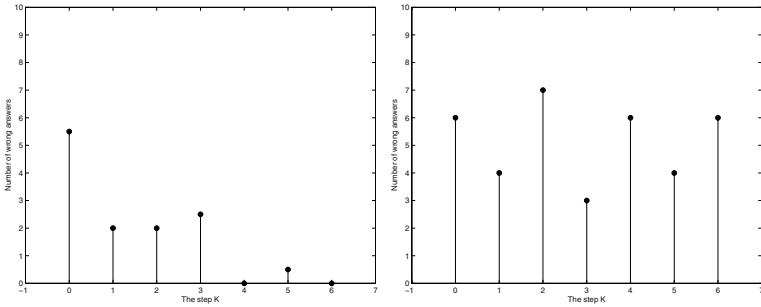


Fig. 8. The rate of incorrect answers as a function of ranking according to query based measure (left) and to additive difference based measure (right)

the query count based measure and human judgment. Repeating this experiment for the difference measure shows little, if not zero, agreement (Figure 8).

6 Conclusions

We proposed a generic grouping quality measure, quantifying the information available from the grouping result on the true grouping. A (simple and automatic) simulated interaction between a questions strategy and an oracle was used to estimate this information. We found that the proposed measure more closely approximates human judgment than other methods and as such gives better results when used to optimize a segmentation algorithm. The proposed methods is associated with the following two main advantages:

Generality and fairness - Most previous, similarity-based measures, involve unavoidable arbitrary choices. The proposed information-based quality measure is free from such arbitrary choices, treats different types of grouping errors in a uniform way and does not favor any algorithm.

Non-heuristic justification - The number of queries is interpreted as a *surprise* in an information theory context. While the questioning strategy is not ideal, the query count was found to be approximately monotonic in the entropy, independent of the grouping error type, indicating both that this interpretation is valid and that the query count is an adequate unbiased means for comparing grouping results.

This work was done in the context of a unique ground truth. One future direction would be to generalize our measure to multiple ground truths (as was shown to be more meaningful in [11]). This could be done by finding the query count for all ground truths and calculating the quality from the minimal value.

Acknowledgments. M.L. would like to thank Ronny Roth and Neri Merhav for discussions on information theory and Leonid Glykin for his help in the experiments.

References

1. A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(2):168–185, 1998.
2. A. Berengots and M. Lindenbaum. On the performance of connected components grouping. *Int. J. of Computer. Vision*, 41(3):195–216, 2001.
3. I. Blayvas, A. Bruckstein, and R. Kimmel. Efficient computation of adaptive threshold surfaces for image binarization. In *CVPR01*, pages I:737–742, 2001.
4. A. Cavallaro, E.D. Gelasca, and T. Ebrahimi. Objective evaluation of segmentation quality using spatio-temporal context. In *ICIP02*, pages III: 301–304, 2002.
5. T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley series in telecommunications. Wiley-Interscience, 1991.
6. E.A. Engbers, M. Lindenbaum, and A.W.M. Smeulders. An information based measure for grouping quality. Technical Report CIS-2003-04, CS dept., Technion, 2003.
7. M. Everingham, H. Muller, and B. Thomas. Evaluating image segmentation algorithms using the pareto front. In *ECCV02*, page IV: 34 ff., 2002.
8. W. Foerstner. 10 pros and cons against performance characterization of vision algorithms. In *Performance Characteristics of Vision Algorithms*, Cambridge, 1996.
9. G.D. Forney. Information theory. unpublished lecture notes, EE dept. Stanford university.
10. D.G. Lowe. *Perceptual Organisation and Visual Recognition*. Kluwer Academic Publishers, 1985.
11. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV01*, pages II: 416–423, 2001.
12. P. Meer, B. Matei, and K. Cho. Input guided performance evaluation. In R. Klette, H.S. Stiehl, M. Viergever, and K.L. Vincken, editors, *Performance Characterization in Computer Vision*, pages 115–124. Kluwer, Amsterdam, 2000.
13. S. V. Rice, Horst Bunke, and T. A. Nartker. Classes of cost functions for string edit distance. *Algorithmica*, 18(2):271–280, 1997.
14. W. Richards and A. Bobick. Playing twenty questions with nature. In Z. Pylyshin, editor, *Computational Processes in Computer Viosion: An interdisciplinry Perspective*, chapter 1, pages 3–26. Ablex, Norwood, 1988.
15. Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Proc. 6th ICCV IEEE Int. Conf. on Computer Vision*, pages 59–66, 1998.
16. C.E. Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 30:50–64, 1951.
17. K.L. Vincken, A.S.E. Koster, C.N. De Graaf, and M.A. Viergever. Model-based evaluation of image segmentation methods. In *Performance Characterization in Computer Vision*, pages 299–311. Kluwer Academic Publishers, 2000.
18. L. Williams and K.Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. In *ECCV98*, 1998.
19. A. Witkin and J. Tenenbaum. On the role of structure in vision. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, pages 481–543. Academic Press, 1983.
20. Y.J. Zhang. A survey on evaluation methods for image segmentation. *PR*, 29(8):1335–1346, August 1996.

Bias in Shape Estimation

Hui Ji and Cornelia Fermüller

Center for Automation Research
University of Maryland
College Park, MD 20742-3275, USA
`{jihui,fer}@cfar.umd.edu`

Abstract. This paper analyses the uncertainty in the estimation of shape from motion and stereo. It is shown that there are computational limitations of a statistical nature that previously have not been recognized. Because there is noise in all the input parameters, we cannot avoid bias. The analysis rests on a new constraint which relates image lines and rotation to shape. Because the human visual system has to cope with bias as well, it makes errors. This explains the underestimation of slant found in computational and psychophysical experiments, and demonstrated here for an illusory display. We discuss properties of the best known estimators with regard to the problem, as well as possible avenues for visual systems to deal with the bias.

1 Introduction

At the apogee of visual recovery are shape models of the scene. Cues such as motion, texture, shading, and contours encode information about the scene surfaces. By inverting the image formation process (optical and geometrical) it is possible to recover three dimensional information about the scene in view. However, despite tremendous progress there are still many difficulties. For the case of reconstruction of structure from multiple views, and even when the 3D viewing geometry is estimated correctly, the shape often is incorrect.

Why? Is there some fundamental reason that this happens, or is it due to the inadequacy and lack of sophistication of our computational models? The literature in psychophysics reports that humans also experience difficulties in computing 3D shape and this has been demonstrated by many experiments. For a variety of conditions and from a number of cues the mis-estimation is an underestimation of slant. For example, planar surface patches estimated from texture [6], contour, stereopsis, and motion of various parameters [11] have been found to be estimated with smaller slant, that is, closer in orientation to a front-parallel plane than they actual are.

In this paper we investigate the problem of 3D shape estimation from multiple views (motion and stereo). We show that there exist inherent computational limitations. These result from the well known statistical dilemma. Shape recovery processes are estimation processes. But because there is noise in the image data, and because the complexity of the visual computations does not allow to

accurately estimate the noise parameters, there is bias in the estimation. Thus, we find, that one of the reasons for inaccuracy in shape estimation, is systematic error, i.e. bias.

The paper accomplishes three things. (a) We introduce a new constraint for shape from multiple views (motion and stereo) which relates shape and rotation to image lines. Why image lines (i.e. edges)? Because shape (i.e. surface normals) change only with rotation and there is a natural way to deal with the rotation of projected lines. The new constraint makes it possible to: (b) provide a statistical analysis of shape from motion, which reveals an underestimation of slant as experienced by humans and by most programs. An understanding of the bias allows us to create displays that give rise to illusory erroneous depth perception. Since we understand the parameters involved in the bias we can set them such that the bias is very large causing mis-perception. (c) We discuss and implement the statistical procedures which are best for shape estimation. We found that we only can slightly reduce the bias. The theoretically best thing to do is to partially correct for the bias. We then suggest how we may do better in structure from motion.

2 Overview and the Main Concepts

The idea underlying the statistical analysis is simple. The constraints in the recovery of shape can be formulated as linear equations in the unknown parameters. Thus the problem is reduced to finding the “best” solution to an over-determined equation system of the form $A'u' = b'$ where $A' \in R^{N \times K}$ and $b' \in R^{N \times 1}$ and $N \geq K$. The observations A' and b' are always corrupted by the errors, and in addition there is system error. We are dealing with what is called the errors-in-variable (EIV) model in statistical regression, which is defined as:

Definition 1. (*Errors-In-Variable Model*)

$$\begin{aligned} b &= Au + \epsilon \\ b' &= b + \delta_b \\ A' &= A + \delta_A \end{aligned}$$

u are the true but unknown parameters. A' and b' are observations of the true but unknown values A and b . δ_A, δ_b are the measurement errors and ϵ is the system error which exists if A and b are not perfectly related.

The most common choice to solving the system is by means of LS (least squares) estimation. However, it is well known, that the LS estimator u_l , whose solution is characterized by $u_l = (A'^T A')^{-1} A'^T b'$, generally is biased [12].

Consider the simple case where all elements in δ_A and δ_b are i.i.d random variables with zero mean and variance σ^2 . Then

$$\lim_{n \rightarrow \infty} E(u_l - u) = -\sigma^2 \left(\lim_{n \rightarrow \infty} \left(\frac{1}{n} A^T A \right) \right)^{-1} u, \quad (1)$$

which implies that u_l is asymptotically biased. Large variance in δ_A , ill-conditioned A or an u which is oriented close to the eigenvector of the smallest

singular value of A all could increase the bias and push the LS solution u_l away from the real solution. Generally it leads to an underestimation of the parameters.

Using the bias from least squares we analyze in Sections 3 the estimation of shape from motion and extend the analysis in Section 4 to stereo. Section 5 describes other properties of other estimators with respect to our problem. Section 6 shows experiments, and Section 7 summarizes the study and discusses possible avenues to deal with the bias.

Some previous studies analysed the statistics of visual processes. In particular, bias was shown for 2D feature estimation [3] and optical flow [10].

In [8] bias was discussed for a number of visual recovery processes, and some studies analysed the statistics of structure from motion [1,2]. However, these analyses stayed at the general level of parameter estimation; no one has shown before the effects on the estimated shape.

Shape from motion, or in general shape from multiple views is an active research area, and many research groups are involved in extracting 3D models on the basis of multiple view geometry [7]. The theory proceeds by first solving for camera geometry (where are the cameras?). After the cameras are placed, the structure of the scene is obtained by extending the lines from the camera centers to corresponding features; their intersections provide points in 3D space which make up a 3D model.

Thus, the structure of the scene requires both the translation and the rotation between views. But the structure can be viewed as consisting of two components: (a) the shape, i.e. the normals to the surface and (b) the (scaled) depth. It is quite simple to show that shape depends only on the rotation between two views, while depth depends also on the translation. This new constraint, which is explained next, allows us to perform an uncertainty analysis of the estimation of shape from motion and deduce the underestimation of slant.

3 Shape from Motion

3.1 Formulation of the Constraint

Consider the scene to be a textured plane with surface normal n . The texture is described by the lines on the plane. A line L in 3d space is described by Plücker coordinates $L = (L_d, L_m)$, where

$$\begin{cases} L_d = P_1 - P_2; \\ L_m = L_d \times P = P_2 \times P_1. \end{cases}$$

for any points P, P_1, P_2 on the line. L_d denotes the direction of the line in space, and L_m its moment. L_d and L_m are perpendicular, that is $L_d \cdot L_m = 0$. The projection of the 3D line L on the image is just L_m and normalized to have the third coordinate 1, it is:

$$\ell = \frac{1}{\hat{z} \cdot L_m} L_m.$$

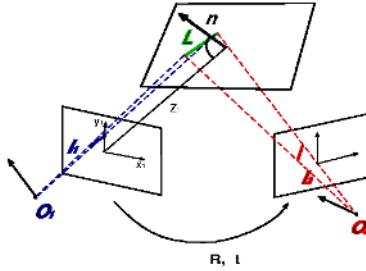


Fig. 1. Two views of a planar patch containing a line.

Let us first describe the intuition: The camera undergoes a rigid motion described by a translation vector T and a rotation matrix R , as shown in Fig. 1. Let subscripts 1 and 2 denote quantities at time instances t_1 and t_2 . Using projective coordinates l_1 and l_2 represent the normals to the planes defined by the camera centers (O_1 and O_2) and the projections of a line L in space on the images. L is the intersection of these two planes. Thus the crossproduct of l_1 and l_2 is parallel to L_d . Writing this equation in the first coordinate system, we obtain

$$l_1 \times R^T l_2 = k L_d \quad (2)$$

with k a scalar. (Note: We can find using rotation only a line parallel to L , but not the line L . To find L we need translation.) The immediate consequence is that from two corresponding lines in two views we can find the shape of the patch containing the lines. Since L_d is perpendicular to the surface normal, n , we obtain

$$(l_1 \times R^T l_2) \cdot n = 0.$$

We model here differential motion, that is a point in space has velocity $\dot{P} = t + \omega \times P$, in which case we have that

$$\begin{cases} \dot{L}_d = \dot{P}_1 - \dot{P}_2 = \omega \times (P_1 - P_2) = \omega \times L_d \\ \dot{L}_m = \dot{P}_2 \times P_1 + P_2 \times \dot{P}_1 = t \times L_d + \omega \times L_m \end{cases}$$

Hence

$$\dot{\ell} = \frac{\dot{L}_m}{\hat{z} L_m} - \frac{\hat{z} \dot{L}_m}{\hat{z} L_m} \frac{L_m}{\hat{z} L_m} = \frac{1}{\hat{z} L_m} t \times L_d + \omega \times \ell + \frac{\hat{z} \dot{L}_m}{\hat{z} L_m} \ell,$$

and the constraint in (2) takes the form

$$\ell \times (\dot{\ell} - \omega \times \ell) = \frac{t \cdot \ell}{\hat{z} L_m} L_d. \quad (3)$$

Thus, if the 3D line is on the plane with normal vector n , its image ℓ must obey the following constraint

$$n \cdot (\ell \times (\dot{\ell} - \omega \times \ell)) = 0 \quad \text{or} \quad (4)$$

$$n \cdot e = 0 \quad (5)$$

with $e = (\ell \times (\dot{\ell} - \omega \times \ell))$

3.2 Error Analysis

Let $n = (n_1, n_2, 1)$ be the surface normal, and let $\{\ell_i = (a_i, b_i, 1)\}$ denote the lines on the plane, and $\{\dot{\ell}_i = (\dot{a}_i, \dot{b}_i, 0)\}$ denote the motion parameters of the lines ℓ_i . We estimate the orientation of the plane using LS estimation.

From (5) we know, that n in the ideal case should satisfy equation,

$$(e_{1i}, e_{2i}) \cdot \begin{pmatrix} n_1 \\ n_2 \end{pmatrix} = -e_{3i}, \quad (6)$$

where

$$\begin{cases} e_{1i} = -\dot{b}_i + (-1 + b_i^2)\omega_1 + a_i b_i \omega_2 + a_i \omega_3 \\ e_{2i} = \dot{a}_i + (a_i b_i \omega_1 - (1 + a_i^2)\omega_2 + b_i \omega_3) \\ e_{3i} = (\dot{a}_i b_i - \dot{b}_i a_i) + (a_i \omega_1 + b_i \omega_2 - (a_i^2 + b_i^2)\omega_3). \end{cases}$$

There is noise in the measurements of the line locations and the measurements of line motion. For simplicity of notation, let us ignore here the error in the estimates of the rotation parameters. Throughout the paper let primed letters denote estimates, unprimed letters denote real values, and δ 's denote the errors. That is, $\delta\dot{a}_i = \dot{a}'_i - \dot{a}_i$ and $\delta\dot{b}_i = \dot{b}'_i - \dot{b}_i$ with expected value 0, variance δ_1^2 ; $\delta a_i = a'_i - a_i$ and $\delta b_i = b'_i - b_i$ with expected value 0 and variance δ_2^2 . Then we have

$$(e_1 + \delta e_1)n'_1 + (e_2 + \delta e_2)n'_2 = -(e_3 + \delta e_3).$$

Let E and δE denote the $N \times 2$ matrices and G denote the $N \times 1$ matrix as follows,

$$\begin{aligned} E &= (e_{1i}, e_{2i})_N, \delta E = (\delta e_{1i}, \delta e_{2i})_N, \\ G &= (-e_{3i})_N, \delta G = (-\delta e_{3i})_N. \end{aligned}$$

Then the estimation $u' = (n'_1, n'_2)$ is obtained by solving the equation,

$$(E + \delta E)^T(E + \delta E)u' = (E + \delta E)^T(G + \delta G).$$

Let M denote $E^T E$. Assuming that the errors are much smaller than the real values, we develop the LS solution of u' in a Taylor expansion and obtain as an approximation for the estimate:

$$u' = u - \sum_i \sum_{t_i \in V} \delta t_i^2 \left(M^{-1} \begin{pmatrix} \frac{\partial^2 e_{1i}^2}{\partial t_i^2} & \frac{\partial e_{1i} e_{2i}}{\partial t_i^2} \\ \frac{\partial e_{1i} e_{2i}}{\partial t_i^2} & \frac{\partial^2 e_{2i}^2}{\partial t_i^2} \end{pmatrix} u + M^{-1} \begin{pmatrix} \frac{\partial^2 e_{1i} e_{3i}}{\partial t_i^2} \\ \frac{\partial^2 e_{2i} e_{3i}}{\partial t_i^2} \end{pmatrix} \right),$$

where V is the set of all variables $\{a_i, b_i, \dot{a}_i, \dot{b}_i\}$.

For the simplicity of expression, we consider a_i and b_i to be independent random variables which are symmetric with respect to the center of the image coordinate system; in other words, $E(a_i^k) = E(b_i^k) = 0, k = 1, 2$. Then with enough equations, the expected value for the LS solution u' is well approximated by

$$E(u') = u - M^{-1}(\delta_1^2 D + \delta_2^2 F)u - M^{-1}\delta_2^2 H, \quad (7)$$

where

$$D = \begin{pmatrix} N & 0 \\ 0 & N \end{pmatrix}; \quad H = \omega_3 \sum_i^N \begin{pmatrix} \omega_1 b_i^2 & 0 \\ 0 & a_i^2 \omega_2 \end{pmatrix},$$

$$F = \sum_i^N \begin{pmatrix} 4b_i^2 \omega_1^2 + c_i \omega_2^2 + \omega_3^2 & c_i \omega_1 \omega_2 \\ c_i \omega_1 \omega_2 & c_i \omega_1^2 + 4a_i^2 \omega_2^2 + \omega_3^2 \end{pmatrix}$$

where $c_i = a_i^2 + b_i^2$

3.3 The Effects on Slant

The slant σ is the angle between the surface normal and the negative Z -axis (0° slant corresponds to a plane parallel to the image plane, 90° slant corresponds to a plane that contains the optical axis) and the tilt τ is the angle between the direction of the projection of the surface normal onto the XY -plane and the X -axis. Using these coordinates $\frac{n}{\|n\|} = (\cos \tau \sin \sigma, \sin \tau \sin \sigma, \cos \sigma)$.

For the case when rotation around the Z -axis can be ignored (i.e., $\omega_3 = 0$) equation (7) simplifies to

$$E(u') = (I - \delta_A)u = (I - M^{-1}(\delta_1^2 D + \delta_2^2 F))u.$$

Since D and F are positive definite matrices, so is δ_A . And usually the perturbation δs are small. Then the eigenvalues of $(I - \delta_A)$ are between zero and one, which leads to the Rayleigh quotient inequality:

$$\frac{E(u')^T E(u')}{u^T u} \leq \|I - \delta_A\| < 1.$$

Since $\sigma = \cos^{-1}(1 + u^T u)$ is a strictly increasing function, by linear approximation, we have

$$E(\sigma') < \sigma,$$

which shows that slant is underestimated. The degree of underestimation highly depends on the structure of matrix M ; the inverse of M is involved in equation (7). Thus, the smaller the determinant of matrix M , the larger the bias in the estimation. The velocity of rotation also contributes to the magnitude of the bias as can be seen from matrix F ; larger velocity more bias.

We can say more about the dependence of slant estimation on the texture distribution. Recall from equation (3) that

$$e = \frac{(t \cdot \ell)}{\hat{z} L_m} L_d,$$

Let us consider a slanted plane whose texture only has two major directional components. Let the directional components be $L_{d_1} = (\cos \tau_1 \sin \sigma_1, \sin \tau_1 \sin \sigma_1, \cos \sigma_1)$ and $L_{d_2} = (\cos \tau_2 \sin \sigma_2, \sin \tau_2 \sin \sigma_2, \cos \sigma_2)$. Then we have

$$\begin{aligned}
M = E^T E &= \begin{pmatrix} \sum e_{1_i}^2 & \sum e_{1_i} e_{2_i} \\ \sum e_{1_i} e_{2_i} & \sum e_{2_i}^2 \end{pmatrix} \\
&= \sum \left(\frac{T \cdot \ell_i}{\hat{z} L_{m_i}} \right)^2 \sin^2 \sigma_1 \begin{pmatrix} \cos^2 \tau_1 & \sin \tau_1 \cos \tau_1 \\ \sin \tau_1 \cos \tau_1 & \sin^2 \tau_1 \end{pmatrix} \\
&\quad + \sum \left(\frac{T \cdot \ell'_i}{\hat{z} L'_{m_i}} \right)^2 \sin^2 \sigma_2 \begin{pmatrix} \cos^2 \tau_2 & \sin \tau_2 \cos \tau_2 \\ \sin \tau_2 \cos \tau_2 & \sin^2 \tau_2 \end{pmatrix}
\end{aligned}$$

and the determinant $\det(M)$ of M amounts to

$$\det(M) = [(\frac{1}{N} \sum (\frac{t \cdot \ell_{1_i}}{\hat{z} \cdot L_{M_{1_i}}})^2)^{\frac{1}{2}} (\frac{1}{N} \sum (\frac{t \cdot \ell_{2_i}}{\hat{z} \cdot L_{M_{2_i}}})^2)^{\frac{1}{2}} \sin \sigma_1 \sin \sigma_2 \sin(\tau_1 - \tau_2)]^2.$$

The smaller $\det(M)$, the larger the underestimation. Using our model we can predict the findings from experiments in the psychological literature ([11]). For example, it has been observed in [11], that an increase in the slant of a rotating surface causes increased underestimation of the slant. By our formula, it is easy to see that $\det(M)$ has a factor $\sin(\sigma_1) \sin(\sigma_2)$, where σ_1 and σ_2 are the angles between the directions of the line in space and the negative Z-axis. Unless, they are 0 degree, these values decrease with an increase of the slant of the plane, and this leads to a smaller $\det(M)$. Hence, we get a larger error towards underestimation of the slant.

To demonstrate the predictive power of the model we created two illusory displays. In the first one, the scene consists of a plane with two textures, one in the upper half, the other in the lower half. Figure 2a shows the plane when it is parallel to the screen. The texture in the upper part consists of two line clusters with slope 8° and 98° . The lower part has two lines clusters with slope 45° and 135° . A video was created for the camera orbiting the sphere along a great circle in the YZ plane as shown in Figure 2b – that is the camera translates and rotates such that it keeps fixating at the center. At the beginning of the motion, the slant of the plane with respect to the camera is 15° , at the end it is 45° . The image sequence can be seen in [4]. As can be experienced, it creates the perception of the plane to be segmented into two parts, with the upper part having a much smaller slant.

This is predicted by the biases in the different textures. For the upper texture the bias is much larger, thus producing larger underestimation of the slant, and the underestimation gets worse as the slant increases. The ratio of the determinants of the upper and lower texture is a good measure. For the given scene it takes values between 0.08 (for 15° slant) and 0.25. (for 45° slant). In a second display the plane is divided into multiple segments with two alternating textures. In every other segment there is large bias, and this gives rise to the perception of the plane folding as if it were a staircase.

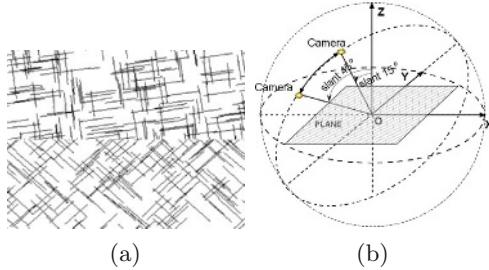


Fig. 2. (a) The plane in view (b) Scene geometry in the shape from motion demonstration.

4 Shape from Stereo

Here we adopt the symmetric stereo setting. That is, the coordinate system is in between the two cameras whose rotations with respect to this coordinate system are described by the rotation matrices R and R^T . We obtain the linear system

$$kL_d = (R\ell_1) \times (R^T\ell_2)$$

The transformation between the two views is a translation in the XZ plane and a rotation around the Y -axis with angle $2t$. By the same notion as in the previous section, we have as the prime equation for $n = (n_1, n_2, 1)$: $(e_{1i}, e_{2i}) \cdot \begin{pmatrix} n_1 \\ n_2 \\ 1 \end{pmatrix} = -e_{3i}$, where

$$\begin{cases} e_{1i} = b_{1i}(a_{2i} \sin t + \cos t) - (-a_{2i} \sin t + \cos t)b_{1i} \\ e_{2i} = -(a_{1i} \sin t + \cos t)(a_{2i} \cos t - \sin t) - (a_{1i} \cos t + \sin t)(a_{2i} \sin t + \cos t) \\ e_{3i} = -(a_{1i} \cos t + \sin t)b_{2i} + (a_{2i} \cos t - \sin t)b_{1i} \end{cases}$$

The measurement errors in the line locations are assumed to be i.i.d. with zero-mean and covariance δ_2 . Let $E = (e_{1i}, e_{2i})_n$. Under the small baseline assumption and some alignment of the image, we obtain as approximation for the expected value of the LS solution:

$$E(u') = u - nM^{-1}Fu,$$

where $M = E^t E$ and asymptotically

$$F = \delta_2 \begin{pmatrix} G(a_{1i}^2 + a_{2i}^2 + b_{1i}^2 + b_{2i}^2, t) & 0 \\ 0 & G(a_{1i}^2 + a_{2i}^2, 2t) \end{pmatrix},$$

where $G(x, t) = E(x) \sin^2 t + \cos^2 t$. By the same arguments as in the case of shape from motion, the slant is underestimated.

5 Statistical Alternatives

The statistical model that describes the data in visual estimation processes is the errors-in-variable model (Definition 1). The main problem with Least squares (LS) estimation is that it does not consider errors in the explanatory variables, that is δ_A . The obvious question thus arises: Are there better alternatives that reduce the bias? Clearly, bias is not the only thing that matters. There is a trade-off between bias and variance. Generally an estimator correcting for bias increases the variance while decreasing the bias.

Next we discuss well known approaches from the literature.

CLS (Corrected Least Squares) estimation is the classical way to correct for the bias. If the variance of the error is known, it gives asymptotically unbiased estimation. The problem is that accurate estimation of the variance of the error is a challenging task if the sample size is small. For small amounts of data the estimation of the variance has high variance itself. Consequently this leads to higher variance for CLS.

Usually the mean squared error (MSE) is used as a criterion for the performance of an estimator. It amounts to the sum of the square of the bias plus the variance of the estimator. According to this criterion the best linear estimation (linear in b) should be a partial correction using the CLS; the smaller the variance the larger the correction.

TLS (Total Least Square). The basic idea underlying this nonlinear technique [12] is to deal with the errors in A' and b' symmetrically. If all errors δ_A, δ_b are i.i.d., then TLS estimation is asymptotically unbiased. In the case they are not, one would need to whiten the data. But this requires the estimation of the ratio of the error variances δ_A and δ_b , which is at least as hard as obtaining the variance of δ_b . An incorrect value of the ratio often results in an unacceptably large *over correction* for the bias. However, the main problem for TLS is system error. We can have multiple tests to obtain the measurement error, like re-measuring or re-sampling; but unless we know the exact parameters of the model, we can't test the system error.

Resampling techniques, such as bootstrap and Jackknife are useful for estimating the variance, but cannot correct for the bias.

The **technique of instrumental variables** is an attractive alternative which deals with the errors in the explanatory variables but does not require the error variance as a priori. This techniques uses additional variables, the instrumental variables, which could be additional measurements of the explanatory variables (multiple edge detections, fitting schemes, and difference operators). If the errors in the measurements of the two methods can be treated as independent, an asymptotically unbiased estimator [5] can be created, whose variance is close to the variance of the CLS estimator.

5.1 Discussion of the Errors in Shape from Motion

The measurements are the line parameters $\{a_i, b_i\}$, and the image motion parameters of the lines, $\{\dot{a}_i, \dot{b}_i\}$. We can expect four types of noise:

Sensor noise: effects the measurements of image intensity $I(x, y, t)$. It seems reasonable to approximate the sensor noise as i.i.d.. But we have to consider dependencies when the images are smoothed.

Fitting error: Estimating the line parameters a_i, b_i amounts to edge detection. Clearly there are errors in this process. Longer edgels are associated with smaller errors and shorter edgels with larger errors.

Discretization Error: Derivatives are computed using difference operators, which have truncation errors associated with them. The magnitude of the error depends on the smoothness and the frequency of the texture.

System error: When computing the motion of lines, we assume that the image intensity is constant between frames. Significant errors occur at specular components. We use first order expansions when deriving velocities. Thus, errors are expected for large local velocities. Furthermore, the modeling of the scene as consisting of planar patches is an approximation to the actual surface of the scene.

Among the errors above, sensor noise has been considered in a number of papers in structure from motion ([9,10]). Other errors have hardly been mentioned or have been simply ignored. But actually other errors could contribute much more to the error than the sensor noise. Furthermore, the sensor characteristics may stay fixed. But other noise components do not. They change with the lighting conditions, the physical properties of the scene, and viewing orientation.

Considering all the errors, the errors δA_i and δb_i are due to a number of different components and cannot be assumed to be independent and identical. This makes the estimation of the variance unreliable. Thus CLS and TLS are not useful for correcting the bias. The technique of instrumental variables still can handle this model. Our experiments showed that this method resulted in some improvement, although minor.

6 Experiments

We compared the different regression methods for the estimation of slant from motion using simple textured planes as in the illusory video. TLS estimation was implemented by assuming all errors to be i.i.d.. CLS was implemented by assuming the errors in e_1 and e_2 to be i.i.d.. The variance of the errors was estimated by the SVD method, that is by taking the smallest singular value of the matrix $[A; b]$ as the estimation of the variance. In the first bootstrap methods the samples (e_{1i}, e_{2i}) were bootstrapped, in the second the residuals $e_{1i}n_1 + e_{2i}n_2 - e_{3i}$. For the instrumental variable method we used three differently sized Gaussian filters to obtain three samples for the image gradients.

We generated data sets of random textures with sparse line elements as in the videos. In texture set No.1 the lines have dominant directions 10° and 100° ; in texture set No.2 the dominant directions are 45° and 135° . We tested for two slants, 45° and 60° . The motion in the sequences was only translation, thus there is no error due to rotation. The tables below show the average estimated value of the slant for the four data sets.

Experiments with the slant 45°

No.	LS	CLS	TLS	Jack	Boot 1	Boot 2	Inst. Var.
1	41.8675	37.5327	54.8778	44.4426	43.0787	41.5363	43.0123
2	39.8156	40.8279	42.7695	39.0638	40.4007	40.1554	41.9675

Experiments with the slant 60°

No.	LS	CLS	TLS	Jack	Boot 1	Boot 2	Inst. Var.
1	45.7148	46.0307	46.6830	45.9929	46.2710	45.6726	49.3678
2	42.5746	44.4127	43.3031	47.1324	45.5572	42.8377	48.1202

The experiments demonstrate that LS tends to underestimate the parameters. TLS tends to give larger estimates than LS, but sometimes it overestimates the parameters, that is, it tends to over-correct the bias. CLS corrects the bias little. The reason could be either that the estimation of the variance is not trustable, or that the assumption that the measurement errors are independent is not correct. The performance of Bootstrap and Jackknife is not much better. The bias hardly gets corrected. The instrumental variable method seems a bit better than the other methods, but it still only corrects the bias by a small amount.

7 Conclusions and Discussion: The Key is Not in the Estimation

This paper analyzed the statistics of shape estimation. We showed that bias is a serious problem. We analyzed the bias for least squares estimation and we showed that it predicts the underestimation of slant, which is known from computational and psychophysical experiments.

One may question that LS estimation is a proper model for human vision. We discussed and showed experimentally that most elaborate estimators (CLS (unless one largely overestimates the variance) Bootstrap, Instrumental Variables) also have bias which is qualitatively of the same form as the one of LS. Thus these estimators, too, would lead to an underestimation of slant. TLS, depending on the ratio of variances, may give the opposite result.

Our analysis of shape from motion was possible because of a new constraint which relates shape and rotation only to image features. One may argue that using non-linear techniques we may estimate iteratively the parameters of the errors as well as the parameters of motion and structure. If we knew the error model and had a lot of data available, theoretically it would be possible to correct. However, we usually don't have enough data to obtain the errors, which depend on many factors. Furthermore, we don't know the exact error model.

The question, thus, for computational vision arises: How can we deal with the bias? Clearly, bias is not confined to shape estimation only. Other processes of visual reconstruction are estimation processes as well and thus will suffer from

the same problem. Since better estimation techniques are not the answer, we have to use the data such that bias does (mostly) not effect the goal, that is what we want to do with the data. First, we should use the data selectively. Since we understand how the different parameters influence the bias, we can choose data that is not effected much by the bias. For example in computing shape from motion we can avoid patches with textures corresponding to a badly conditioned matrix M . Second, we should use when possible, the data globally. Large amounts of data usually are not directionally biased, and thus the bias in estimation will be small. For example, when estimating 3D motion from image motion we should make use of all the data from the whole image. The same applies to shape estimation. Third, the statistics of structure from motion is easier for the discrete case than the continuous case, since in the discrete case the errors in the image measurements are expected to be less correlated. Thus, it is advantageous to estimate shape from views far apart. Of course, using far away views we run into the difficulty of finding good correspondence. The way to address structure from motion then is to use continuous motion to obtain a preliminary estimate of the 3D motion and shape, and subsequently use these estimates to obtain shape from views far apart. New constraints are needed for this task.

References

1. A.R. Chowdhury and R.Chellappa. Statistical error propagation in 3d modeling from monocular video. In *CVPR Workshop on Statistical Analysis in Computer Vision*, 2003.
2. K. Daniilidis and H.-H. Nagel. Analytical results on error sensitivity of motion estimation from two views. *Image and Vision Computing*, 8:297–303, 1990.
3. C. Fermüller and H. Malm. Uncertainty in visual processes predicts geometrical optical illusions. *Vision Research*, 44.
4. Cornelia Fermüller. <http://www.optical-illusions.org>, 2003.
5. W. Fuller. *Measurement Error Models*. Wiley, New York, 1987.
6. J. Garding. Shape from texture and contour by weak isotropy. *J. of Artificial Intelligence*, 64:243–297, 1993.
7. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
8. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier, Amsterdam, 1996.
9. N. Lydia and S. Victor. Errors-in-variables modeling in optical flow estimation. *IEEE Trans. on Image Processing*, 10:1528–1540, 2001.
10. H.H. Nagel. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *International Journal of Computer Vision*, 15:271–288, 1995.
11. J.M. Todd and V. J. Perotti. The visual perception of surface orientation from optical motion. *Perception & Psychophysics*, 61:1577–1589, 1999.
12. S. van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, 1991.

Contrast Marginalised Gradient Template Matching

Saleh Basalamah¹, Anil Bharath¹, and Donald McRobbie²

¹ Dept. of Bioengineering, Imperial College London, London SW7 2AZ, UK

{s.basalamah,a.bharath}@imperial.ac.uk

² Imaging Sciences Dept., Charing Cross Hospital, Imperial College London, UK

Abstract. This paper addresses a key problem in the detection of shapes via template matching: the variation of accumulator-space response with object-background contrast. By formulating a probabilistic model for planar shape location within an image or video frame, a vector-field filtering operation may be derived which, in the limiting case of vanishing noise, leads to the Hough-transform filters reported by Kerbyson & Atherton [5]. By further incorporating a model for contrast uncertainty, a contrast invariant accumulator space is constructed, in which local maxima provide an indication of the most probable locations of a sought planar shape. Comparisons with correlation matching, and Hough transforms employing gradient magnitude, binary and vector templates are presented. A key result is that a posterior density function for locating a shape marginalised for contrast uncertainty is obtained by summing the functions of the outputs of a series of spatially invariant filters, thus providing a route to fast parallel implementations.

1 Introduction

Primate vision employs various strategies in visual processing that appear to be highly effective. The apparent separation of visual input into various streams, or channels, of representation is one obvious example. These interpretations, collectively referred to as “channel models” of vision have been suggested for decades, and is backed up by much physiological [12] and psychophysical evidence. From a pattern recognition viewpoint, such ideas might not be surprising: the construction of multiple channels is closely analogous to the extraction of the components of a feature space constructed for pattern recognition purposes. In this paper, the issue of constructing a posterior distribution over spatial variables from a generic gradient field is discussed as a means of obtaining shape-specific responses.

Gradient field estimates are thought to play a significant role in biological vision, and particularly for distinguishing form (shape). At the same time, gradient field estimates are used as the basis of the Compact Hough Transform for detecting closed shapes [1].

In the rest of the paper, a generic probabilistic framework is suggested for locating planar shapes from gradient field observations, and this leads to (i) a

new interpretation of the Hough Transform filters suggested by Kerbyson and Atherton [5], and related techniques for medial axis detection [9] suggested by Bharath [2] (ii) an improvement in the principle of contrast invariant shape measures based on non-linear combinations of the outputs of linear spatial filters.

2 A Probabilistic Formulation

This paper addresses the task of locating compact planar shapes from gradient-fields of digital images, expressed as follows:

Q1: Given an estimate of a gradient field of a digital image, what is the most likely location of a known planar shape, S_0 , in the image ?

In answering this question, one may derive techniques that are related to existing methods of planar shape localisation and are in the spirit of Hough-transform approaches. For compact shapes, Q1 requires the definition of a unique point of origin on the shape. The problem of finding the position of a known shape is reduced to one of finding its point of origin within the image plane.

Remark 1. The reason for restricting the observations to a gradient field, rather than the original image itself, is to focus on the nature of *shape*. There are, however, some other intriguing reasons for investigating this route: in sparse image (transform coding) it is quite easy to extract a gradient field estimate from the transform domain representation.

A statistical formulation for the question posed above exists in the form of conditional probability, and the most probable shape location, \mathbf{t}_{opt} , given the gradient field data may be found from

$$\mathbf{t}_{opt} = \arg \max_{\mathbf{t} \in \mathcal{R}_I} \{f_t(\mathbf{t} | \mathcal{B}, \mathcal{X}; \Theta)\} \quad (1)$$

where $(\mathcal{B}, \mathcal{X})$ represents N pairs of vector tuples, $\{(\mathbf{b}^{(i)}, \mathbf{x}^{(i)})\}_{i=1..N}$, with $\mathbf{b}^{(i)}$ denoting an estimate (observation) of the gradient vector field at position $\mathbf{x}^{(i)}$. The possible location of the shape is denoted by the position vector, \mathbf{t} . More generally, one may seek local maxima of the posterior distribution on the right hand side of Equation (1) as an indication of the most probable locations of the shape in question, or of multiple instances of the same shape. The information about the particular shape being sought, and its observation conditions (such as rotation and scale), are contained in the parameter vector Θ .

Equation (1) represents a concise formulation of the shape detection problem, given observed data and known parameters. The first difficulty is that of estimating this probability density function (or, more precisely, finding its local maxima). Using Bayes' Theorem,

$$f_t(\mathbf{t} | \mathcal{B}, \mathcal{X}; \Theta) = \frac{f(\mathcal{B}, \mathcal{X} | \mathbf{t}; \Theta) f_t(\mathbf{t} | \Theta)}{f(\mathcal{B}, \mathcal{X} | \Theta)} \quad (2)$$

For a suitable choice of likelihood function and prior, it may be shown that Equation (2) leads to several algorithms for efficient construction of an accumulator space through convolution. Some of these have been identified previously, but new results may be obtained by a careful consideration of the likelihood model, and by marginalising over variations in contrast.

3 Plausible Likelihood Functions

A plausible form for the likelihood function, $f(\mathcal{B}, \mathcal{X}|\mathbf{t}; \Theta)$ is required. For a scalar intensity image, the value of the gradient vector field at location $\mathbf{x}^{(i)}$, $\mathbf{b}(\mathbf{x}^{(i)})$, is typically obtained by convolving the image with gradient masks¹ Such masks are designed to have zero average value, and are well localised in space. Thus, these masks approximate the properties of discrete wavelets, and the resulting gradient fields, despite their non-completeness, have univariate statistics that are well understood from the wavelet literature [10,11].

To characterise the statistical behaviour of a gradient field estimate consisting of two components, the *joint* statistics of the components of the vector $\mathbf{b}(\mathbf{x}^{(i)})$ should be determined, as should the joint statistics between two different field locations. Such high-dimensional density functions are difficult to model and approximate. To simplify the specification of a probabilistic model of gradient field behaviour, independence of these components, across all scenery, might be suggested. This is unrealistic, even for quite simple visual scenes.

A solution is to assume that the statistics are conditionally independent; that is, *given* the presence of some structure in the image, and certain restricted observation conditions, the vector components at any pixel deviate from a mean *vector* in an independent fashion. Conditional independence between observations is precisely what is needed for the construction of a useful likelihood function. Conditional independence between the components represents a simplified model, but it is a far weaker assumption than that of unconditional independence. Furthermore, although the independence assumption may fail under certain conditions, it is likely that there exist some variables such that the independence assumption of the joint statistics *conditioned on those variables* does hold. Finally, it should also be pointed out that marginalising over the variables under which the density function is defined yields unconditional joint statistics which are *no longer* independent.

3.1 Models for Gradient Field Statistics

It is widely recognized that the univariate statistics of wavelet coefficient space are well approximated by a Laplacian probability density function (PDF). For

¹ The dual notation that is used serves to (a) formulate the detection problem in terms recognizable as Bayesian inference, (b) illustrate the important link with spatial correlation/convolution, and (c) lay the ground for future work in which possible spatial distortion (uncertainty in the location $\mathbf{x}^{(i)}$) is corrected for.

example, Sendur and Selesnick [10] developed a bivariate Laplacian model for a wavelet coefficient space which they applied to denoising.

It turns out that the Laplacian density function suggested above is not critical for expressing the distribution of values in a *conditional* gradient field in a region near to and enclosing a shape of interest. Indeed, adopting the view of wavelet statistics, the Laplacian distribution is merely a reflection of the sparseness of the “detail” coefficient space. A Laplacian-like PDF can arise from marginalising a conditional bivariate Gaussian model over the location variable, \mathbf{t} .

3.2 Importance of Conditional Statistics

To develop the formulation in a concise way, it will be assumed that the observation conditions, Θ , are presumed fixed over all experimental space; the effect of contrast variations will be developed in Section (3.3). Then, *given* that a shape S_0 is present at location \mathbf{t} in an image, the statistics of the transform space can be modelled by a distribution in a simple, but elegant way: the *deviation* from the conditional mean vector is approximated by an appropriate distribution, where the conditional mean vector field is specified as a function of position in the observation field. To express this more clearly, the auxiliary random vector field conditional variable $\beta(\mathbf{x}^{(i)}|\mathbf{t}, S_0)$ is introduced:

$$\beta(\mathbf{x}^{(i)}|\mathbf{t}, S_0) = \mathbf{b}(\mathbf{x}^{(i)}) - \mu(\mathbf{x}^{(i)}|\mathbf{t}, S_0) \quad (3)$$

where $\mu(\mathbf{x}^{(i)}|\mathbf{t}, S_0)$ is the mean gradient vector field (over experimental space) of the gradient field *conditional* on shape S_0 being present at location \mathbf{t} .

A bivariate Gaussian version of the statistics of $\beta(\mathbf{x}^{(i)})$ is, therefore, both convenient and realistic, as will be demonstrated; it has the form

$$f_\beta(\beta(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \mu(\mathbf{x}^{(i)})) = \frac{\alpha_\beta}{\pi} e^{-\alpha_\beta |\beta(\mathbf{x}^{(i)}|\mathbf{t}, S_0, \mu(\mathbf{x}^{(i))))|^2} \quad (4)$$

which is a “standard” bivariate normal distribution, with α_β being half the inverse variance.

Remark 2 Caution is required in interpreting Equation(4); it is *not* a specification of the distribution of the *magnitude* of the gradient field deviation, but a bivariate distribution which decays (as a Gaussian) with the magnitude of the field deviation - this is a subtle, but crucial, distinction.

By considering well-defined shape localisation problems based on gradient estimates, it is possible to demonstrate that distributions of the form of Equation(4) can be representative of real gradient field observations, and, furthermore lead to the derivation of previous reported results and new methodologies for shape detection.

To validate the proposed model for gradient field statistics in the presence of objects, a simple experiment was conducted. An image of garden peas scattered across a rug, Figure (1a), was acquired using a standard consumer digital camera

(FujiFilm FinePix, 4700) on the “Normal Quality” setting at a resolution of 1280×960 . Normal artificial room lighting conditions were used, and the object-camera distance was of the order of 1.5 metres. The “V” channel of the HSV colourspace representation was extracted from the JPEG file of the camera. Using this channel, a thresholding operation was applied to binarise the image, then connected components labelling used to find individual peas. The centroids of the binary pea images were found, and a 33×33 area centred on several pea centroids was extracted from a gradient field estimate by using simple Prewitt operators. A mean gradient field was computed, and the bivariate probability density function of the deviation of actual gradient estimates from this mean was estimated by histogram binning. The result is shown in Fig. (1b). The total number of pixels used to estimate this histogram (on a 20×20 grid of bins) is approximately 20,000.

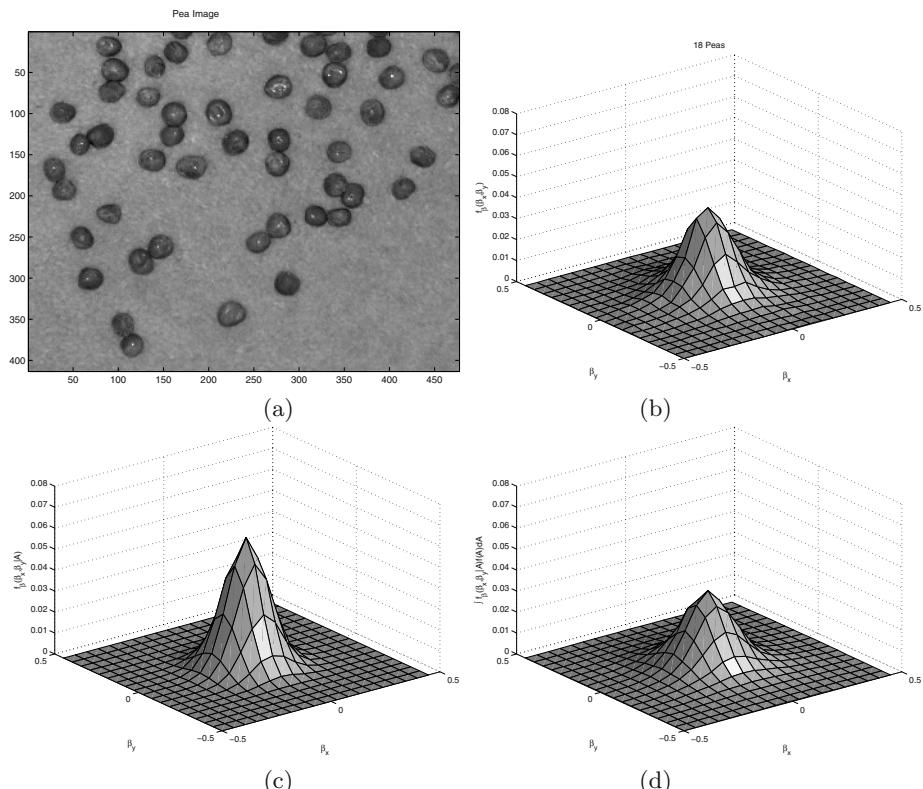


Fig. 1. (a) Peas-on-Rug image (b) Experimentally estimated bivariate PDF or region containing single peas from the “Pea-on-rug” image obtained by methods described in the text. (c) Sampling from $f_{\beta}(\beta_x, \beta_y | A)$ for $A = 1$. (d) Sampling from $f_{\beta}(\beta_x, \beta_y | A)$ and the prior on A , $f(A)$, and marginalising over A .

3.3 A Model for Contrast Uncertainty

Between any two gradient images, there will be a certain degree of contrast uncertainty. This contrast uncertainty will scale all values, including the x and y components of the gradient estimate around objects of interest, and might be considered one mechanism which leads to a violation of independence between components or pixels of a single image.

Although there will be uncertainty in this contrast value, some prior information about this uncertainty, and particularly about its role in providing “coupling” between the components of the gradient field, may be postulated. The following model for auxiliary conditional gradient field is proposed,

$$\beta(\mathbf{x}^{(i)}|\mathbf{t}, S_0, \mu(\mathbf{x}^{(i)}), A) = A\mathbf{b}(\mathbf{x}^{(i)}) - \mu(\mathbf{x}^{(i)}|\mathbf{t}, S_0) \quad (5)$$

together with a prior for the contrast (amplitude scaling) parameter, A :

$$f_A(A) = \sqrt{\frac{\alpha_A}{\pi}} e^{-\alpha_A(A-\mu_A)^2} \quad (6)$$

and, for the auxiliary conditional random vector field deviation,

$$f_\beta(\beta(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \mu(\mathbf{x}^{(i)}), A) = \frac{\alpha_\beta}{\pi} e^{-\alpha_\beta |\beta(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \mu(\mathbf{x}^{(i)}), A|^2} \quad (7)$$

In Fig. (2b), a sample is drawn from the gradient field model of small, bright circles of radius around 3 pixels against a darker background. The parameter μ_A was set to 1, α_β was set to 200, and $\alpha_A = 1$, i.e. corresponding standard deviations of 0.05 and $\sqrt{1/2}$ respectively. The conditional mean vector field, $\mu(\mathbf{x}^{(i)}|\mathbf{t} = (0, 0), A = 1)$ is illustrated in Fig. (2a). The reader should note that the model includes a variation in contrast which introduces coupling between x and y gradient components and *independent* noise on the x and y components.

To show that this approach leads this to plausible PDF's that reflect those of real image data, one may use sampling principles [3] to marginalise the conditional density function, $f_\beta(\beta(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \mu(\mathbf{x}^{(i)}), A)$ over A using a standard normally distributed random number generator², with $\mu_A = 1$, and $\alpha_A=1$. The result of this is shown in Fig. (1d). This should be compared with the estimates from the “pea on carpet” image of Fig. (1b).

3.4 Analytic Marginalisation over Contrast

Analytically, the marginalisation of Equation (7) over A using the prior of Equation (6) is also tractable, and leads to the main new useful results of this paper:

$$f_\beta(\beta(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \mu_t(\mathbf{x}^{(i)})) = \int_{-\infty}^{\infty} f_\beta(\beta(\mathbf{x}^{(i)})|\mathbf{t}, S_0, A, \mu_t(\mathbf{x}^{(i)})) f_A(A) dA \quad (8)$$

² Matlab 6.5 running under Windows 2000

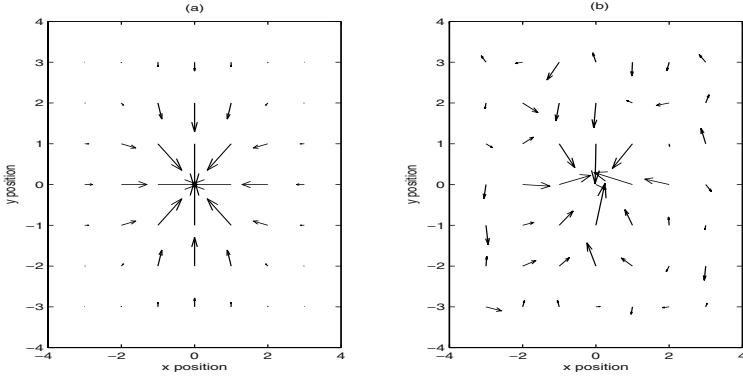


Fig. 2. (a)Left. Illustration of mean vector field conditional on a bright circle being present at location $(0,0)$ on a dark background. (b) Right. Sample of vector field drawn from $f_\beta(\mathbf{b}(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \boldsymbol{\mu}(\mathbf{x}^{(i)}), A)$ using vector field to the left. S_0 is a circle of radius 3 pixels

where, using a standard book of integral tables [4]

$$\int_{-\infty}^{\infty} \exp \left\{ -p^2 x^2 \pm qx \right\} dx = \frac{\sqrt{\pi}}{p} \exp \left\{ \frac{q^2}{4p^2} \right\}, \quad p > 0 \quad (9)$$

and Equation (3) results in the following,

$$\begin{aligned} f_b(\mathbf{b}(\mathbf{x}^{(i)})|\mathbf{t}, S_0, \boldsymbol{\mu}(\mathbf{x}^{(i)})) \\ = \frac{\alpha_\beta}{\pi} \sqrt{\frac{\alpha_A}{Z_0^{(i)}}} \exp \left\{ \frac{\alpha_\beta}{Z_0^{(i)}} \left[2\alpha_A \left(b_x(\mathbf{x}^{(i)})\mu_x(\mathbf{x}^{(i)}|\mathbf{t}) \right. \right. \right. \\ \left. \left. + b_y(\mathbf{x}^{(i)})\mu_y(\mathbf{x}^{(i)}|\mathbf{t}) \right) \right. \\ \left. + 2\alpha_\beta \left(b_x(\mathbf{x}^{(i)})b_y(\mathbf{x}^{(i)})\mu_x(\mathbf{x}^{(i)}|\mathbf{t})\mu_y(\mathbf{x}^{(i)}|\mathbf{t}) \right) \right. \\ \left. - \left(\alpha_A + \alpha_\beta(b_y^2(\mathbf{x}^{(i)})) \right) \mu_x^2(\mathbf{x}^{(i)}|\mathbf{t}) \right. \\ \left. - \left(\alpha_A + \alpha_\beta(b_x^2(\mathbf{x}^{(i)})) \right) \mu_y^2(\mathbf{x}^{(i)}|\mathbf{t}) \right. \\ \left. - \alpha_A \left(b_x^2(\mathbf{x}^{(i)}) + b_y^2(\mathbf{x}^{(i)}) \right) \right] \right\} \end{aligned} \quad (10)$$

with

$$Z_0^{(i)} = \alpha_A + \alpha_\beta \left[b_x^2(\mathbf{x}^{(i)}) + b_y^2(\mathbf{x}^{(i)}) \right] \quad (11)$$

and where $\mu_x(\mathbf{x}^{(i)}|\mathbf{t})$ and $\mu_y(\mathbf{x}^{(i)}|\mathbf{t})$ are the x and y components of the mean gradient field, $\boldsymbol{\mu}(\mathbf{x}^{(i)}|\mathbf{t})$; $b_x(\mathbf{x}^{(i)})$ and $b_y(\mathbf{x}^{(i)})$ are the x and y components of the observed vector gradient field field at the i^{th} pixel, located at $\mathbf{x}^{(i)}$.

Assuming that all gradient observation pixels are drawn independently from this model, then, for a set of N observations,

$$\begin{aligned} f_b(\{\mathbf{b}(\mathbf{x}^{(i)})\}_{i=1..N} | \mathbf{t}, S_0, \boldsymbol{\mu}(\mathbf{x}^{(i)})) &= \prod_{i=1}^N f_b(\mathbf{b}(\mathbf{x}^{(i)}) | \mathbf{t}, S_0, \boldsymbol{\mu}_t) \\ &= Z_N \exp\{\alpha_\beta(2\alpha_A C_1 + 2C_2 - C_3 - C_4 - \alpha_A C_5)\} \end{aligned} \quad (12)$$

where $Z_N = \prod_{i=1}^N \frac{\alpha_\beta}{\pi} \sqrt{\frac{\alpha_A}{Z_0^{(i)}}}$, and

$$C_1 = \sum_{i=1}^N \frac{b_x(\mathbf{x}^{(i)})\mu_x(\mathbf{x}^{(i)}|\mathbf{t}) + b_y(\mathbf{x}^{(i)})\mu_y(\mathbf{x}^{(i)}|\mathbf{t})}{Z_0^{(i)}} \quad (13)$$

$$C_2 = \sum_{i=1}^N \frac{b_x(\mathbf{x}^{(i)})b_y(\mathbf{x}^{(i)})\mu_x(\mathbf{x}^{(i)}|\mathbf{t})\mu_y(\mathbf{x}^{(i)}|\mathbf{t})}{Z_0^{(i)}} \quad (14)$$

$$C_3 = \sum_{i=1}^N \frac{\alpha_A + \alpha_\beta b_y^2(\mathbf{x}^{(i)})\mu_x^2(\mathbf{x}^{(i)}|\mathbf{t})}{Z_0^{(i)}} \quad (15)$$

$$C_4 = \sum_{i=1}^N \frac{\alpha_A + \alpha_\beta b_x^2(\mathbf{x}^{(i)})\mu_y^2(\mathbf{x}^{(i)}|\mathbf{t})}{Z_0^{(i)}} \quad (16)$$

and

$$C_5 = \sum_{i=1}^N \frac{b_x^2(\mathbf{x}^{(i)}) + b_y^2(\mathbf{x}^{(i)})}{Z_0^{(i)}} \quad (17)$$

Each of the terms C_1, C_2, C_3, C_4 involves spatial convolution (or cross correlation) between the appropriate mask and simple functions of the gradient field components. Term C_1 , for example approaches Kerbyson and Atherton [5] filtering technique if α_A is significantly greater than α_β . Returning to Equations (1) and (2), one may write

$$f_t(\mathbf{t}|\mathcal{B}, \mathcal{X}; \boldsymbol{\Theta}_1) \propto Z_N \exp\{\alpha_\beta(2\alpha_A C_1 + 2C_2 - C_3 - C_4 - \alpha_A C_5)\} f_t(\mathbf{t}|\boldsymbol{\Theta}_1) \quad (18)$$

The choice of the prior $f(\mathbf{t}|\boldsymbol{\Theta}_1)$ may be completely non-informative, or may be chosen as coming from other data fields in a real image. For example, in searching for faces in a scene, the prior may be derived from some measure in HVS colour space; in searching for sub-components within an automotive inspection application, geometric considerations may yield a useful prior.

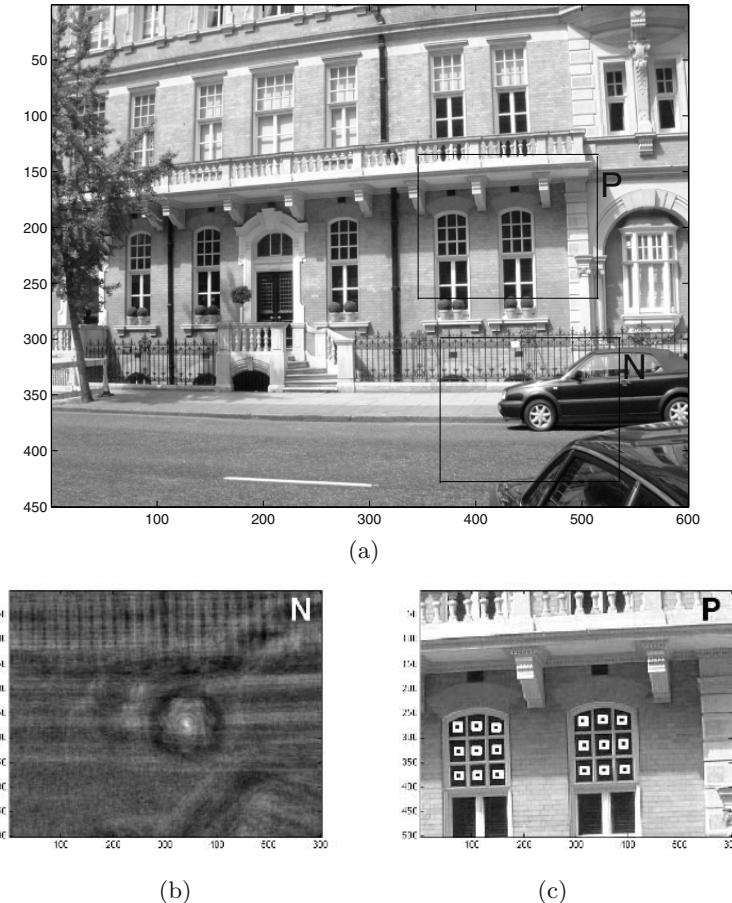


Fig. 3. (a) Image of building and car (downsampled by a factor of 4) (b) Accumulator space for circles, or $\log(f_t(\mathbf{t}|\mathcal{B}, \mathcal{X}; \Theta_1))$ (c) Detected locations of square centroids. For reasons of space and clarity, only the results after eliminating disconnected pixels in thresholded space are shown. An image with overlaid accumulator space is presented in the supplementary material.

4 Experimental Results

The performance of the technique has been tested on a range of images. Comparisons with other detection methods have been performed such as the correlation and gradient magnitude implementation of the Compact Hough transform. In all cases a flat prior on \mathbf{t} was chosen.

4.1 Natural Image Test

To represent the performance of the technique in natural images a test has been conducted on a 1800x2400 image of a building and car (Fig. (3a)), where the

aim is to detect the small rectangular windows in the image and the car wheel. The windows have a size of 34x48 pixels and the car wheel has a diameter of 80 pixels. Two different geometrical templates (a rectangle for detecting the windows and a circle for detecting the wheel) were created using generalized rectangular functions [6] that mimic optical blurring.

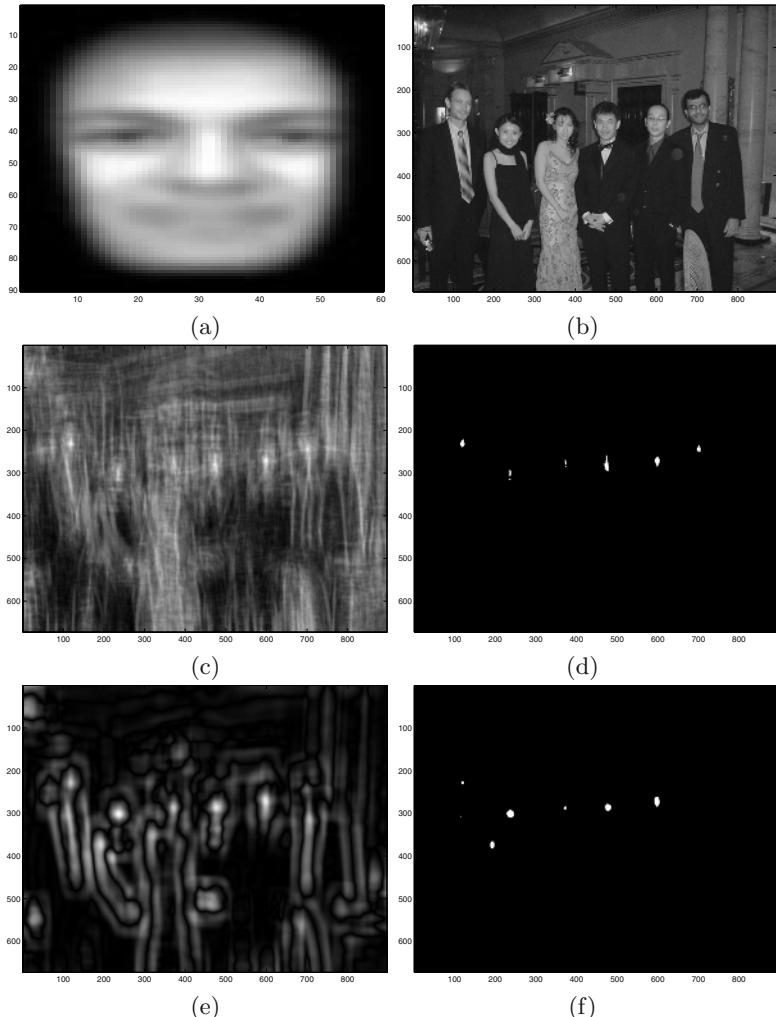


Fig. 4. (a) Average face template. (b) Image of people. (c) Marginalized gradient accumulator space. (d) Thresholded accumulator space showing face locations. (e) Correlation accumulator space. (f) Thresholded correlation accumulator space.

4.2 Face Detection

The technique has also been applied for detecting non-geometric objects. “A typical averaged face”, Fig.(4a), prepared by Adam D. Bradley, was downloaded from Dr. Robert Frischholz’s Face Detection Homepage³ and was used to detect faces in Fig. (4b). The technique correctly detects the faces as they are the highest local maxima in the accumulator space (Fig.(4c)). In comparison, the accumulator space of correlation gives high responses at most face locations, but also responds to other objects in the image as well (Fig.(4e)). The design of the gradient template, including the scale of the gradient field operators, remains an important aspect of this formulation and much work would need to be done in this area.

4.3 Circle and Square Test

To test the behaviour of the technique under different contrast conditions and noise levels, a 241x241 image containing circles and squares was synthesized. The background contrast of the image is varied linearly along the horizontal axis and also random noise is added to the image. There are squares and circles of the same size, aligned horizontally and vertically. Circles and squares with variation of both size and contrast are also placed along the vertical axis. Fig.(5a) represents the image with a random noise of $\sigma = 1$.

To demonstrate the discriminative performance of the technique, receiver operating characteristic (ROC) curves were plotted for each of 4 techniques: (a) Cross Correlation between image and template, (b) Template matching between gradient magnitude of image and template, (c) Template matching between thresholded gradient magnitude of image and binary template, (d) Kerbyson and Atherton’s method: template matching between x-y gradients of image and x-y gradients of template.

The ROC is a plot of true positive rate (TP) against the false positive rate (FP). The true positives represent the correct detection of squares with a side-length of 9 pixels, whereas the false positives represent the detection of all the other objects (circles with different diameters and squares with different side-lengths). 100 threshold levels are applied on the accumulator space (Fig. (5b)) where TP and FP are calculated for each level. The ROC plot in Fig. (4c) compares our technique with 4 other detection techniques in images with random noise of $\sigma = 2$.

Since the total number of objects in the image is only 25, the number of experiments is increased to include 10 images, each with different noise realisations thus raising the total number of objects at each threshold level to 250.

5 Conclusions and Further Work

Marginalisation over a contrast variation solves the problems of early threshold setting that plague many Hough transform techniques. Whilst exact sampling

³ <http://home.t-online.de/home/Robert.Frischholz/face.htm>

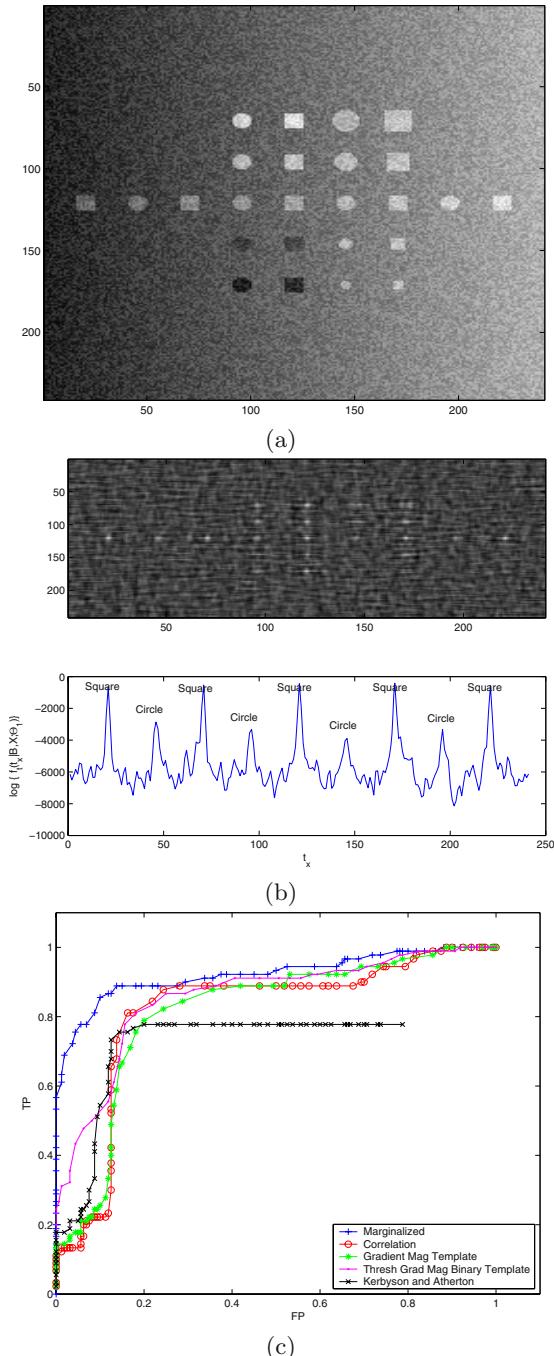


Fig. 5. (a) Synthesized image of circles and squares. (b) (Top) $\log(f_t(\mathbf{t}|\mathcal{B}, \mathcal{X}; \Theta_1))$ for 9 pixel squares detection, (Bottom) Profile through top response at row=121. (c) ROC comparison of different detection methods.

methods provide the possibility of more accurate estimates of posterior density functions of shape locations, the approach employed in this paper leads to a direct implementation through spatial convolution. This work has not addressed the crucial problem of handling within plane affine geometric transformation of the shape being sought [8]. Strategies for handling this problem using either sequential parameter estimation techniques [7] or marginalisation over, for example, rotation, are under investigation. Finally it should also be pointed out that the scale of gradient estimators is likely to play a critical role in distinguishing shapes that are close to each other and in the tolerance of marginalised filtering to noise.

Acknowledgement. This work is sponsored by the Saudi Ministry of Higher Education and is partially supported by the UK Research Council under the Basic Technology Research Programme "Reverse Engineering Human Visual Processes" GR/R87642/02.

References

1. D.H. Ballard. Generalising the Hough Transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.
2. A. A. Bharath and C.J. Huberson. Obtaining medial responses from steerable filters. *IEE Proceedings on Vision, Image and Signal Processing*, 146(5):1087–1088, 1999.
3. Michael Evans and Tim Swartz. *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press, 2000.
4. I.S. Gradshteyn, I.M. Ryzhik, and Alan Jeffrey. *Tables of Integrals, Series and Products: 5th Edition*. Academic Press, 1994.
5. J.H. Kerbyson and T.J. Atherton. Circle detection using hough transform filters. In *Image Analysis and its Applications, 1995, IEE Conference Publication No. 410*, pages 370–374, 1995.
6. M.J. Lighthill. *Introduction to Fourier Analysis and Generalised Functions*. Cambridge monographs on mechanics and applied mathematics. Cambridge: Cambridge University Press, 1958.
7. David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
8. Maria Petrou and Alexander Kadyrov. Affine invariant features from the trace transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:30–44, 2004.
9. Stephen M. Pizer, Christina A. Burbeck, James M. Coggins, Daniel S. Fritsch, and Bryan S. Morse. Object shape before boundary shape: Scale-space medial axes. *Journal of Mathematical Imaging and Vision*, 4:303–313, 1994.
10. Levent Sendur and Ivan W. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Transactions on Signal Processing*, 50(11), November 2002.
11. E.P. Simoncelli and E.H. Adelson. Noise removal via bayesian wavelet coring. In *Proceedings of the 1996 International Conference on Image Processing*, volume 1, pages 379–382, September 1996.
12. Martin J. Tovée. *An Introduction to the human visual system*. Cambridge Press, 1996.

The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition

Nuno Vasconcelos¹, Purdy Ho², and Pedro Moreno²

¹ Department of Electrical and Computer Engineering,
University of California San Diego,
9500 Gilman Drive, MC 0407, San Diego, CA 92093, USA
nuno@ece.ucsd.edu

² HP Cambridge Research Laboratory,
One Cambridge Center, Cambridge MA 02142
{purdy.ho,pedro.moreno}@hp.com

Abstract. The recognition accuracy of current discriminant architectures for visual recognition is hampered by the dependence on holistic image representations, where images are represented as vectors in a high-dimensional space. Such representations lead to complex classification problems due to the need to 1) restrict image resolution and 2) model complex manifolds due to variations in pose, lighting, and other imaging variables. Localized representations, where images are represented as bags of low-dimensional vectors, are significantly less affected by these problems but have traditionally been difficult to combine with discriminant classifiers such as the *support vector machine* (SVM). This limitation has recently been lifted by the introduction of probabilistic SVM kernels, such as the *Kullback-Leibler* (KL) kernel. In this work we investigate the advantages of using this kernel as a means to combine discriminant recognition with localized representations. We derive a taxonomy of kernels based on the combination of the KL-kernel with various probabilistic representation previously proposed in the recognition literature. Experimental evaluation shows that these kernels can significantly outperform traditional SVM solutions for recognition.

1 Introduction

The formulation of visual recognition as a problem of statistical classification has led to various solutions of unprecedented success in areas such as face detection, face, texture, object, and shape recognition, or image retrieval. There are, however, various fundamental questions in the design of classifiers for recognition that remain largely unanswered. One of the most significant is that of identifying the most suitable classification architecture. Broadly speaking, there are two major architecture classes: that of discriminant classifiers and that of classifiers based on generative models. On one hand, modern learning theory favors the use of discriminant solutions, namely the large-margin classifiers inspired by VC

theory [5], for which there is an appealing guiding principle (“do not model more than what is needed”) and a more rigorous understanding of properties such as the generalization error than what is available for generative solutions. On the other hand, generative models have various properties of great appeal for the implementation of recognition systems. In particular they 1) have much better scalability in the number of classes and amount of data per class, 2) enable the encoding of knowledge about the classification problem in the choice of statistical models and, therefore, are significantly more flexible, and 3) allow modular solutions, where Bayesian inference is used to integrate the contributions of various modules into the optimal decision for a large classification problem.

For visual recognition, one of the fundamental differences between the two approaches is the set of constraints that are imposed on image representation. While generative models favor a representation of the image as a large collection of relatively low-dimensional features, discriminant solutions work best when images are represented as points in some high-dimensional space. Hence, while a *localized* image representation is usually adopted in the generative setting (e.g. by representing each image as a bag of 8×8 image blocks), on the discriminant setting the representation frequently consists of a *holistic* low-resolution replica, e.g. 20×20 pixels, of the original image. While this holistic representation has the clear advantage of capturing global attributes of the objects of interest, e.g. that eyes, nose, and mouth always appear in a given configuration in face images, it has various disadvantages over the localized representation. These include 1) a much higher susceptibility to invariance problems due to either image transformations, non-rigid objects, or occlusion and 2) a significant loss of information due to the need to downsample images severely in order to keep the dimensionality of the space tractable. Due to these problems, localized representations are frequently advocated or adopted for recognition tasks, leading to generative classifiers [4,6]. While there is a sense that such classifiers imply some loss in recognition accuracy, the difficulty of combining discriminant techniques with the localized representation makes the discriminant alternative impractical.

In this work we consider one of the most popular discriminant architectures, the *support vector machine* (SVM). SVMs are large-margin classifiers obtained by solving a convex programming problem that depends on the training data through a kernel matrix that captures the distances between all pairs of examples. For a training set of size N , this results in a $O(N^2)$ complexity for any SVM learning algorithm, rendering localized representations (where each image can lead to a bag of thousands of examples) intractable. It has, however, been recently observed [7] that a natural extension of this formulation is to consider kernel matrices that capture distances between *the generative models associated with each bag of examples instead of the examples themselves*. This observation has motivated the introduction of various kernels based on probabilistic models, e.g. the *Fisher* [7], *Kullback-Leibler* [8], *TOP* [11], and *Battacharya* [12] kernels. In this paper, we investigate the benefits of the Kullback-Leibler (KL) kernel for visual recognition. In particular, we show that it subsumes many kernels based on localized representations that have been argued to be interesting, or shown to work well, for recognition. We provide closed-form expressions for the kernel

as a function of the parameters of the probabilistic models whenever they exist, and discuss alternatives for the construction of the kernel matrix when this is not the case. Finally, a detailed experimental evaluation is presented, illustrating the result of the various trade-offs associated with the various combinations of localized vs holistic representations and generative vs discriminant classifiers.

2 SVMs and Kernel Functions

In this section, we present a brief review of the SVM architecture and the constraints that it poses on image representation for recognition.

2.1 The SVM Architecture

Consider a binary classification problem with a training set consisting of (input,output) pairs $(\mathbf{x}_i, y_i) \in \mathcal{X} \times Y$, $Y = \{-1, 1\}$. Assuming that the data is separable¹, the optimal (in the maximum margin sense) *linear* separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ is the solution to the constrained optimization problem [5]

$$\max \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad \text{subject to} \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad (1)$$

where $\{\alpha_i\}$ is a set of Lagrange multipliers, and

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \quad b = 1/|I| \sum_{i \in I} [y_i - \sum_j y_j \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)] \quad (2)$$

with $I = \{i | \alpha_i > 0\}$. One of the appealing properties of this formulation is that it depends only on the dot products of the training vectors. This allows the automatic extension of the optimal solution to the, seemingly much more complicated, problem of designing large-margin classifiers with non-planar boundaries.

This extension consists of introducing a non-linear mapping $\Phi : \mathcal{X} \rightarrow \mathcal{Z}$ from the original input space \mathcal{X} to a new feature space \mathcal{Z} . Typically $\mathcal{X} = R^d$ and $\mathcal{Z} = R^p$ where p is significantly larger than d and linear boundaries in \mathcal{Z} are equivalent to non-linear boundaries in \mathcal{X} . It follows from the discussion above that the optimal solution in \mathcal{Z} is given by (1), (2) with the inner products $(\mathbf{x}_i \cdot \mathbf{x}_j)$ replaced by $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. In \mathcal{X} , this is equivalent to simply introducing a *kernel function* $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, under the constraint that this function must be an inner product in some space \mathcal{Z} , i.e.

$$\exists(\mathcal{Z}, \Phi), \Phi : \mathcal{X} \rightarrow \mathcal{Z} \quad \text{such that} \quad \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \quad (3)$$

Mercer's theorem assures that this condition holds whenever $\mathcal{K}(\mathbf{x}, \mathbf{y})$ is a positive definite form [5]. Notice that an inner product is nothing but a measure of vector

¹ All the results in this paper apply equally well to the extension to the non-separable case [5]. We omit it here for simplicity.

similarity and, since $\|\mathbf{x} - \mathbf{y}\|^2 = (\mathbf{x} \cdot \mathbf{x}) - 2(\mathbf{x} \cdot \mathbf{y}) + (\mathbf{y} \cdot \mathbf{y})$, the standard dot product implies an Euclidean metric on \mathcal{X} . Under this interpretation, the role of the kernel is to enable extensions to non-Euclidean measures of similarity. Hence, the *kernel matrix* $\mathcal{K}(\mathbf{x}_i \cdot \mathbf{x}_j)$ can be seen as capturing the similarity between points $\mathbf{x}_i, \mathbf{x}_j$ under the similarity measure that is most suited for the problem at hand.

2.2 Constraints on Image Representation

Consider a binary recognition problem² with a training set consisting of I example images per class. The formulation of this problem as one of statistical classification can be based on two alternative image representations. The first, the holistic representation, makes \mathcal{X} the space of all images and represents each image as a point in this space. Since images are high-dimensional, downsampling is always required to guarantee a space of manageable dimensionality. Typical image sizes after downsampling are on the order of 20×20 pixels, i.e. $\dim(\mathcal{X}) \approx 400$. On the other hand, localized representations are based on a collection of local measurements (or *features*) extracted from the image. For example, the image can be broken down into a collection (*bag*) of small neighborhoods, e.g. 8×8 pixels, and \mathcal{X} made the space of such neighborhoods, $\dim(\mathcal{X}) = 64$. Each image no longer corresponds to a single point in \mathcal{X} , but to a collection of points.

While the dependence on the training examples only through their inner products is, theoretically, a very appealing feature of the SVM architecture, it also introduces a significant computational burden that places serious constraints on the image representations compatible with it. In particular, because SVM learning is an optimization problem with coefficients given by the entries of the kernel matrix, its complexity is quadratic in the size of the training set. Hence, if the localized representation originates K neighborhoods per image, this implies a $O(K^2 I^2)$ complexity and a K^2 -fold increase over the complexity associated with the holistic representation. As we will see in section 4, it is not difficult for the localized representation to originate on the order of 5,000 neighborhoods per image, corresponding to a 25×10^6 -fold increase in computation that is always undesirable and usually intractable. Furthermore, under the SVM formulation, there is no way to capture the natural grouping of image neighborhoods into images, i.e. the fact that the goal is to classify bags of examples instead of the examples independently. For these reasons, the localized representation is not suitable for traditional SVM-based recognition.

While the holistic representation has been successful for recognition [13,3,14] it should not be taken for granted that it is inherently better than its localized counterpart. On the contrary, it suffers from the following problems.

- **Resolution:** when images are severely downsampled a significant amount of information is lost. While this information may not be important for the classification of images far away from the classification boundary, it can be

² The discussion in this section generalizes to any number C of classes, since a C -way classifier can be implemented as a combination of C binary (one-vs-all) classifiers.

quite relevant to distinguish the ones that are close to it. Since the latter determine the classification error, low resolution can have an impact on recognition error. The best example of this phenomena are current state-of-the-art face detectors [13,1]. While visual inspection of the errors committed by the classifier, at the low-resolution on which its decisions are based, reveals that it is quite hard to distinguish between faces and non-faces, a significant percentage of those errors becomes clear at full image resolution.

- **Invariance:** when images are represented as points in \mathcal{X} , a relatively simple image transformation can send the point associated with an image to another point that is significantly far away in the Euclidean sense. In fact, when subject to transformations, images span manifolds in \mathcal{X} which can be quite convoluted and the correct distance for classification is the distance to these manifold. While the kernel function can, in principle, encode this, the traditional SVM formulation provides no hints on how to learn the kernel from examples. This can lead to significant invariance problems.
- **Occlusion:** since, for the holistic representation, occlusion originates a (possibly dramatic) change in some of the components of vector associated with the image to classify, an occluded pattern can, once again, be quite distant from the unoccluded counterpart. Unlike invariance, it is not even clear that occlusion leads to an image manifold (there could be creases, folds, or singularities in the space of occluded images) and it is therefore even less clear what metric, or kernel, would be appropriate to deal with occlusion.

Note that the localized representation does not place constraints on resolution (larger images simply generate more neighborhoods), and is significantly more invariant and robust to occlusion.

3 Probabilistic Kernels Based on the KL-Divergence

Since there are advantages to the localized representation, enabling the SVM architecture to support it is a relevant problem for visual recognition. This is the motivation behind the KL-kernel that we briefly review in this section. We then show that it 1) enables truly discriminant localized representations, and 2) can be naturally adapted to each classification problem. This allows the derivation of various kernels tailored for representations previously proposed for recognition.

3.1 The KL-Kernel

The combination of SVMs and localized visual representations is related to that of SVMs and data sequences, a topic that has been addressed by various authors [7,11,8,12,15]. Since the role of the kernel is to capture similarities between examples, and sequences are naturally described by their probability densities, one idea that has recently received some attention is to replace the sequences by their probabilistic descriptions [7,11,8,12]. This has various advantages, including the ability to 1) deal with sequences of variable lengths, 2) rely on a compact

sequence representation, and 3) exploit prior knowledge about the classification problem (through the selection of probability models) [7]. The KL-kernel is the extension of the standard Gaussian kernel to this family of probabilistic kernels: while the Gaussian is proportional to the negative exponent of the weighted Euclidean distance between two vectors, the KL-kernel is the negative exponent of the symmetric KL divergence [16]. This divergence is a measure of distance between two densities and has various interesting connections to the geometry of the manifold of probability distributions [17]. In particular, given densities $p(\mathbf{x})$ and $q(\mathbf{x})$, the KL-kernel is

$$KLK = \exp^{-a\mathcal{J}[p(\mathbf{x}), q(\mathbf{x})] + b}, \quad (4)$$

where $\mathcal{J}(p(\mathbf{x}), q(\mathbf{x})) = KL(p(\mathbf{x}), q(\mathbf{x})) + KL(q(\mathbf{x}), p(\mathbf{x}))$ is the symmetric KL divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$,

$$KL(p(\mathbf{x}), q(\mathbf{x})) = \int_{-\infty}^{\infty} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \quad (5)$$

the KL divergence between the two densities, and a and b constants [8].

3.2 A Kernel Taxonomy

One of the main attractives of probabilistic kernels is a significant enhancement of the flexibility of the SVM architecture. For example, the KL-kernel can be tailored to a classification problem by either 1) matching it to the statistics of the datasets under consideration, 2) taking advantage of approximations to the KL-divergence that have been shown to work well in certain domains, or even 3) combining feature and kernel design. In this section we give some examples of such tuning, but various other kernels could be derived in a similar fashion.

Parametric densities. There are many problems where the class-conditional densities are known, or can be well-approximated, by parametric densities. In these cases (5) can usually be simplified. One common setting is for the densities to be members of a parametric family, such as the popular *exponential family*

$$p(\mathbf{x}|\theta) = \alpha(\mathbf{x}) \exp[a(\theta) + \mathbf{b}(\theta)\mathbf{c}(\mathbf{x})], \quad (6)$$

which includes densities such as Gaussian, Poisson, Binomial, Beta, among various others [18]. The KL-divergence between two such densities is

$$KL(p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)) = a(\theta_i) - a(\theta_j) + [\mathbf{b}(\theta_i) - \mathbf{b}(\theta_j)]^T E_{\theta_i}[\mathbf{c}(\mathbf{x})] \quad (7)$$

where E_{θ_i} is the expectation with respect to $p(\mathbf{x}|\theta_i)$. One case of significant interest is that of the Gaussian density,

$$p(\mathbf{x}|\{\mu, \Sigma\}) = \mathcal{G}(\mathbf{x}, \mu, \Sigma) = \frac{1}{2\pi^{d/2}|\Sigma|} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right\} \quad (8)$$

for which (7) becomes

$$KL(\mathcal{G}(\mathbf{x}, \mu_i, \Sigma_i), \mathcal{G}(\mathbf{x}, \mu_j, \Sigma_j)) = \\ \frac{1}{2} \log \frac{|\Sigma_j|}{|\Sigma_i|} - \frac{d}{2} + \frac{1}{2} \text{tr}(\Sigma_j^{-1} \Sigma_i) + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \quad (9)$$

where d is the dimensionality of the \mathbf{x} . Since image data is not always well-approximated by densities in the exponential family, other probabilistic models are also used in the recognition literature. One popular model is the histogram, $\pi = \{\pi_1, \dots, \pi_b\}$, where π_i are estimates for the distribution of the feature probability mass over a partition of the feature space \mathcal{X} defined by a collection of non-overlapping cells, or bins, $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_b\}$. The KL-divergence between two histograms, $\pi^i = \{\pi_1^i, \dots, \pi_b^i\}$ and $\pi^j = \{\pi_1^j, \dots, \pi_b^j\}$, defined on \mathcal{C} is

$$KL(\pi^i, \pi^j) = \sum_{k=1}^b \pi_k^i \log \frac{\pi_k^i}{\pi_k^j} \quad (10)$$

and extensions to the case where the two histograms are defined on different partitions, \mathcal{C}_i and \mathcal{C}_j , are also available [9]. There are, nevertheless, models for which a closed-form solution to the KL-divergence does not exist. In these cases it is necessary to resort to approximations or sampling methods.

Approximations and sampling. One popular approximation to the KL-divergence consists of linearizing the log around $x = 1$, i.e. $\log(x) \approx x - 1$. It is straightforward to show [10] that, under this approximation, the KL-divergence becomes the χ^2 statistic, a function that has been frequently proposed as a measure of histogram similarity [19]. For other models, the χ^2 approximation can still be quite difficult to compute in closed form. One such case is the popular Gaussian mixture and various approximations to the KL-divergence between Gaussian mixtures have been recently proposed in the literature, including 1) the *log-sum bound* [20], 2) the *asymptotic likelihood approximation* [9], and 3) approximations based on the *unscented transformation* [21]. Our experience is that, while these approximations tend to work rather well for ranking images by similarity, they do not always provide an approximation that is sufficiently tight for the purpose of evaluating the KL-kernel. An alternative that, at the cost of increase computation, eliminates this problem is a Monte-Carlo approximation

$$KL[p(\mathbf{x}|\theta_i), p(\mathbf{x}|\theta_j)] \approx \frac{1}{s} \sum_{m=1}^s \log \frac{p(\mathbf{x}_m|\theta_i)}{p(\mathbf{x}_m|\theta_j)} \quad (11)$$

where $\mathbf{x}_1, \dots, \mathbf{x}_s$ is a sample drawn according to $p(\mathbf{x}|\theta_i)$.

4 Experiments and Results

We conducted a detailed experimental evaluation of the performance of the KL-kernel for recognition. The Columbia object database, COIL-100 [2], was the

source of data for these experiments. It consists of 100 classes, each containing 72 views of an object, obtained by rotating the object at 5° clockwise over 360° . All images have resolution of 128×128 pixels and 24-bit RGB color.

4.1 Holistic versus Localized Representation

To evaluate the impact of image resolution on the performance of the various classifiers, we created replicas of COIL-100 at three resolutions: 32×32 , 64×64 , and 128×128 pixels by downsampling and converting all images to grayscale. To test invariance, we created from each database 4 different combinations of train/test sets, following [3]: for each image class, I images were set aside as a training set, by sampling the view angle uniformly, the remaining ones being used for testing. As in [3], we considered $I \in \{4, 8, 18, 36\}$. We refer to the dataset with $I = n$ as \mathcal{D}_n . In all experiments, the holistic representation was obtained by scan-converting each image into a vector. For the localized representation the image was transformed into a bag of 8×8 neighborhoods (obtained by shifting a 8×8 window by increments of two pixels horizontally and vertically). The discrete cosine transform (DCT) of each window was then computed and scanned into a vector of 64 features ordered by frequency of the associated DCT basis function. Only the 32 lowest frequency DCT coefficients were kept. This is a standard procedure that enables speeding-up the estimation of the density associated with each image without compromising classification performance.

Results. The performance of the holistic representation was evaluated with traditional SVMs based on three different kernels: linear (L-SVM), polynomial of order 2 (P2-SVM), and Gaussian (G-SVM) [5]. The localized representation was evaluated with both a standard maximum-likelihood Gaussian mixture model (GMM) classifier and the KL-kernel using GMMs as probability models (KL-SVM). We used mixtures of 32 Gaussians in the former case and of 16 in the latter. Classification rates for all resolutions and datasets \mathcal{D}_n are shown in Table 1. The best result for each combination of resolution/number of training images is shown in bold. These results support various interesting conclusions.

First, among the holistic kernels, G-SVM was consistently the best. Its performance is excellent when the number of training vectors is large, $I = 36$, achieving the best results of all classifiers tested. However, as the number of training examples decreases, the recognition rate drops significantly. In fact, for values of I other than 36, it is usually even inferior to that of the non-discriminant GMM classifier. While this may appear surprising, it underlines one of the points of section 2.2: *that the localized representation is inherently more invariant than the holistic one, therefore leading to simpler classification problems*. Due to this, the weaker classifier (GMM) outperforms the stronger one (SVM) when there are less views of each object in the training set and, therefore, the ability to generalize becomes more important. On the other hand, as expected, the combination of the localized representation with a discriminant classifier (KL-kernel SVM) outperforms that of the localized representation with a generative classifier (GMM). Overall, the *KL-kernel SVM achieves the best performance of all*

Table 1. Recognition rate (in %) for the various classifiers discussed in the text.

	Resolution 32×32				Resolution 64×64				Resolution 128×128			
	\mathcal{D}_4	\mathcal{D}_8	\mathcal{D}_{18}	\mathcal{D}_{36}	\mathcal{D}_4	\mathcal{D}_8	\mathcal{D}_{18}	\mathcal{D}_{36}	\mathcal{D}_4	\mathcal{D}_8	\mathcal{D}_{18}	\mathcal{D}_{36}
L-SVM	67.24	82.67	92.98	97.31	67.54	82.84	92.85	97.39	67.85	82.80	92.89	97.50
P2-SVM	63.02	80.03	93.09	98.11	62.27	79.11	92.30	97.89	62.53	77.78	92.85	97.58
G-SVM	72.79	88.67	96.85	99.78	75.75	90.80	97.78	99.68	75.54	90.13	97.04	99.17
GMM	76.41	91.05	96.30	97.83	80.82	90.27	94.89	95.31	82.48	90.89	94.72	94.89
KL-SVM	79.56	93.20	97.32	98.28	83.69	94.36	98.89	98.83	84.32	95.22	98.65	98.67

methods by combining the higher invariance of the localized representation with the better classification performance of discriminant methods.

Regarding the impact of resolution on classification rate, the tables also support some interesting observations. The first is that the performance of G-SVM is approximately constant across resolutions. This is remarkable since, for the holistic representation, 128×128 corresponds to a 16,384 dimensional feature space. The fact is that, as resolution increases, the classification performance is subject to a tug-of-war between the nefast consequences of the curse of dimensionality and the benefits of added image information. For the holistic SVM these effects cancel out and performance is approximately constant. The localized representation, on the other hand, does not suffer from any increase in the dimensionality (only more vectors per image) and only has to benefit. Hence, *the gain in recognition rate of KL-SVM over G-SVM increases with image resolution*. For the hardest problems considered ($I = 4$) the decrease in error rate was as large as 36%. Once again, this underlines the points of section 2.2.

4.2 The Flexibility of the KL-Kernel

Given that the most discriminant visual attributes for recognition depend on the recognition task (e.g. while shape might achieve the best results for digit recognition, texture is a better cue for discriminating outdoor scenes) a general-purpose classifier should support multiple image representations. As discussed in section 3.2, the flexibility of the KL-kernel makes it very attractive from this point of view. In this section, we evaluate the performance on COIL-100 of its combination with previously proposed representations for recognition, in particular, representations based on color, appearance, and joint color and appearance. Color-histograms were initially proposed for recognition in [22] and are today commonly used for object recognition and image retrieval [19]. Histogram similarity is frequently measured with the histogram intersection metric, which is equivalent to the L_1 distance between the histograms [22]. In the SVM context, this metric has been proposed as a kernel for visual recognition by [23], and denoted by *Laplacian kernel*. We compared its performance with that of the χ^2 approximation to the KL-divergence, a popular approximation for histogram-based recognition. For modeling local appearance we used the representation of the previous section (DCT coefficients of the luminance channel for appearance alone, DCT coefficients of the three color channels for joint color and appear-

Table 2. Recognition rate (in %) of classifiers based on different visual cues: color, appearance, and joint color and appearance.

	histogram-based	local appearance	global appearance
grayscale	χ^2 kernel: 71.72 Laplacian kernel: 69.90	KL-SVM: 84.32	G-SVM: 75.54
color	χ^2 kernel: 98.12 Laplacian kernel: 97.81	KL-SVM: 96.74	G-SVM: 84.90

ance). For global appearance we used the holistic representation. To jointly model color and global appearance we concatenated the vectors from the three color channels into a vector three times larger.

Results. All experiments were based on 128×128 images and dataset \mathcal{D}_4 . Color histograms were computed with $16 \times 16 \times 16$ bins, gray-level histograms with 16 bins. For joint color and local appearance, the DCT coefficients were interleaved into a 192 dimensional vector of which only the first 64 dimensions were used for density estimation. Table 2 presents a comparison of the recognition rates. The first interesting observation from this table is the importance of color as a cue for recognition on COIL, since all representations perform significantly better when color is used. Interestingly, in this case, the extremely localized histogram representation (features of pixel support) beats the less-localized (8×8 supported) appearance-based counterpart and both significantly outperform the holistic representation. This illustrates the trade-off between localization and invariance at an extreme level: *because color is so discriminant, even the invariance loss associated with the small 8×8 neighborhood is sufficient to degrade recognition performance. The invariance loss of the holistic representation is so large that its performance is not even close to those of the localized representations.* Note that the point is not to claim that the color histogram is the ultimate solution for object recognition. In fact, it would likely not perform as well if, for example, there were more objects with similar colors in the database. The point is that different visual attributes are most discriminant for different databases, and less discriminant attributes require representations of larger spatial support (which allow modeling *configurations* of features therefore increasing the discriminant power). However, larger support usually implies less invariance (since the manifolds spanned by the configurations are increasingly more complex) and the result is a trade-off between discriminant power and invariance. In table 2 the best value for this trade-off is achieved by the localized representation, for grayscale images, and by the histogram-based one, when color is used. The conclusion is that, even for a given classification problem, the optimal representation can vary depending on factors such as the composition of the database, its size, the visual features that can be reliably extracted, etc. In this sense, the ability of the KL-kernel to support a diversity of representations can be a great asset.

A second interesting observation is to compare the results in the table with those obtained by Roth et al. They used shape as the cue for recognition and

proposed two representations. One based on explicit encoding of the position of pixels in the object contour, the second based on conjunctions of edges. The first achieved a rate of 81.46, i.e. superior only to the combination of the KL-kernel with the grayscale histogram, and the grayscale G-SVM. The second achieved a rate, 88.28, slightly superior to the grayscale KL-SVM kernel, and superior to the two holistic SVM representations, but clearly inferior to any of the KL-SVM kernels using color. Again, these results highlight the importance of different representations for different databases. While color does not produce a winner when combined with holistic appearance, it completely shatters the performance of shape when combined with any of the localized representations. On the other hand, shape appears to be more discriminant than appearance in the absence of color. This suggests that it would be interesting to have a shape-based kernel for the KL-SVM, an area that we are now exploring.

References

1. P. Viola and M. Jones. Robust Real-Time Object Detection. In *2nd International Workshop on Statistical and Computational Theories of Vision*, 2001.
2. H. Murase and S. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.
3. D. Roth, M. Yang, and N. Ahuja. Learning to Recognize Three-Dimensional Objects. *Neural Computation*, 14:1071–1103, 2002.
4. M. Weber, M. Welling, and P. Perona. Unsupervised Learning of Models for Recognition. In *European Conf. on Computer Vision*, pages 18–32, Dublin, Ireland, 2000.
5. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
6. H. Schneiderman and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina*, 2000.
7. T. Jaakkola, M. Diekhans, and D. Haussler. Using the fisher kernel method to detect remote protein homologies. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, August 1999.
8. P. J. Moreno, P. P. Ho, and N. Vasconcelos. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications in *Proc. of NIPS 2003*
9. N. Vasconcelos. On the Efficient Evaluation of Probabilistic Similarity Functions for Image Retrieval, *IEEE Transactions on Information Theory*, to appear
10. N. Vasconcelos. A Unified View of Image Similarity. In *Proc. Int. Conf. Pattern Recognition*, Barcelona, Spain, 2000.
11. K. Tsuda, M. Kawanabe, G Ratsch, S. Sonnenburg, and K. Muller. A New Discriminative Kernel from Probabilistic Models. *Neural Computation*, 14(10):2397–2414, 2002.
12. R. Kondor and T. Jebara. A kernel between sets of vectors. In *International Conference on Machine Learning*, 2003.
13. H. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998.
14. B. Moghaddam and M. Yang. Gender Classification with Support Vector Machines. In *4th IEEE Int'l Conference on Automatic Face & Gesture Recognition*, 2000.
15. L. Wolf and A. Shashua. Kernel Principal Angles for Classification Machines with Applications to Image Sequence Interpretation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, Madison, Wisconsin.

16. S. Kullback. *Information Theory and Statistics*. Dover, New York, 1968.
17. D. Johnson and S. Sinanovic. Symmetrizing the Kullback-Leibler Distance. *IEEE Transactions on Information Theory*, March 2001. Submitted.
18. R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
19. J. Puzicha, Y. Rubner, C. Tomasi, and J. Buhmann. Empirical Evaluation of Dissimilarity Measures for Color and Texture. In *International Conference on Computer Vision, Korfu, Greece*, pages 1165–1173, 1999.
20. Y. Singer and M. Warmuth. Batch and On-line Parameter Estimation of Gaussian Mixtures Based on Joint Entropy. In *Neural Information Processing Systems, Denver, Colorado*, 1998.
21. J. Goldberger, S. Gordon, and H. Greenspan. An Efficient Image Similarity Measure based on Approximations of the KL-Divergence Between Two Gaussian Mixtures. In *International Conference on Computer Vision*, 2003.
22. M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, Vol. 7(1):11–32, 1991.
23. O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, September 1999.

Partial Object Matching with Shapeme Histograms*

Y. Shan, H.S. Sawhney, B. Matei, and R. Kumar

Computer Vision Laboratory

Sarnoff Corporation

201 Washington Road, Princeton, NJ 08540

{yshan, hsawhney, bmatei, rkumar}@sarnoff.com

Abstract. Histogram of shape signature or prototypical shapes, called shapemes, have been used effectively in previous work for 2D/3D shape matching & recognition. We extend the idea of shapeme histogram to recognize partially observed query objects from a database of complete model objects. We propose to represent each model object as a collection of shapeme histograms, and match the query histogram to this representation in two steps: (i) compute a constrained projection of the query histogram onto the subspace spanned by all the shapeme histograms of the model, and (ii) compute a match measure between the query histogram and the projection. The first step is formulated as a constrained optimization problem that is solved by a sampling algorithm. The second step is formulated under a Bayesian framework where an implicit feature selection process is conducted to improve the discrimination capability of shapeme histograms. Results of matching partially viewed range objects with a 243 model database demonstrate better performance than the original shapeme histogram matching algorithm and other approaches.

1 Introduction

The effectiveness of using semi-local shape signature, like spin image and shape context, for 2D and 3D shape matching and recognition has been demonstrated in previous work. Shapemes are defined as clusters of shape signatures that correspond to different parts on the objects. Histograms of shapemes characterizing shape distributions is a compact, albeit rich descriptor ideal for rapid and accurate shape matching. Readers can refer to the next section for a detailed description of shapeme histogram. Normally, matching shapeme histograms of two objects requires both objects to be complete. It would not work properly if the query object is only a part of the model object, e.g., the query is a range image that covers only a limited portion of an object. To address this problem, we propose to divide a model into smaller parts and compute shapeme histograms for each of them. Matching a query histogram with a model involves two steps: (i) find the parts on the model object that correspond to the query object, and (ii) match the

* This work was supported in part by DARPA/AFRL Contract No. F33615-02-C-1265

shapeme histogram computed from those parts to the histogram of the query object. The first step is formulated as a constrained optimization problem where the goal is to find the optimal projection of the query histogram onto the subspace spanned by all the shapeme histograms of the model. We then propose a sampling based algorithm to solve this optimization problem efficiently. The second step is expressed into a Bayesian framework where an implicit feature selection process is conducted to improve the discrimination capability of shapeme histograms.

The proposed method provides a general framework that can be used for matching partial objects of any kind with shapeme histograms. In this paper, we are interested in the application of finding 3D models in a database that are similar to an object presented in a range image. The 3D model objects and the query range images are both represented as 3D point clouds. The point clouds for a model object cover the complete object, whereas the query range images provide only partial views of a query object. The goal is to find a short list of model objects that are the closest matches to a query object.

2 Related Work

Our basic representation is based on the idea of shapeme and shapeme histogram, proposed by Mori, Belongie and Malik [1] for 2D object recognition. A shapeme is a prototype shape feature that represents a cluster of similar invariant features. These features are computed at basis points that are densely sampled on the 2D object, and hence a single object may contain many features. Each feature on an object is assigned to its closest shapeme. Counting the frequency of shapemes over the whole object gives the shapeme histogram of the object. Fig. 1 shows an example. Shapeme histogram is used by [2] and [3] for texture recognition. In this paper, we extend shapeme histogram based approach to handle the matching of partially observed objects. Moreover, we also propose to select shapemes that are critical to a given task and embed this process into a Bayesian matching framework.

The shapeme histogram for 3D objects can be constructed likewise in terms of 3D shape features and the associated 3D shapemes. Invariant, semi-local shape features such as splash feature [4], spherical harmonic [5], and spin image [6] have been developed in the past few years for matching and recognizing 3D objects. Among these features, we have chosen spin image representation because it has been widely used and reported to be robust against clutter and occlusion [7,8]. It should be noted that though selecting an optimal invariant feature is always important, it is not critical to the points that we want to make in this paper. More details about the spin image representation can be found in [6].

Figure 2 is the picture showing the relationships of the original shapeme histogram method and our proposed method with respect to other recognition approaches. Obviously, the most accurate method is to align the objects being matched [4,6,9], because it takes into account the global geometric constraints. The alignment-based approach is expensive, and has difficulty when aligning non-rigid objects. The nearest neighbor approach without global geometric constraints can be quite accurate when using semi-local features such as spin images.

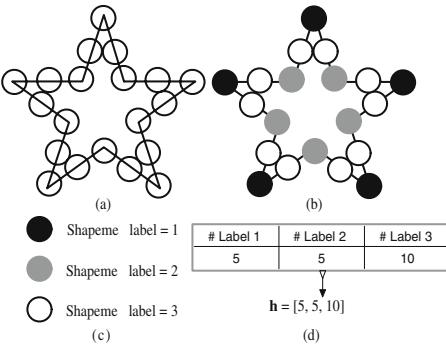


Fig. 1. Shapeme histogram. (a) A star-shaped object and the 20 basis points (the circles) where 2D invariant features are computed. (b) The same object with each feature point labeled with a set of 3 shapemes as in (c). (d) The shapeme histogram of the object in both table and vector formats. Since there are 5 #1, 5 #2, and 10 #3 labels on the object, respectively, the shapeme histogram of this object is $\mathbf{h} = [5, 5, 10]$

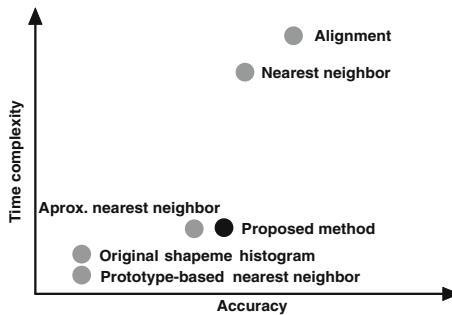


Fig. 2. The proposed method with respect to other approaches for partial object recognition. The alignment-based method is the most accurate but expensive approach. Our method is fast and accurate, and can ideally be used as a model pruning front-end for the alignment-based approach. See the text for more details

The problem is that it is too slow especially when the feature dimension is high and the database is large. The approximate nearest neighbor approach [10] and the prototype-base nearest neighbor approach [11,12] are employed to address this problem. However, the performance of the former depends on the type of feature and the distribution of the features in the database, while the latter is in general not accurate. Shapeme histogram is a fast method because both the model and the query are represented with single vectors. However, the original shapeme histogram approach does not handle a partial query. This justifies the use of our proposed approach, which is fast and accurate even when the query object is partially observed. An ideal application of our method is to use it as a

model pruning front-end where the goal is to find a short list of model objects that can be verified by more sophisticated and accurate approaches.

3 Histogram Projection

We now introduce the histogram projection idea to address the problem of matching the histogram of a partial query against a complete model. As a illustration example, consider an image matching problem. Figure 3 shows an original image (image A) and another image (image B) consisting of only four segments from the original image. Obviously the intensity histogram computed from image A looks different from image B. Given the intensity histogram \mathbf{h}_i^s for each segment i of A, and the histogram \mathbf{h}^q of B, the problem is to find the set \mathcal{S} of all the segments such that $\mathbf{h}^q = \sum_{k \in \mathcal{S}} \mathbf{h}_k^s$.

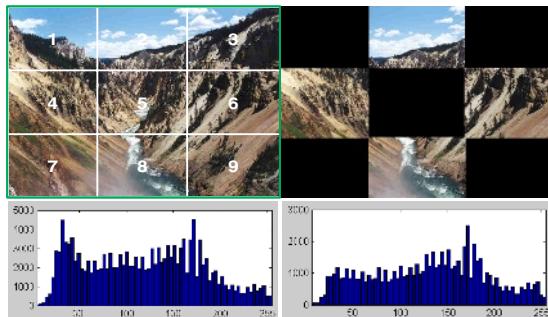


Fig. 3. An example of using histogram projection for 2D image matching. The left column shows the original image and its intensity histogram. The image is segmented into 9 pieces and each of them is labeled with a number. The right column shows an image consisting of only 4 pieces of the original image. The corresponding intensity histogram is computed without including the black areas

To solve the problem, let $\mathbf{A} = [\mathbf{h}_1^s, \mathbf{h}_2^s, \dots, \mathbf{h}_9^s]$ be the $l \times 9$ matrix, where l is the number of bins of each histogram, and $\mathbf{x} \in \mathbb{R}^9$ is an unknown vector. Suppose that $\text{rank}(\mathbf{A}) = 9$, solving the linear system $\mathbf{A}^T \mathbf{A} \mathbf{x} - \mathbf{A}^T \mathbf{h}^q = 0$ gives us the solution $\mathbf{x} = [0, 1, 0, 1, 0, 1, 0, 1, 0]^T$. We now conclude that $\mathcal{S} = \{2, 4, 6, 8\}$, i.e., image B consists of these numbered segments from image A.

In summary, we have projected \mathbf{h}^q onto the subspace spanned by \mathbf{h}_i^s , and the projection is \mathbf{Ax} , which in this case is exactly the same as \mathbf{h}^q . Note that this is just an ideal example where the query tiles are the same as the tiles used to construct the histogram subspace.

4 Shapeme Histogram Projection for Partial 3D Object Recognition

This section elaborates the idea in the previous section, and proposes a shapeme histogram projection approach for matching partially viewed 3D objects. A schematic description of the approach is given in Fig. 4. Blocks 1, 2, and 3 are covered in this section, and block 4 is covered in the next section. In the following discussion, model objects are assumed to be complete.

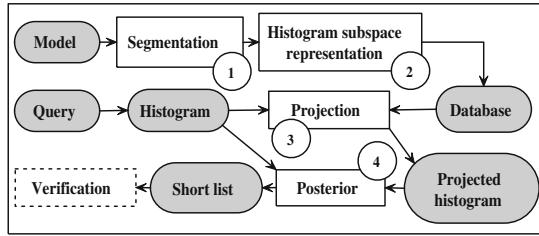


Fig. 4. Shapeme histogram projection for partial 3D object recognition. Model objects are segmented in block 1 and component based histogram are computed to form a subspace representation in block 2. The component histograms are added into a histogram database. Query histogram is computed and projected in block 3 onto the subspace spanned by the component model histograms in the database. Posteriors are computed in block 4 based on the query histogram and its projected histogram, as described later in Sec.5 (14). A short list of matching model objects are generated for optional verification

4.1 Spatial Object Segmentation

In Block 1, a set of spin image \mathcal{B}_i is computed for object M_i at basis points \mathcal{P}_i . Each spin image $\mathbf{b}_{i,j} \in \mathcal{B}_i$ is associated with a 3D basis point $\mathbf{p}_{i,j} \in \mathcal{P}_i$. The spatial segmentation algorithm performs k-means clustering in the basis point set \mathcal{P}_i and returns n clusters of 3D points, where n is a predetermined number. The set of spin images \mathcal{B}_i is split into n groups accordingly. Note that this process should not be confused with clustering in the spin image space, where the goal is to find the prototype features, the shapemes. Figure 5 shows two views of a segmented pickup truck.

4.2 Histogram Subspace Representation

In Block 2, once the spin images of object M_i have been separated into n groups, a shapeme histogram \mathbf{h}_i^s can then be computed for each group. The histogram subspace representation is then the $m \times n$ matrix \mathbf{A} , as mentioned in the previous section, where $\mathbf{A} = [\mathbf{h}_1^s, \mathbf{h}_2^s, \dots, \mathbf{h}_n^s]$, and m is the number of shapemes. This

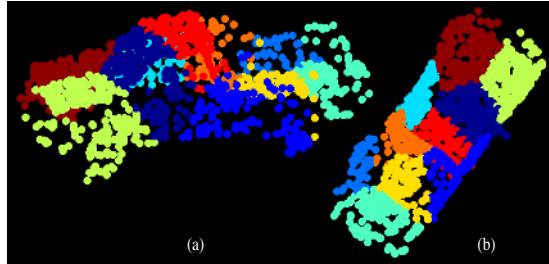


Fig. 5. Spatial object segmentation of a pickup truck (best viewed with color). The circular points represent the 3D basis points where the spin images are computed. Basis points from different spatial segments are labeled with different color. (a) A side view of the segmented object. (b) The top view

matrix is added into the model database. Obviously, as compared with the approaches that put all the raw features inside the database, the saving in storage space with the shapeme histogram representation is huge.

4.3 Shapeme Histogram Projection

In Block 3, given a query shapeme histogram \mathbf{h}^q , its projection (the closest approximation) in the subspace spanned by the i th model histograms can be computed as $\hat{\mathbf{h}}^q = \mathbf{A}_i \mathbf{x}$, where \mathbf{A}_i is the \mathbf{A} matrix of the i th model, and \mathbf{x} can be computed as

$$\text{minimize} \|\mathbf{A}_i \mathbf{x} - \mathbf{h}^q\|_2 \quad \text{subject to } \mathbf{x} \in \{0, 1\}^n. \quad (1)$$

The solution of the unconstrained version of (1) is the solution of the linear system $\mathbf{A}_i^T \mathbf{A}_i \mathbf{x} - \mathbf{A}_i^T \mathbf{h}^q = 0$, if \mathbf{A}_i has full column rank. This is what we used to solve the ideal image matching problem in Fig. 3. In the case when the query histogram is noisy, or the query range image cuts through significant portions of some segments, the constraint $\mathbf{x} \in \{0, 1\}^n$ can be used to “regularize” the solution. Solving (1) with exhaustive search requires 2^n operations, which is tractable only when n is small. When n is large, a Gibbs sampler [13] is employed to explore the binary solution space more efficiently. The Gibbs distribution that corresponds to the objective function in (1) is

$$G(\mathbf{x}) = \frac{1}{Z} \exp[-(\|\mathbf{A}_i \mathbf{x} - \mathbf{h}^q\|_2)/T], \quad (2)$$

where Z is an unknown normalization factor, and T is the temperature constant. Because x is binary, the local conditional probability (or the local characteristic function) can be derived easily from (2) as

$$\begin{aligned} G(x_j = 0 | \{x_k | k \neq j\}) \\ = \frac{G(x_j = 0, \{x_k\})}{G(\{x_k\})} \end{aligned}$$

$$\begin{aligned}
&= \frac{G(x_j = 0, \{x_k\})}{\sum_{x_j \in \{0,1\}} G(x_j, \{x_k\})} \\
&= \frac{1}{1 + G(x_j = 0, \{x_k\})/G(x_j = 1, \{x_k\})}, \tag{3}
\end{aligned}$$

where x_k is the k th coordinate of \mathbf{x} . Note that the unknown factor Z is canceled out in (3). Given a random initial guess, the sampler sweeps through each coordinate x_k sequentially, and flips its value according to the local conditional probability in (3). The computational cost in each step is negligible since only one coordinate is touched, and the actual cost $\mathbf{A}_i \mathbf{x} - \mathbf{h}^q$ can be computed incrementally. The same process is repeated for several iterations, and the \mathbf{x} that has the smallest $G(\mathbf{x})$ is selected as the solution. Because our objective function is simple, and the dimension is relatively small ($n < 100$), Gibbs sampler can quickly converge to a solution very close to the global optimum. In fact, we find that when $n = 10$, Gibbs sampler gives the identical result as the exhaustive search.

5 A Bayesian Framework for Shapeme Histogram Matching

This section provides a Bayesian framework for the shapeme histogram matching. It explains why under some mild assumptions, matching shapeme histogram is equivalent to computing the posterior of a model given the query. More importantly, it reveals that an implicit feature selection process is naturally embedded when matching under the proposed framework. To simplify the discussion, we will lay out the framework based on the assumption that both the query object, and the model object are complete, and then apply it to a partial query using the histogram projection approach proposed in the previous section.

5.1 Notations

Let M_i be the i th model object, and Q the query object. The problem is to find a set \mathcal{C} of candidate model objects with the largest posteriors $P(M_i | Q)$, such that $\sum_{i \in \mathcal{C}} P(M_i | Q) \geq \epsilon$, where $0 \leq \epsilon \leq 1$ is a predefined value close to 1. For each model object M_i , a set of spin images $\mathcal{B}_i = \{\mathbf{b}_{i,1}, \mathbf{b}_{i,2}, \dots, \mathbf{b}_{i,t_i}\}$ is computed at a number of basis points $\mathcal{P}_i = \{\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \mathbf{p}_{i,t_i}\}$ densely sampled along the surface of the object, where $\mathbf{b}_{i,\cdot} \in \mathcal{R}^d$, $\mathbf{p}_{i,\cdot} \in \mathcal{R}^3$, d is the spin image dimension, and t_i is the number of spin images in the object. For all the model objects under consideration, a set of shapemes $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is computed from the full set of spin images $\Omega = \bigcup_{j=1}^o \mathcal{B}_j$ by clustering in the spin image space, where m and o are the number of shapemes and model objects, respectively. For each shapeme \mathbf{a}_k , a median radius r_k is computed during the clustering process. Median radius is the median distance from the center of the shapeme (cluster) to the spin images that belong to this shapeme (cluster).

Each model object M_i is then represented by a shapeme histogram $\mathbf{h}_i = [u_{i,1}, u_{i,2}, \dots, u_{i,m}]^T$, where the number of bins m is the same as the number of

shapemes in \mathcal{A} , and $u_{i,k}$ is the number of spin images that have been labeled with the k th shapeme. The probability of each shapeme \mathbf{a}_k given a model object M_i is then given by

$$P(\mathbf{a}_k | M_i) = u_{i,k} / u_i , \quad (4)$$

where $u_i = \sum_k u_{i,k}$ is the total number of labeled spin images in the object. The query is represented by the set of all spin images computed from the object.

5.2 Model Posterior Given a Query

The posterior probability of a model given a query can be computed as

$$\begin{aligned} & P(M_i | Q) \\ &= \sum_{\mathbf{a}_k} P(\mathbf{a}_k | Q) P(M_i | \mathbf{a}_k, Q) \end{aligned} \quad (5a)$$

$$\approx \sum_{\mathbf{a}_k} P(\mathbf{a}_k | Q) P(M_i | \mathbf{a}_k) , \quad (5b)$$

where (5b) is an approximation of (5a) assuming that $P(M_i | \mathbf{a}_k, Q) \approx P(M_i | \mathbf{a}_k)$, that is when both a shapeme and the query are given, the probability of M_i is determined only by the shapeme. The same assumption is used in [11], and is coined as the “homogeneity assumption”. Applying Bayes’ rule to (5b) leads to

$$P(M_i | Q) \approx \sum_{\mathbf{a}_k} P(\mathbf{a}_k | Q) \frac{P(\mathbf{a}_k | M_i) P(M_i)}{\sum_{M_l} P(\mathbf{a}_k | M_l) P(M_l)} . \quad (6)$$

Assuming that all models are equally likely, we have

$$P(M_i | Q) \approx \sum_{\mathbf{a}_k} P(\mathbf{a}_k | Q) \frac{P(\mathbf{a}_k | M_i)}{\sum_{M_l} P(\mathbf{a}_k | M_l)} . \quad (7)$$

The first term in (7) is the likelihood of a shapeme conditioned on the query, and can be computed as

$$\begin{aligned} & P(\mathbf{a}_k | Q) \\ &= \sum_{\mathbf{b}_l} P(\mathbf{a}_k | \mathbf{b}_l, Q) P(\mathbf{b}_l | Q) \end{aligned} \quad (8a)$$

$$\approx \sum_{\mathbf{b}_l} P(\mathbf{a}_k | \mathbf{b}_l) P(\mathbf{b}_l | Q) \quad (8b)$$

$$\propto \sum_{\mathbf{b}_l} P(\mathbf{a}_k | \mathbf{b}_l) \quad (8c)$$

$$= \sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l | \mathbf{a}_k) P(\mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l | \mathbf{a}_g) P(\mathbf{a}_g)} \quad (8d)$$

$$= \sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l | \mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l | \mathbf{a}_g)} , \quad (8e)$$

where (8b) is obtained as in (5b) with the homogeneity assumption, \mathbf{b}_l is the l th spin image of Q . Note that the raw feature likelihood conditioned on the query $P(\mathbf{b}_l | Q)$ is assumed to be uniformly distributed in Eqs. (8b)–(8c), and $P(\mathbf{a}_g)$ is assumed to be uniformly distributed in Eqs. (8d)–(8e). Substituting (8e) into (7), $P(M_i | Q)$ is proportional to

$$\sum_{\mathbf{a}_k} \sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l | \mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l | \mathbf{a}_g)} \frac{P(\mathbf{a}_k | M_i)}{\sum_{M_l} P(\mathbf{a}_k | M_l)}, \quad (9)$$

where $P(\mathbf{b}_l | \mathbf{a}_k)$ is the likelihood of a spin image in the query conditioned on a shapeme. This likelihood is defined as

$$P(\mathbf{b}_l | \mathbf{a}_k) = \begin{cases} 1 & \mathbf{a}_k = \arg \min_{\mathbf{a}_g} D(\mathbf{a}_g, \mathbf{b}_l) \\ 0 & \text{otherwise} \end{cases}, \quad (10)$$

where $D(\cdot, \cdot)$ is the distance between two features. Eq. (10) says that if \mathbf{a}_k is the shapeme that is the closest to the spin image \mathbf{b}_l , the likelihood of \mathbf{b}_l is one. This is spin image labeling based on a nearest neighbor vector quantization approach. Since the query object may be noisy, and obscured by scene clutter, a restricted labeling process defined as follows is preferred.

$$P(\mathbf{b}_l | \mathbf{a}_k) = \begin{cases} 1 & D(\mathbf{a}^*, \mathbf{b}_l) \leq 2.5 r^* \\ 0 & \text{otherwise} \end{cases}, \quad (11)$$

where \mathbf{a}^* is the shapeme closest to \mathbf{b}_l , and r^* is the median radius of the shapeme as defined in Sec. 5.1. Spin images severely corrupted by noise or scene clutter may not belong to any shapeme. By simply deleting them from the query spin image set, probabilities computed based on the remaining spin images become more reliable. It is then obvious from both (10) and (11) that always $\sum_{\mathbf{a}_g} P(\mathbf{b}_l | \mathbf{a}_g) = 1$. More generally,

$$\sum_{\mathbf{b}_l} \frac{P(\mathbf{b}_l | \mathbf{a}_k)}{\sum_{\mathbf{a}_g} P(\mathbf{b}_l | \mathbf{a}_g)} = v_k, \quad (12)$$

where v_k is the number of spin images in the query that are labeled by \mathbf{a}_k . Finally, substituting (4) and (12) into (9) leads to

$$P(M_i | Q) \propto \sum_{k=1}^m \left(\frac{v_k u_{i,k}}{u_i} \gamma_k^{-1} \right), \quad (13)$$

where $\gamma_k = (\sum_{M_l} P(\mathbf{a}_k | M_l))/o$ is a normalization factor that counts for the discrimination capability of the k th shapeme, and o is the number of model objects. It is obvious that (13) represents the correlation between the model and the query, with each bin scaled by γ_k^{-1} . The normalization factor γ_k can be regarded as the average probability of the shapeme \mathbf{a}_k 's occurrence within the whole model database. Eq. (13) gives high weights to shapemes that are rare in the database since they have small γ_k . This represents a feature selection process where shapemes that belong to many model objects are down weighted. γ is similar to the $tf \times idf$ weight [14] used widely in the document analysis community.

5.3 Posterior for Projected Histograms of a Partial Query

When the query is partially observed, we can use the method from Sec. 4 to find its projection $\hat{\mathbf{h}}_i = [\hat{u}_{i,1}, \hat{u}_{i,2}, \dots, \hat{u}_{i,a}]^T$ in the subspace spanned by all the component histograms of model M_i . This histogram corresponds to a virtual model \hat{M}_i that matches the observed part of the query object. The model posterior can then be computed as

$$P(\hat{M}_i | Q) \propto \sum_{k=1}^m \left(\frac{v_k \hat{u}_{i,k}}{\hat{u}_i} \gamma_k^{-1} \right). \quad (14)$$

Strictly speaking, γ_k should also be computed according to the virtual models corresponding to the query object. In practice, we find it sufficient to compute γ_k from all the complete model objects in the database.

6 Experimental Results

We have tested our method with a model set of 243 vehicles on a 2GHz PC with 2GB main memory. We will present comparative results of our proposed method against other methods. We used 500 prototype features for all the prototype-based methods and the shapeme histogram methods. We observed in our experiments that increasing the number of prototype features beyond 500 did not lead to a significant improvement in the recognition accuracy. The following is a list of the analyzed approaches.

1. The proposed histogram projection method, denoted as HP, is described in Sec. 4.3. For the Gibbs sampler, we use 30 iterations (n steps per iteration, where n is the number of segments of each model), and set the temperature const $T = 2$.
2. Nearest neighbor method, denoted as NN, matches each spin image in the query object with all the spin images in the database. A vote for that model is incremented whose spin image is the closest to the query spin image. The total matching score for each model object is computed by w_i/w , where w_i is the number of votes the i th model collects, and w is the total number of scene points.
3. Locality-sensitive hashing method, denoted as LSH, uses the same voting mechanism as in NN, but uses an approximated nearest neighbor search strategy while speeding up the search. We use 8 hash tables. See [10] for details of the LSH method.
4. Prototype-based method, denoted as PR, is detailed in [11].
5. Prototype-based subspace method, denoted as SPR, is detailed in [12]. We use a 3 dimensional subspace.

6.1 Noise Free Data Results

A laser range sensor simulator is used to convert facet models into range images. View points are sampled from a hemisphere centered around the object, and the

viewing direction is always targeting the center. The spherical coordinate for each view point is denoted as (r, ϕ, θ) , where r is the radius, ϕ is the azimuthal angle, and θ is the polar angle. The radius is set to be a constant such that the object occupies the full view of the simulated laser sensor for most of the times. It is therefore ignored in the view point notation hereafter. Each model object is generated by combining 8 pieces of point clouds sampled from the view point set of $\{(0, 45^\circ), (45^\circ, 45^\circ), \dots, (360^\circ, 45^\circ)\}$ that covers the whole object. By constructing an octree-like spatial structure from the combined point cloud of an object M_i , a set of basis points \mathcal{P}_i is uniformly selected from the (implicit) object surface, and the corresponding set of spin images \mathcal{B}_i is computed (see Sec. 5.1 for notations). A set of 500 shapemes \mathcal{A} is computed from the set Ω of all spin images from the 243 models. Each object M_i is then segmented into n pieces, and a histogram h_i^s is computed for each piece. The model histogram database is then constructed as described in Sec. 4. The query set contains also 243 views, one for each object. Each view is randomly sampled from $\phi \in \{10^\circ, 20^\circ, 30^\circ\}$, and $\theta \in \{40^\circ, 45^\circ, 50^\circ\}$, respectively, and covers 20–60% of the complete object.

Table 1 compares the recognition accuracy of our method against the others, where the accuracy is given with respect to the number of models in the short candidate list. As compared with other HP methods, the one with $n = 30$ components produces the best result. If we represent each model as a single histogram, i.e., $n = 1$, the HP method degenerates into the original histogram matching method. The bad result is expected since histogram matching in its original form does not handle partially observed queries. It can be also observed that increasing n from 10 to 30 does not improve the accuracy as dramatic as from 1 to 10. In fact, the HP method with $n = 50$ produces almost the identical result as $n = 30$.

From Table 1, it can also be seen that the LSH method does not work well for this set of database. On the other hand, the accuracy of the HP method with $n = 30$ and three candidates is close to the nearest neighbor method, and is much better than other approaches.

Table 2 compares the time and storage space used by each method. The time reported is the average time for a single query, while the space is for the whole

Table 1. Recognition accuracy of our approach as compared with other methods. HP-1, HP-10, and HP-30 are the histogram projection methods with $n = 1$, 10, and 30, respectively. HP-1 is the original shapeme histogram method

Methods	NN	LSH	HP-1	HP-10	HP-30	PR	SPR
Accuracy with top 1 model (%)	97.0	85.2	78.5	87.2	91.1	78.0	84.0
Accuracy with top 3 models (%)	1.0	93.0	88.9	94.8	97.2	87.0	91.0

Table 2. Time and storage space used by each method

Methods	NN	LSH	HP-1	HP-10	HP-30	PR	SPR
Time (sec)	240	6	1.7	2.7	3.8	1.5	4.1
Space (MB)	480	500	0.6	6	18	1	720

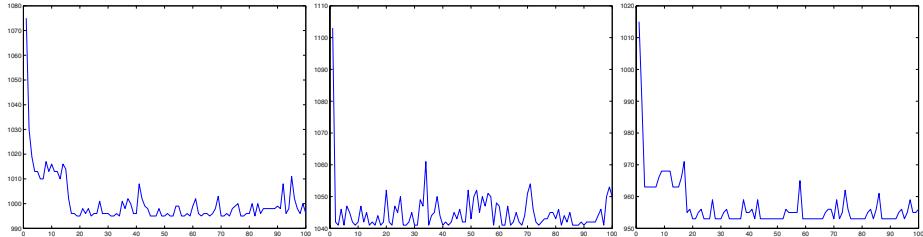


Fig. 6. Snapshots of the iterative sampling process for histogram projection of three queries to their corresponding models. The x -axis is the number of iterations (0 – 100), and the y -axis is the value of the objective function in (3). Thanks to the rapid convergence of the process, the best solution can usually be found within 30 iterations

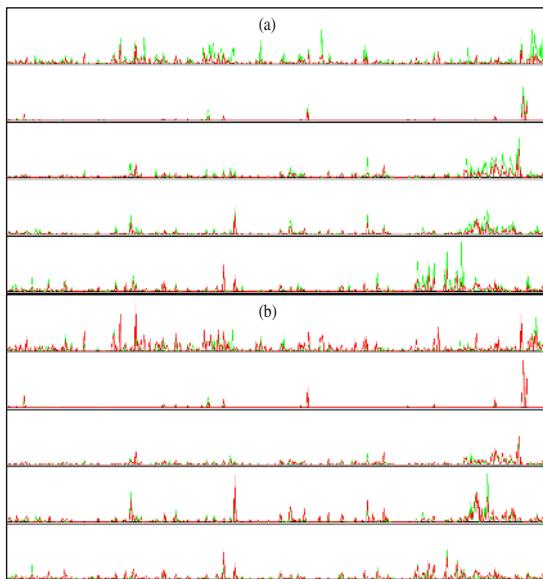


Fig. 7. The accuracy of the projected histograms (best viewed with color). (a) Model histograms (green) overlaid with query histograms (red) (b) Projected histograms (green) overlaid with query histograms (red). The projected histograms are much more similar to the query histograms

model database. It can be seen that the speed of the HP method is in general slower than the PR method, but is still more than 60 times faster than the nearest neighbor approach. It is also faster than the LSH method. It can also be observed from the table that the HP method requires only a small fraction of the storage space needed by both the NN method and the LSH method.

One interesting aspect from the table is that the sampling based histogram projection process does not bring in much overhead in terms of the computational time. This is because our Gibbs sampler usually converges within 30 iterations,

Table 3. Accuracy of the HP method with different level of synthesized structural noise. The larger the λ , the more noise is added into the query

λ	0.0	0.1	0.2	0.3	0.4	0.5
Accuracy with top 1 (%)	91.1	89.0	87.3	84.0	76.5	70.1

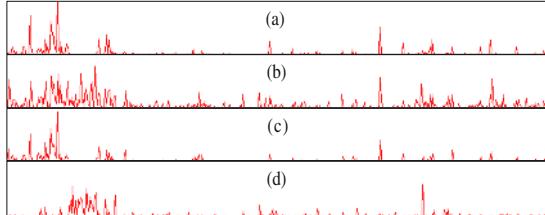


Fig. 8. Projection of a query histogram with large structural noise. See text for details

and each iteration involves a trivial update of the objective function. Fig. 6 shows some snapshots of this iterative convergence process. Fig. 7 compares query histograms with the corresponding complete model histograms, and the projected histograms. Obviously, the projected version is more similar to the query.

6.2 Data with Synthesized Clutter and Noise

To test the performance of our method under scene clutter and noise, we added to each query histogram h_i^q a percentage of another query histogram h_j^q , $i \neq j$, and form a new query $\tilde{h}_i^q = h_i^q + \lambda h_j^q$, which now contains a certain degree of “structural” noise from another query object. This is a much more difficult test than just adding random noise or unstructured clutter into the query. We varied λ from 0.1 to 0.5 in the test and the result for the HP method ($n = 30$) is shown in Table 3. It can be seen that the accuracy of the method decreases gracefully as the level of noise increases. Note here the decrease in accuracy is mainly caused by the confusion between the projected histogram and the query histogram. In most cases, we observed that the projected histograms were still very accurate. To see this, Figure 8 shows an example where (b) is the sum of (c) and (d), i.e., $\lambda = 1$. If we project (b) onto the histogram subspace of the model corresponding to (c), we get the projection (a), which indeed looks similar to (c).

7 Conclusion

We have proposed a two-step approach to address the problem of shapeme histogram matching with partially observed objects. We have applied the proposed method to the problem of 3D object recognition with range image. We then compared our method with other approaches and demonstrated its advantages on a commercially available database of 243 models.

References

1. Mori, G., Belongie, S., Malik, J.: Shape contexts enable efficient retrieval of similar shapes. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. (2001) 723–730
2. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision (IJCV01) **43** (2001) 29–44
3. Lazebnik, S., Schmid, C., Ponce, J.: Affine-invariant local descriptors and neighborhood statistics for texture recognition. In: International Conference on Computer Vision(ICCV03). (2003)
4. Stein, F., Medioni, G.: Structural indexing: Efficient 3-D object recognition. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), **14** (1992) 125–145
5. Kazhdan, M., Funkhouser, T.: Harmonic 3D shape matching. In: Technical Sketch, SIGGRAPH (2002). (2002)
6. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), **21** (1999) 433–449
7. Ruiz-Correa, S., Shapiro, L.G., Meila, M.: A new signature-based method for efficient 3-d object recognition. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. (2001) 769–776
8. Ruiz-Correa, S., Shapiro, L.G., Meila, M.: A new paradigm for recognizing 3-d object shapes from range data. In: International Conference on Computer Vision(ICCV03). (2003)
9. Chen, C.S., Hung, Y.P., Cheng, J.B.: Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), **21** (1999) 1229–1234
10. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: The VLDB Journal. (199) 518–529
11. Kuncheva L.I., Bezdek, J.C.: Presupervised and postsupervised prototype classifier design. IEEE Transactions on Neural Networks **10** (1999) 1142–1152
12. Chien, J.T.: Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **24** (2002) 1644–1649
13. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI) **6** (1984) 721–741
14. Salton, G.: Developments in automatic text retrieval. Science **253** (1991) 974–980

Modeling and Synthesis of Facial Motion Driven by Speech

Payam Saisan¹, Alessandro Bissacco¹, Alessandro Chiuso², and Stefano Soatto¹

¹ University of California, Los Angeles - CA 90095

saisan@ee.ucla.edu, {bissacco,soatto}@cs.ucla.edu

² University of Padova - Italy 35131

chiuso@dei.unipd.it

Abstract. We introduce a novel approach to modeling the dynamics of human facial motion induced by the action of speech for the purpose of synthesis. We represent the trajectories of a number of salient features on the human face as the output of a dynamical system made up of two subsystems, one driven by the deterministic speech input, and a second driven by an unknown stochastic input. Inference of the model (learning) is performed automatically and involves an extension of independent component analysis to time-dependent data. Using a shape-texture decompositional representation for the face, we generate facial image sequences reconstructed from synthesized feature point positions.

1 Introduction

Human facial motion carries rich information that we use to interact: We constantly read cues from people's faces, conveying a wide range of useful information often altering our state. While facial motion in isolation is quite interesting, the coupling with the action of speech adds yet another dimension to the problem. Our goal is to understand the dynamic behavior of facial motion as it relates to speech, and infer a model that can be used to generate synthetic sequences of images driven by speech. A great challenge in this task is the evolutionary acuity of human perception to details of the face and facial motion. For a facial model to meet this high standard, we must devise models that can capture subtleties. While there has been remarkable progress in the area of speech content recognition and general facial motion based on speech utterances [10,2,3], there remains an open question of capturing dynamic complexities and interactions between facial motion and speech signals. Such subtleties are encoded largely in the dynamics of facial motion as opposed to static pose geometry and photometry.

The problem is simple to state. We want to collect motion-capture data¹ for an individual, and the associated speech waveform, and from these data build a model that can be used to generate novel synthetic facial motions associated with novel speech segments, for instance for an animated character. However, we want to be able to do this while retaining the "distinctive character" of the

¹ In particular, trajectories of a collection of feature point positions in space.

individual person in the training set. For instance, if we observe Mr. Thompkins says “happy birthday,” our long term goal is to develop a model that can be used to synthesize novel facial motions that “looks” like Mr. Thompkins’.

The rationale of our approach is based on the fact that facial motion is the result of word utterances combined with physical characteristics of the face that are peculiar to each individual.

2 Relation to Previous Work and Contribution of This Paper

The topic of speech-driven facial animation has been the subject of considerable attention recently. A scheme for modifying emotional attributes of facial motion, such as happiness or anger, associated with utterances is discussed in [7]. In [10] Ezzat et al. propose a variant of the multidimensional morphable model as a representation for images, particularly effective in describing a set of images with local variations in shape and appearance. He uses this representation to develop a statistical interpolation technique, in the space of morphable models, to interpolate novel images corresponding to novel speech segments. In [2] Brand introduces the idea of driving the facial model with a related control signal derived from the speech signal. He introduces a modified hidden Markov model for identification of non-stationary piecewise linear systems. He uses this model to approximate the nonlinear behavior of the face via “quasi-linear” submanifolds. In [3], Bregler et. al propose an image-based method called “Video Rewrite.” This method relies on constructing audiovisual basic building blocks called triphones. It uses a large amount of training data to construct a basis for the entire utterance space. By identifying the correct audio-visual building blocks corresponding to a novel speech utterance and concatenating them it forms image sequences corresponding to novel speech segments. Unlike the past work on constructing generic facial motion synthesizers, we are interested in utilizing the information in speech to capture and drive a facial motion that is realistic and closer to the speaker’s personal dynamic character. Our goal is not to demonstrate a model that spans the entire utterance space, but at this stage to develop the concept and demonstrate its efficacy using only a small set of samples.

Our model decouples the deterministic dynamics driven by speech from the stochastic dynamics driven by samples from a stochastic process with unknown and non-Gaussian distribution. We show how to perform inference of this model, which involves independent component analysis (ICA) [8] applied to a dynamic context. We apply our inference algorithm to a model of a face based on decoupling transformations of the domain of the image from transformation of the intensity values, akin to so-called “active appearance” or “linear morphable models” [9,10,1]. However, unlike traditional active appearance models, we do not require manual selection and registration of interest points, but instead perform the learning automatically. Unlike [17], we do not use a pre-defined grid, but instead we use a geometric structure defined by salient regions of the images where geometric deformations are well-defined.

3 Modeling

In this section we describe a model that is motivated by the considerations above. We first describe the decoupling of appearance and motion, and then the decoupling of speech-driven motion, and noise-driven motion.

3.1 Modeling the Face: Shape and Radiance Decomposition

We make the assumption that a face is a smooth parameterized surface $S : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$, supporting a diffuse albedo $\rho : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^+$, moving and deforming under the action of a group² $g(t)$, viewed under perspective projection $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, so that a given point $p = S(x)$ generates an image I at pixel $w(x, t)$ at time t according to

$$I(w(x, t), t) = \rho(x, t) \quad \forall x \in \Omega \quad (1)$$

where we have defined the “domain warping” $w(x, t) \doteq \pi(g(t)S(x))$. Without loss of generality we can assume that Ω corresponds to the image-plane at time $t = 0$. Note that the actual shape of the surface, i.e. the quotient $S/g(t)$, cannot be untangled from the deformation $g(t)$ in $w(x, t)$, and from the deformed radiance $\rho(x, t)$ and therefore the “responsibility” of modeling changes in radiance due to shape deformations is shared between the domain warping w and the range transformations ρ . Estimating the two infinite-dimensional functions $w : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^2$ and $\rho : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^+$ in the case of a general scene is very complex, and falls into the general domain of *deformable templates* [13]. Here, we do not have a general scene, but various deformations of the same face due to speech. Therefore, in the spirit of active appearance models [9], we assume that local variability of the domain can be modeled as linear transformations of a number of basis elements:

$$w(x, t) = w_0(x) + W(x)y(t); \quad x \in \Omega, t = 1, 2, \dots \quad (2)$$

where $W = [W_1, \dots, W_{k_w}]$; $W_i : \Omega \rightarrow \mathbb{R}^2$ are basis elements, $y(t) \in \mathbb{R}^{k_w} \forall t$. In “active appearance models”, one assumes that equation (2) is satisfied *not* on all of Ω , but only at a fixed number of known (often manually selected and registered) “landmark” points x_1, \dots, x_l . Then $W(x_i)$, $i = 1, \dots, l$ can be estimated using principal component analysis (PCA). In [17], x_i are fixed points on a pre-defined grid, so no manual selection of landmarks is necessary. However, whether $w(x, t)$ in (1) can be inferred at a point x depends on the values of I in a neighborhood of $w(x, t)$. If x falls in a region of constant radiance, $w(x, t)$ is not well defined, which can result in unlikely domain deformations being estimated.

In this work, we adopt an intermediate approach, where we evaluate (2) only at photometrically distinct points, modeling the deformation of all and only the points where the deformation can be inferred. However, we rely on

² The deformation here is represented by a complex and possibly infinite-dimensional group. We will use a simpler model, which we will elucidate shortly.

the fact that we have a sequence of adjacent views of the image from video to automatically detect and track such photometrically distinct points and maintain point registration. We use a standard point tracker (our implementation of Lucas and Kanade's [20]) to track and obtain trajectories of a number of points on the face and thus the associated shape parameters $y(t)$. The process, including facial image synthesis, is further elaborated in section 5.

In the next subsection we discuss how to model the temporal evolution of these parameters. If the number of available points is small, we could bypass the dimensionality reduction of the deformation w and simply model the trajectory of all the landmark points $\{x_1(t), \dots, x_l(t) \in \mathbb{R}^2\}_{t=1, \dots, T}$. In either case, we call the "state" of interest $y(t)$, the latter case corresponding to $W = I$.

3.2 Modeling the Dynamics of the Face

In this section we model the temporal evolution of $\{y(t)\}_{t=1, \dots, T}$. As we mentioned in the introduction, such evolution compounds the effect of deterministic speech and a more ephemeral input that is not associated with speech characteristics. Here we are interested in *decoupling* these effects.

One possible way is to assume that $y(t)$ is in fact the sum of two components $y(t) = y_d(t) + y_s(t)$, the first generated by a, say, linear system driven by the input sound channels $u(t)$, while the second, $y_s(t)$, generated through a linear system driven by an IID random process $e(t)$ with unknown distribution p_e , independent of $u(t)$. This kind of philosophy has been introduced in the area of subspace identification in [18] and further elaborated upon in [4,6,5].

Assuming that the dynamics of the "deterministic" and "stochastic" models are disjoint, one can give a state space description in decoupled form as follows. We introduce hidden "states" ξ , that we partition into two components: $\xi = [\xi_d^T, \xi_s^T]^T$, a "deterministic" one ξ_d that receives input from the sound channels $u(t)$, and a "stochastic" one ξ_s that receives input from an IID random process $e(t)$ with unknown distribution p_e .

While we have reasons to believe that the dynamics of facial motion can be faithfully modeled with a linear model (faces usually do not exhibit nonlinear behaviors such as limit cycles, bifurcations, or chaos, at least for the majority of individuals), in order to model the subtleties associated to each individual we allow the stochastic input to be drawn from a non-Gaussian distribution p_e . The model we consider, therefore, is in the following decoupled form

$$\begin{aligned} \begin{bmatrix} \xi_d(t+1) \\ \xi_s(t+1) \end{bmatrix} &= \begin{bmatrix} A_d & 0 \\ 0 & A_s \end{bmatrix} \begin{bmatrix} \xi_d(t) \\ \xi_s(t) \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ B_s \end{bmatrix} e(t) \\ y(t) &= [C_d \ C_s] \begin{bmatrix} \xi_d(t) \\ \xi_s(t) \end{bmatrix} + D_d u(t) + e(t) \end{aligned} \quad (3)$$

where $e(t) \stackrel{IID}{\sim} p_e$; We assume that the model above is stable and has *minimum phase* ($|\lambda(A_s)| < 1$, $|\lambda(A_d)| < 1$, $|\lambda(A_s - B_s C_s)| < 1$, where λ denotes the largest eigenvalue), and that $e(t)$ is a (strict sense) white process³. Further-

³ I.e. $e(t)$ and $e(s)$ are indepedent for $t \neq s$

more, we assume that there exists a (square invertible) matrix D so that the components of

$$v(t) \doteq D^{-1}e(t) = [v_1(t), \dots, v_{k_w}(t)]^T \quad (4)$$

are *independent* with density function $q_i(\cdot)$ ⁴. In the next section we argue that there are procedures to chose the dimension of the states ξ_d, ξ_s , but we shall not discuss this point in the paper. Note that we assume that the dynamics are decoupled (off-diagonal blocks of the transition matrix are zero). This is in the spirit of the so-called Box-Jenkins model, well known in the system identification literature [16]. The goal of the inference process (learning) is, given a sequence of measured trajectories $\{y(t)\}_{t=1,\dots,T}$, to estimate the states $\{\xi_d(t), \xi_s(t)\}$, the model parameters $A_d, A_s, B_d, B_s, C_s, C_d, D_d$, the mixing matrix D and the non-Gaussian density of the stochastic input q . While the first part (identification of a model in decoupled form) has been studied in the system identification literature [18,6,5], dealing with (and estimating) a non-Gaussian driving noise is a non-standard task, which we discuss in the next section.

Once the model is identified, we can generate synthetic sequences by feeding the model with a speech input, and samples from the density q , as we explain in section 5.

4 Inference

In this section we discuss how to identify the model parameters and estimate the states of the model (3). Despite the linear structure, the model does not fall in the standard form suitable for applying off-the-shelf system identification algorithms, due to (a) the decoupled structure of the input-to-state relationship and (b) the non-Gaussian nature of the stochastic input. We will address these problems separately in the following subsections.

4.1 Combined Identification of the Model

In this section we concentrate on the identification of the model (3), following the approach proposed in [18,6,5]. Under a technical assumptions called “absence of feedback” (see Granger [12]) the stochastic processes y_d and y_s , called the *deterministic* and the *stochastic component* of y , defined by the conditional expectations

$$y_d(t) \doteq E[y(t) \mid u(t), u(t-1), \dots, u(t-k), \dots] \quad y_s(t) \doteq y(t) - y_d(t) \quad (5)$$

are uncorrelated at all times [18]. It follows that $y(t)$ admits an orthogonal decomposition as the sum of its deterministic and stochastic components

$$y(t) = y_d(t) + y_s(t) \quad E[y_s(t)y_d(\tau)^T] = 0 \quad \text{for all } t, \tau.$$

⁴ Note that, if $y(t)$ is a full-rank purely non-deterministic process $e(t)$ has the same dimension k_w as $y(t)$.

Note that y_d is actually a *causal* linear functional of the input process, and is hence representable as the output of a causal linear time-invariant filter driven only by the input signal u . Consequently, $y_s(t)$ is also the "causal estimation" error of $y(t)$ based on the past and present inputs up to time t . Its input-output relation has the familiar form $y = F(z)u + G(z)v$ with "stochastic" and "deterministic" transfer functions $F(z) = C_d(zI - A_d)^{-1}B_d + D_d$ and $G(z) = I + C_s(zI - A_s)^{-1}B_s$.

Up to this point there is no guarantee that combining a state space realization of $F(z)$

$$\begin{aligned}\xi_d(t+1) &= A_d\xi_d(t) + B_d u(t) \\ y_d(t) &= C_d \xi_d(t) + D_d u(t)\end{aligned}\quad (6)$$

and one of $G(z)$

$$\begin{aligned}\xi_s(t+1) &= A_s \xi_s(t) + B_s e(t) \\ y_s(t) &= C_s \xi_s(t) + e(t)\end{aligned}\quad (7)$$

yielding (3) results in a minimal model (i.e. with the minimum number of state components).

In most practical cases the stochastic and deterministic dynamics will be completely different, and hence (3) will be minimal.

A subspace identification procedure based on this decomposition has been introduced in [18] and later refined and analyzed in a series of papers by the same authors [4,6,5] and can be summarized as follows. Using available data $\{y(t), u(t), t = 1, \dots, T\}$:

1. Estimate the deterministic component $\hat{y}_d(t) \doteq E[y(t) \mid u(1), u(2), \dots, u(T)]$. (see [18,4,6,5] for details)
2. Use a standard "deterministic" subspace identification technique to identify the system parameters A_d, B_d, C_d, D_d . (see [6,5] for details)
3. Estimate the stochastic component $\hat{y}_s(t) \doteq y(t) - \hat{y}_d(t)$
4. Prefilter the stochastic component with a filter constructed from the identified deterministic system to compensate for a certain distortion due to the fact that only finite data are available (see [4] for details).
5. Use the prefiltered data as an input to the algorithm in [21] to estimate the stochastic parameters A_s, C_s, B_s .

The subspace procedures used in step (2.) and (5.) provide also order estimation techniques which allow to suitable choose the dimension of the states ξ_d and ξ_s , we refer to [4] for details.

4.2 Isolating the Stochastic Part: Revisiting "Dynamic ICA"

From the identification step we obtain a minimum-phase realization of the stochastic component of $y(t)$:

$$\begin{aligned}\xi_s(t+1) &= A_s \xi_s(t) + B_s D v(t) \\ y_s(t) &= C_s \xi_s(t) + D v(t).\end{aligned}\quad (8)$$

where $v(t)$, defined in equation (4), has independent components.

A predictor $\hat{y}_s(t|t-1)$ for the system output at time t is a (in general non-linear) function of the data up to time $t-1$, $\hat{y}_s(t|t-1) = f(y_s(t-1), \dots, y_s(t-k), \dots)$ that is designed to approximate the output $y_s(t)$ according to some criterion. The optimal predictor, in the sense for instance of minimum variance of the estimation error $y_s(t) - \hat{y}_s(t|t-1)$, is the conditional mean $\hat{y}_s(t|t-1) = E[y_s(t)|y_s(t-1), \dots, y_s(t-k), \dots]$. Under our assumptions (i.e. $v(t)$ strictly white and $(A_s - B_s C_s)$ stable) the predictor is just given by the inverse system of (8), which is named the “innovation model”. The process $e(t) = Dv(t)$ is the “innovation process”, i.e. the (optimal) one-step-ahead prediction error:

$$\begin{aligned}\hat{\xi}_s(t+1) &= (A_s - B_s C_s) \hat{\xi}_s(t) + B_s y_s(t) \\ e(t) &= y_s(t) - C_s \hat{\xi}_s(t) = y_s(t) - \hat{y}_s(t|t-1).\end{aligned}\quad (9)$$

At the same time, we want to enforce the constraint that the components of $v(t) = D^{-1}e(t)$, are independent; this can be done by minimizing the relative entropy (Kullback-Liebler divergence) between the joint density of $v(t)$ and the product of the densities $q_i(\cdot)$ of its components $v_i(t) \doteq D_{\cdot i}^{-1}e(t)$:

$$\min_{D, q_i} K \left(|\det(D)| p_e(y(t) - \hat{y}(t|t-1)) \middle\| \prod_{i=1}^{k_w} q_i(D_{\cdot i}^{-1}e(t)) \right) \quad (10)$$

where $D_{\cdot i}^{-1}$ denotes the i -th row of the matrix D^{-1} and $K(p||q) \doteq \int \log \frac{p}{q} dP(x)$. This problem can be considered a dynamic extension of independent component analysis (ICA), a problem that has been addressed both in the literature of blind deconvolution using higher-order statistics [11] and in the learning literature [14, 22]. In particular, [11] estimates the parameters of a non-minimum phase system via blind deconvolution based on high-order cumulants.

Our assumption that the innovation are temporally strictly white allows to solve the dynamic ICA problem rather easily. Of course a more general model would not assume that the optimal prediction is linear or alternatively that the (linear) one step prediction errors are independent. The recent work [22] addresses the problem above, both for the case of non-linear models and for linear models, by using gradient descent algorithms. In the case of minimum-phase models as in (8), this approach do not fully exploit the structure of the (linear) problem. Therefore such a gradient procedure, when successful, cannot do better than a simple algorithm [19] that consists in a closed-form algorithm for identifying the model parameters, followed by a static ICA to whiten the components of the input. A similar approach has been advocated in [14].

In fact, since the system is assumed to be minimum phase and the inputs $e(t)$ temporally strictly white, as we have argued above, the optimal predictor is linear and depends only on second order properties on the process $y_s(t)$. The parameters A_s, C_s, B_s , can be recovered using linear system identification techniques. Subspace identification procedures as the ones previously described solve this problem and are particularly suited to work with high dimensional data (i.e. k_w large).

After the innovation model has been estimated, standard (static) ICA can be used to estimate a mixing matrix D and the density function $q(\cdot)$ from the

residuals $e(t) = y_s(t) - \hat{y}_s(t|t-1)$. This method was first proposed in [19] as being suboptimal. As we have argued, with the hypothesis we make here, it is actually optimal.

The reader may ask what happens if the assumptions made in (8) are not satisfied. If the model is non-linear, then we know no better than running a large optimization problem in the fashion of [22]. If the model is non-minimum phase, our solution will yield the closest minimum-phase model, in the sense of minimum variance. Alternatively one can identify the model using high-order cumulants as in [11].

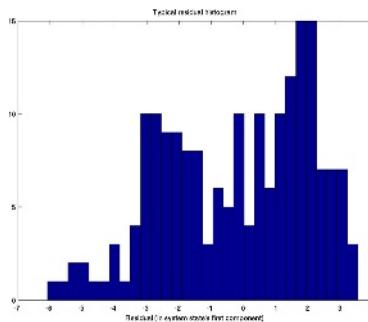


Fig. 1. Typical example of a residual histogram (sample approximation of q). Although the sample pool (number of frames) is small, the non-Gaussian nature of the distribution is clear.

5 Experiments

Face data were obtained using a 60Hz camera and tracking points on the lower region of the face for 200-300 frames. An implementation of the Shi-Tomasi feature tracker [20,15] was developed for this purpose.

We modeled face images using shape and radiance elements as in [9]. The shape element $s = (x_1, x_2, \dots, x_l) \in \mathbb{R}^{2l}$ is defined by vertex coordinates (tracked points) of an n -point triangular mesh encompassing the face. Associated with every $s(t)$ is the supporting face albedo (texture), $\rho(x, t)$, such that $I(x, t) = \rho(x, t)$ where I is the face image at pixel x . To obtain sensible configurations of points to encompass the lower part of the face around the mouth and to reduce the number of outliers, we guided the feature selection by providing an image mask defining the regions to select features from. Figure 2 shows the configuration of tracked points on the subject's face. For every utterance sequence we obtained a training data set $s(t)$ and corresponding $\rho(x, t)$. Speech data was extracted from the synchronized audio signal. We used 256 periodogram coefficients as representation for speech segments corresponding to individual video frames, and we PCA reduced the dimensionality to arrive at $u(t) \in \mathbb{R}^4$. The choice of dimension here was a design parameter adjusted for best results.



Fig. 2. Standard tracking schemes [20] were used to track feature point positions over 200-300 frames, sampled at 60 frames per second. About 200 points were selected in the first frame and tracked throughout the sequence.

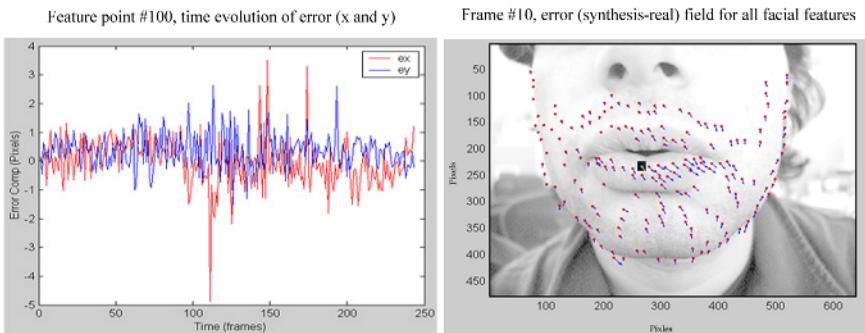


Fig. 3. Typical examples of error plots, feature position discrepancies between synthesis and actual data obtained via cross validation. The right figure is the the time evolution of error (discrepancy between synthesized feature motion vs. actual motion) for feature point number 100, a typical point near the center where fast and error prone motions occur. The left is the error vector field (synthesis-data) for all the points for a typical frame. The error for all feature points remained small for other frames in the sequence as depicted in this example.

Given $s(t)$ data we obtained the PCA reduced shape parameters $y(t)$ representing the output of the system. Following the inference procedure of section 4.1 we first identified the deterministic system parameters A_d, B_d, C_d, D_d using $y(t)$ as output and $u(t)$ as input. Then we identified the stochastic subsystem parameters as outlined in 4.2. As part of this step we get the non-Gaussian histograms corresponding to independent components of $v(t) = D^{-1}e(t)$ which is later used to generate, by sampling from the distribution, the random input driving the stochastic subsystem.

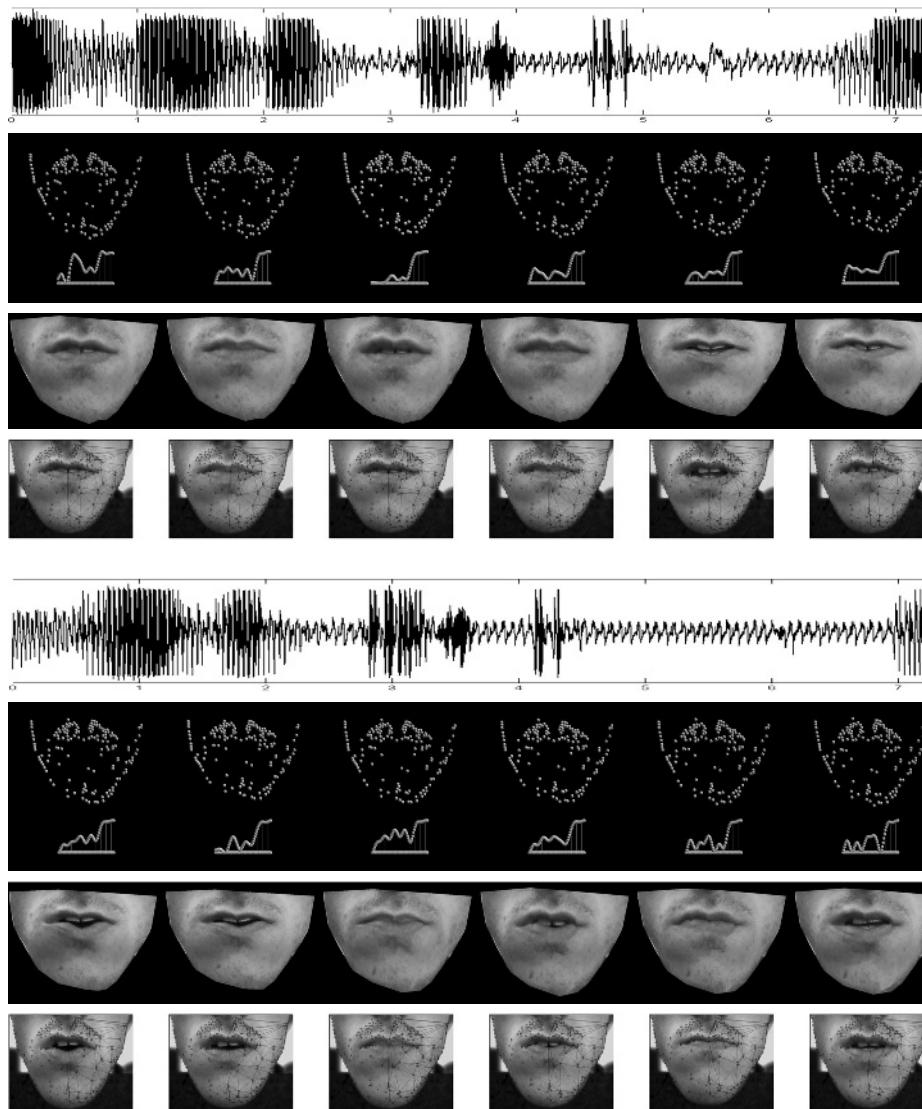


Fig. 4. An example of facial motion driven by a novel speech input. The subject is uttering the quote “live long and prosper”. Cross validation technique was used where first half of the video is utilized for learning system parameters, and the speech part of the second half of the video is used for synthesis and validation. Row one is the speech signal, sampled at 44100 Hz. Row two is the system output, synthetic feature point positions. Row three is the full textured reconstruction of the face by way of identifying the ”closest” feature configuration in the data-set and morphing its corresponding face image into the shape described by synthetic face pose. Row 4 is the actual facial motion associated with the data used for control proposes in cross-validation.

Given the identified model parameters and novel speech segment $u_n(t)$ we evolved the system (3) forward in time to obtain corresponding synthetic facial motion trajectories. This involved feeding $u_n(t)$ to the deterministic component of the system and drawing random samples from the non-Gaussian histogram of q to drive the stochastic component of the system. Note that here $u_n(t)$ corresponds to the same speaker and utterance as in the data, but it is a novel instance of it. For testing purposes we used only half of the data segments for training. The other half was used to extract the speech segment $u_n(t)$.

At the end we used the synthesized facial shapes, $s_n(t)$, to construct facial image sequences. For a given shape $s_n(t)$ we identified the closest (L_2 norm) shape s_i in the training data and morphed its corresponding albedo ρ_i onto $s_n(t)$. Facial image I at pixel x was computed according to $I(x) = \rho_i(\bar{w}(x, s_i, s_n(t)))$ where we have defined the piecewise affine (backward) warp function $\bar{w} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ as $\bar{w}(x, s_i, s_n) = A(x, s_i, s_n)x + b(x, s_i, s_n)$, with $A \in \mathbb{R}^{2 \times 2}$, $b \in \mathbb{R}^2$. \bar{w} maps pixel coordinate x within the triangular grid of s_n to point $\bar{w}(x, s_i, s_n)$ within the corresponding triangular grid of s_i .

The identification of the correct shape s_i from data to match $s_n(t)$ is, of course, highly non-trivial, particularly for systems designed to include the entire span of utterances. Such schemes would require construction of a basic alphabet for the space of utterances in the image space; visemes and other variants have been devised for this purpose and there are existing techniques for identifying viseme sequences corresponding to arbitrary speech waveforms. But in our case this is sufficient for demonstrating the efficacy of the modeling process which is mainly on the geometry of the face.

Motivated by the ultimate prospect of a real-time system, we relied on graphics texture mapping to achieve morphing of the matched albedos onto shapes of synthesized faces. That is, by creating a mesh, in this case 2D, using the shape vectors we mapped the matched albedos in the data onto novel facial shapes. The technique is computationally efficient and benefits from graphics hardware for texture mapping. The resulting dynamics was faithful to original utterances and reconstructed images exhibit no blurring artifacts⁵.

6 Conclusions

We presented a method for modeling facial motion induced by speech. We used a representation for the face where geometric and photometric elements are decoupled. We modeled the dynamics of the geometric (shape) component using a linear dynamical system made up of two parts, a deterministic component driven by the speech waveform and a stochastic part driven by non-Gaussian noise. In our initial stage of development we show examples of the model at work using a set of various utterances, including digits and famous quotes. With this small set, we showed experimental results demonstrating the efficacy of our model in capturing the complexities of time dependent and multi-modal data.

⁵ Sample movies of synthesized sequences can be downloaded from
[http://www.ee.ucla.edu/\\$sim\\$saian/Face.html](http://www.ee.ucla.edu/simsaian/Face.html)

Acknowledgments. This work is supported by AFOSR F49620-03-1-0095, NSF ECS-0200511, CCR-0121778, and ONR N00014-03-1-0850.

References

1. V. Blanz, T. Vetter. A morphable model for synthesis of 3d faces. *Proceedings of ACM SIGGRAPH*, 187–194, 1999.
2. M. Brand. Voice Puppetry *Proceedings of ACM SIGGRAPH 1999*, 21–28, 1999.
3. C. Bregler, M. Covell, and M. Slaney Video Rewrite: Driving Visual Speech with Audio *Proceedings of ACM SIGGRAPH*, 353–360, 1997.
4. A. Chiuso and G. Picci. Subspace identification by orthogonal decomposition. In *Proc. 14th IFAC World Congress*, volume I, pages 241–246, 1999.
5. A. Chiuso and G. Picci. Subspace identification by data orthogonalization and model decoupling. *submitted to Automatica*, 2003.
6. A. Chiuso and G. Picci. Asymptotic variance of subspace methods by data orthogonalization and model decoupling. In *Proc. of the IFAC Int. Symposium on System Identification (SYSID)*, Rotterdam, August 2003.
7. E. Chuang, C. Bregler Facial expression space learning. *To appear in Pacifica Graphics*, 2002.
8. P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.
9. T. F. Cootes, G. J. Edwards and C. J. Taylor, Active Appearance Models. In *Proc. 5th European Conference on Computer Vision*, Freiburg, Germany, 1998
10. T. Ezzat., G. Geiger, T. Poggio. Trainable Videorealistic Speech Animation *Proceedings of ACM SIGGRAPH 2002*, 388–398, 2002.
11. B. Giannakis. and J. Mendel. Identification of nonminimum phase systems using higher order statistics. *IEEE Trans. Acoustic Speech and Signal Processing*, 37(3):360–377, 1989.
12. C. W. J. Granger Economic processes involving feedback In *Information and Control*, 6, 1963, pp.28-48/
13. U. Grenander Elements of Pattern Thoery. The Johns Hopkins University Press, 1996
14. A. Hyvärinen. Independent component analysis for time-dependent stochastic processes. 1998.
15. H. Jin, P. Favaro and S. Soatto. Real-time Feature Tracking and Outlier Rejection with Changes in Illumination. In *Proc. of the Intl. Conf. on Computer Vision*, July 2001
16. L. Ljung System indentification: theory for the user *Prentice-Hall, Inc*, ISBN 0-138-81640-9, 1986.
17. I. Matthews and S. Baker Active Appearance Models Revisited. In *International Journal of Computer Vision*, 2004.
18. G. Picci and T. Katayama. Stochastic realization with exogenous inputs and “subspace methods” identification. *Signal Processing*, 52:145–160, 1996.
19. P. Saisan and A. Bissacco Image-based modeling of human gaits with higher-order statistics. In *Proc. of the Intl. Workshop on Dynamic Scene Analysis*, Copenhagen, June 2002.
20. J. Shi and C. Tomasi Good Features to Track *CVPR*, 1994.
21. P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. *Automatica*, 29:649–660, 1993.
22. L. Zhang. and A. Cichocki Blind deconvolution of Dynamical Systems : A State-Space Approach. *Proceedings of the IEEE. Workshop on NNSP'98*, 123–131, 1998.

Recovering Local Shape of a Mirror Surface from Reflection of a Regular Grid

Silvio Savarese¹, Min Chen², and Pietro Perona¹

¹ California Institute of Technology, Mail stop 136-93, Pasadena, CA 91125, USA
`{savarese,perona}@vision.caltech.edu,`
`http://www.vision.caltech.edu/`

² Oracle Parkway, Redwood City, CA 94065, USA
`{min.chen}@oracle.com`

Abstract. We present a new technique to recover the shape of an unknown smooth specular surface from a single image. A calibrated camera faces a specular surface reflecting a calibrated scene (for instance a checkerboard or grid pattern). The mapping from the scene pattern to its reflected distorted image in the camera changes the local geometrical structure of the scene pattern. We show that if measurements of both local orientation and scale of the distorted scene in the image plane are available, this mapping can be inverted. Specifically, we prove that surface position and shape up to third order can be derived as a function of such local measurements when two orientations are available at the same point (e.g. a corner). Our results generalize previous work [1, 2] where the mirror surface geometry was recovered only up to first order from at least three intersecting lines. We validate our theoretical results with both numerical simulations and experiments with real surfaces.

1 Introduction and Motivation

Shiny surfaces have traditionally been considered a nuisance in computer vision. Many objects of interest and man-made surfaces are smooth and shiny, such as a metal spoon or a clean automobile, and violate the hypothesis of traditional shape reconstruction techniques (e.g. shape from shading, texture gradient, etc...). In fact, it is not possible to observe the intrinsic surface of a mirror but only what is reflecting. This additional cue, however, may be precisely exploited in order to infer the shape of this category of objects.

In this paper we present a new technique to recover local shape of an unknown smooth specular surface by observing the deformation of the reflection of a regular pattern in a calibrated scene (for instance, a checkerboard grid), using a calibrated camera (see Fig. 1). Our approach extends and generalizes our previous work [1,2,3] based on a novel observation that the mapping from the scene grid to the reflected curved grid in the camera image plane due to mirror reflection not only changes the “orientation” of the grid lines but also “stretches” the grid step, modifying the local scale of the pattern. Such a deforming mapping can be easily illustrated by the grid points (1, 2, 3, 4, 5) and

their corresponding points ($1', 2', 3', 4', 5'$) in the curved grid reflected on the mirror surface shown in Fig 1. We first analyze this map and derive analytical expressions for the local geometry in the image (namely, first- and second-order derivatives at the intersection points of the curved grid) as a function of mirror surface position and shape. We then study the inverse problem and derive surface position and shape up to third order as a function of local position, orientation and local scale measurements in the image when two orientations are available at the same point. Such local measurements may be computed at a point (e.g. $1'$ in Fig. 1(c)) from its four neighboring reflected points (e.g. $2', 3', 4', 5'$). By comparing these measurements with their corresponding analytical expressions, we induce a set of constraints, which lead to solutions for surface position, as well as closed-form solutions for normal, curvature and third-order local parameters of the surface around the reflected point of interest. As a result, our reconstruction approach is only "technically" sparse as we can estimate local shape (i.e. surface orientation, curvature and third order parameters) in the neighborhood of each reflected point. In other words, we obtain a "piece-wise parabolic" reconstruction, where each "piece" is a vertex of a paraboloid. A robust estimation of the surface's shape may be ultimately obtained by integrating such information.

1.1 Previous Work and Paper Organization

Pioneering work on specular surfaces reconstruction was carried out by Koenderink [12], Blake [6,5] and Zisserman [15] who tackled the problem under the hypothesis of viewer motion. Other approaches include those based on mathematical models of specular reflections [10,11], analyzing 3D surface profiles travelled by virtual features (Oren and Nayar [13]), as well as their extensions [14]. Halsead et al. [9] proposed a reconstruction algorithm where a surface global model is fitted to a set of normals obtained by imaging a pattern of light reflected by specular surface. Their results were applied to interactive visualization of the cornea. Perard [16] and Tarini et al. [17] proposed a structured lighting technique for the iterative reconstruction of surface normal vectors and topography. Bonfort et al. [18] presented a voxel-based approach in the spirit of multiple view (space carving) algorithms. Among these techniques, some limitations are the necessity of having available a certain degree of knowledge on shape and position of the object; multiple images under different condition of the illuminant; dedicated hardware equipment. Savarese et al. [1,2,3] tried to overcome the above limitations and tackled the monocular single-image case using a local and differential approach: local measurements of position and orientation of three intersecting lines were used to recover first-order surface geometry (and second-order up to one free parameter). By exploiting measurements of both local scale and orientation, our work generalizes and extends this result to be able to recover the local surface geometry up to third-order accuracy using fewer lines. Table 1 summarizes the difference between our results and previous work [2,3].

The rest of the paper is organized as follows. After introducing the problem formulation in Section 2, we present in Section 3 full analytical expressions for

Table 1. Comparison of our results with previous work

Method	Measurements	Estim. surface quantities
Our Method	point \mathbf{q} + orientation & scale of 2 lines through \mathbf{q}	distance, tangent plane, curvature and 3 rd order param. at \mathbf{r}
[2,3]	point \mathbf{q} + orientation of 3 lines through \mathbf{q}	distance, tangent plane at \mathbf{r}

the first- and second-order derivatives of a reflected image curve, and then derive closed-form solutions for the unknown surface parameters in Section 4. In Section 5, we describe how to measure derivatives of a reflected image curve using numerical approximation and address its associated error issues. We finally validate our theoretical results with both numerical simulations and experiments with real surfaces in Section 6.

2 Setup and Problem Formulation

The geometric setup is depicted in Fig. 1(a). A calibrated scene composed of a pattern of intersecting lines is reflected off an unknown smooth mirror surface and the reflection is observed by a calibrated camera. Our goal is to obtain local geometrical information of the surface by analyzing the deformation produced upon the pattern of lines.

Let \mathbf{c} be the center of projection of the camera. The image plane is positioned l unit distance in front of \mathbf{c} , perpendicular to the view direction \mathbf{v} . Given a scene point \mathbf{p} , let \mathbf{q} be the image of \mathbf{p} observed on the image plane through a specular reflection on the mirror surface at \mathbf{r} . See Fig. 2(a). Then \mathbf{q} and \mathbf{r} are constrained by the following relationship:

$$\mathbf{r} = \mathbf{c} + s \mathbf{d}, \quad (1)$$

where the unit vector $\mathbf{d} = (\mathbf{q} - \mathbf{c}) / \| \mathbf{q} - \mathbf{c} \|$ is the view ray direction, and $s = \| \mathbf{r} - \mathbf{c} \|$ is the distance from \mathbf{r} to \mathbf{c} . With \mathbf{c} fixed and \mathbf{q} measured, the surface position at \mathbf{r} is completely determined by a single distance parameter s .

To recover the higher-order surface parameters around the reflection point \mathbf{r} besides its position, we introduce a suitable coordinate reference system $[\mathbf{U} \mathbf{V} \mathbf{W}]$ centered at \mathbf{r} and refer to it as the *principal reference system*, similar to that adopted in [5,2]. Let \mathbf{n}_p be the normal vector to the plane defined by \mathbf{q} , \mathbf{p} and \mathbf{c} , and let \mathbf{n}_r be the surface normal at \mathbf{r} . Then, $\mathbf{W} = \mathbf{n}_r$, $\mathbf{V} = \mathbf{n}_p$, and $\mathbf{U} = \mathbf{V} \times \mathbf{W}$. Given an arbitrary point \mathbf{x} represented in a reference system $[\mathbf{X} \mathbf{Y} \mathbf{Z}]$ centered in \mathbf{c} , its corresponding coordinates \mathbf{x}' in $[\mathbf{U} \mathbf{V} \mathbf{W}]$ can be obtained by a transformation $\mathbf{x}' = \mathbf{R}^T(\mathbf{x} - \mathbf{r})$, where $\mathbf{R} = [\mathbf{n}_p \times \mathbf{n}_r \mathbf{n}_p \mathbf{n}_r]$, which is a function of the unknown parameter s . In the principal reference system, the normal of the surface at the origin is \mathbf{W} and the tangent plane to the surface is the plane defined by \mathbf{U} and \mathbf{V} , thus the surface around \mathbf{r} can be written in the *special Monge form* [8], yielding

$$w = \frac{1}{2!}(a u^2 + 2c uv + b v^2) + \frac{1}{3!}(e u^3 + 3f u^2 v + 3g u v^2 + h v^3) + \dots, \quad (2)$$

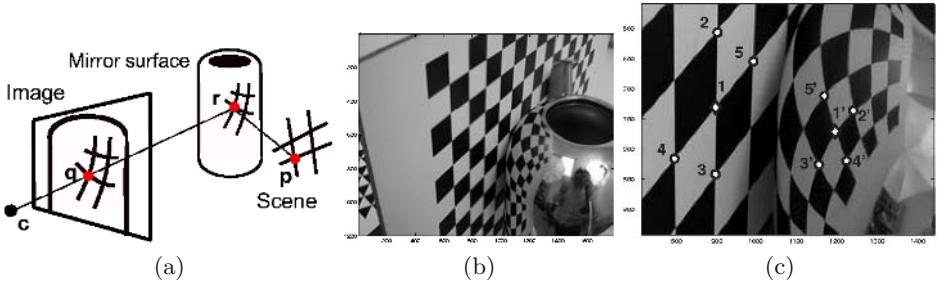


Fig. 1. Setup. (a) A camera is facing a specular surface reflecting a scene. (b) Image seen from the camera. (c) Points correspondence $(1, 2, 3, 4, 5)$ and $(1', 2', 3', 4', 5')$ under reflection.

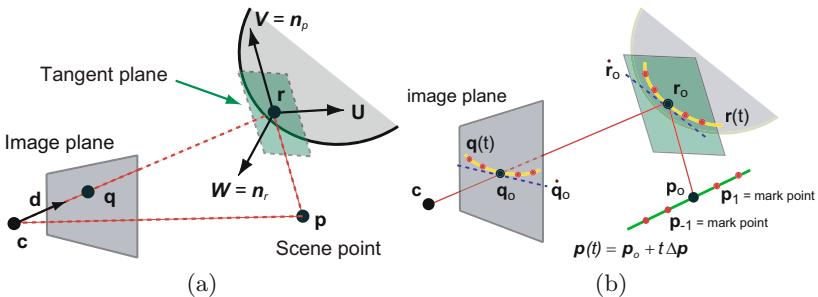


Fig. 2. Specular reflection geometry

where we call a, b, c and e, f, g, h the second-order and third-order surface parameters around \mathbf{r} , respectively. Accordingly, we refer to s as the first-order surface parameter which determines both position and normal of the surface. From now on we assume that we work in the local principal reference system.

2.1 Surface Recovery

Our focus in this paper is to recover first-, second- and third-order surface parameters around \mathbf{r} using quantities that are known or measurable. Note that \mathbf{c} , \mathbf{q} , \mathbf{p} are known by assuming calibrated camera and scene.

Consider two lines intersecting at a point \mathbf{p} . Through specular reflection, these two lines becomes two curves on the mirror surface intersecting at \mathbf{r} and subsequently are observed as two deformed curves on the image plane, intersecting at \mathbf{q} (the image of \mathbf{p}). Our approach is to perform differential analysis around \mathbf{q} . Specifically, we derive analytical expressions for the first- and second-order derivatives of the two deformed image curves at \mathbf{q} , in terms of surface parameters up to the third order (s, a, b, c, e, f, g, h) (see Section 3). By comparing these analytical formulas with their corresponding local measurements in the image obtained from scale and orientation of the lines at \mathbf{q} (see Section 5),

we impose a set of constraints on the unknown surface parameters. The resulting constraint system leads to solution for s and closed-form solutions of the remaining unknowns, allowing us to recover the mirror surface locally around the reflection point \mathbf{r} up to the third-order accuracy (see Section 4). Our results are summarized in Table 1.

3 Differential Analysis

A line passing through a scene point $\mathbf{p}_o (= [\mathbf{p}_{ou} \mathbf{p}_{ov} \mathbf{p}_{ow}]^T)$ in 3D space can be described in a parametric form by $\mathbf{p}(t) = \mathbf{p}_o + t \Delta\mathbf{p}$, where t is a parameter and $\Delta\mathbf{p} = [\Delta p_u \Delta p_v \Delta p_w]^T$ is the orientation vector of the line. Given a fixed camera position \mathbf{c} , a mapping from the parameter t to the corresponding reflection point \mathbf{r} in the mirror surface defines a parameterized space curve $\mathbf{r}(t)$ lying on the mirror surface, which describes the position of the reflection point as t varies. See Fig. 2(b). Consequently, through a perspective projection, $\mathbf{r}(t)$ is mapped to another parameterized curve $\mathbf{q}(t)$ on the image plane. In order to perform differential analysis, we denote the first-order derivatives (tangent vector) of $\mathbf{r}(t)$ and $\mathbf{q}(t)$ respectively by $\dot{\mathbf{r}}$ and $\dot{\mathbf{q}}$, and denote their second-order derivatives respectively by $\ddot{\mathbf{r}}$ and $\ddot{\mathbf{q}}$. They are all functions of t . When $t = t_o = 0$, we denote $\mathbf{r}(t_o)$ by \mathbf{r}_o , which is the reflection point of \mathbf{p}_o on the mirror surface and can be set as the origin of the principal reference system. Accordingly, the values of $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$ and $\ddot{\mathbf{q}}$ evaluated at t_o are denoted by $\dot{\mathbf{r}}_o$, $\dot{\mathbf{q}}_o$, $\ddot{\mathbf{r}}_o$ and $\ddot{\mathbf{q}}_o$. Throughout this paper, if there is no further explanation, we always assume that we evaluate $\dot{\mathbf{r}}$, $\ddot{\mathbf{r}}$ and $\ddot{\mathbf{q}}$ at t_o , and omit the subscript o to make the notation easier on the eye.

3.1 First-Order Derivative of $\mathbf{r}(t)$

By formulating a specular reflection path from \mathbf{p} to \mathbf{q} as Chen and Arvo [7] did and carrying out implicit differentiation [2], we can compute $\dot{\mathbf{r}} (= [\dot{u} \dot{v} \dot{w}]^T)$ analytically as

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = -\frac{1}{\Delta} \begin{bmatrix} J_v - 2b \cos \theta & 2c \cos \theta & 0 \\ 2c \cos \theta & J_u - 2a \cos \theta & 0 \\ 0 & 0 & \Delta \end{bmatrix} \begin{bmatrix} B_u \\ B_v \\ 0 \end{bmatrix} \quad (3)$$

where θ is the reflection angle at \mathbf{r}_o , and

$$\begin{aligned} \Delta &= (J_u - 2a \cos \theta)(J_v - 2b \cos \theta) - 4c^2 \cos^2 \theta \\ B_v &= -\frac{\Delta p_v}{\|\mathbf{p}_o\|}; \quad B_u = \frac{\Delta p_w \cos \theta \sin \theta - \Delta p_u \cos^2 \theta}{\|\mathbf{p}_o\|} \\ J_u &= J_v \cos^2 \theta; \quad J_v = \frac{s + \|\mathbf{p}_o\|}{s \|\mathbf{p}_o\|} \quad \cos \theta = \frac{\sqrt{2}}{2} \sqrt{\frac{s - \langle \mathbf{d}, \mathbf{p}_o \rangle}{\|\mathbf{s}\mathbf{d} - \mathbf{p}_o\|} + 1}. \end{aligned}$$

Equation (3) holds when $\Delta \neq 0$, which is true in general [2]. With \mathbf{p}_o , $\Delta\mathbf{p}$ and \mathbf{c} fixed for a given scene line, Eq. (3) expresses $\dot{\mathbf{r}}$ as a function of unknown surface

parameters s, a, b, c . A similar equation was derived by Zisserman *et al.* [15] in the dual context of a moving observer.

In [2] we found that if (at least) 2 scene lines $\mathbf{p}_i, \mathbf{p}_j$ intersecting at \mathbf{p}_o are available, a, b, c can be expressed as a function of s and a free parameter r :

$$\begin{cases} a = \frac{J_u}{2 \cos \theta} - r \frac{h_1}{2 \cos \theta} \\ b = \frac{J_v}{2 \cos \theta} - r \frac{h_2}{2 \cos \theta} \\ c = r \frac{h_3}{2 \cos \theta} \end{cases}, \quad (4)$$

where $[h_1 \ h_2 \ h_3]^T = \mathbf{h}_k \times \mathbf{h}_j$, $\mathbf{h}_i = [B_{v_i}, -B_{u_i} \tan \phi_i, B_{u_i} - B_{v_i} \tan \phi_i]$ and ϕ_i denotes the angle between $\dot{\mathbf{r}}_i$ and the \mathbf{U} axis at \mathbf{r}_o .

3.2 Second-Order Derivative of $\mathbf{r}(t)$

Let $f(u, v, w) = 0$ denote the implicit function of the mirror surface \mathbf{s} represented in Eq. (2) and $\mathbf{H}_s = \partial^2 f / \partial \mathbf{r}^2$ be the Hessian matrix of the function f with respect to the reflection point \mathbf{r} . Under the assumption that the third-order terms in the Monge form (2) are negligible, in [3] we exploited the property that $\partial \mathbf{H}_s / \partial \mathbf{r} = 0$ at $t = t_o$, and derived a simplified second-order derivative $\ddot{\mathbf{r}}$ of the curve $\mathbf{r}(t)$. We generalize here this result to the case where e, f, g, h are not negligible and obtain a general expression for the second-order derivative $\ddot{\mathbf{r}}$:

$$\ddot{\mathbf{r}} = \begin{bmatrix} \ddot{u} \\ \ddot{v} \\ \ddot{w} \end{bmatrix} = \begin{bmatrix} \ddot{u}_1 \\ \ddot{v}_1 \\ \ddot{w}_1 \end{bmatrix} + \begin{bmatrix} \ddot{u}_2 \\ \ddot{v}_2 \\ 0 \end{bmatrix} = \ddot{\mathbf{r}}_1 + \ddot{\mathbf{r}}_2. \quad (5)$$

The first term $\ddot{\mathbf{r}}_1$ is given by $\ddot{w}_1 = -a\ddot{u}^2 - 2c\ddot{u}\dot{v} - b\dot{v}^2$ and

$$\begin{bmatrix} \ddot{u}_1 \\ \ddot{v}_1 \end{bmatrix} = -\frac{1}{\Delta} \begin{bmatrix} J_v - 2b \cos \theta & 2c \cos \theta \\ 2c \cos \theta & J_u - 2a \cos \theta \end{bmatrix} \begin{bmatrix} D_1 - J_w \ddot{w} \\ D_2 \end{bmatrix}, \quad (6)$$

where $J_w = ((\|\mathbf{p}_o\| - s) \sin \theta \cos \theta) / (s \|\mathbf{p}_o\|)$, and D_1, D_2 are functions depending only on s, a, b, c . The second term $\ddot{\mathbf{r}}_2$ depends on the third-order surface parameters and is expressed as:

$$\begin{bmatrix} \ddot{u}_2 \\ \ddot{v}_2 \end{bmatrix} = \frac{2 \cos \theta}{\Delta} \begin{bmatrix} J_v - 2b \cos \theta & 2c \cos \theta \\ 2c \cos \theta & J_u - 2a \cos \theta \end{bmatrix} \begin{bmatrix} \dot{u}^2 & 2\dot{u}\dot{v} & \dot{v}^2 & 0 \\ 0 & \dot{u}^2 & 2\dot{u}\dot{v} & \dot{v}^2 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix}. \quad (7)$$

Detailed derivation of Eq. (6) and Eq. (7) is available as a technical report.

3.3 Relationship between $\dot{\mathbf{r}}$ and $\dot{\mathbf{q}}$

Based on Eq. (3), we derive an analytical formula for the first-order derivative of $\mathbf{q}(t)$ on the image plane by examining the relationship between $\dot{\mathbf{r}}$ and $\dot{\mathbf{q}}$.

Letting $\lambda(t) = \|\mathbf{q}(t) - \mathbf{c}\| / \|\mathbf{r}(t) - \mathbf{c}\|$ be the ratio between the distance from \mathbf{c} to $\mathbf{q}(t)$ and that from \mathbf{c} to $\mathbf{r}(t)$. We may express the image plane curve $\mathbf{q}(t)$ as follows:

$$\mathbf{q}(t) - \mathbf{c} = \lambda(t)(\mathbf{r}(t) - \mathbf{c}). \quad (8)$$

In our setup the image plane is located l unit distance along the view direction \mathbf{v} , thus λ satisfies the following equation

$$\lambda(t) \langle \mathbf{r}(t) - \mathbf{c}, \mathbf{v} \rangle = l. \quad (9)$$

Here, $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. Using Eq. (9), we may evaluate λ and $\dot{\lambda}$ at t_o as

$$\lambda = \frac{l}{s \langle \mathbf{d}, \mathbf{v} \rangle}, \quad \dot{\lambda} = -\frac{l \langle \dot{\mathbf{r}}, \mathbf{v} \rangle}{s^2 \langle \mathbf{d}, \mathbf{v} \rangle^2}. \quad (10)$$

Then we can differentiate Eq. (8) with respect to t and compute $\dot{\mathbf{q}}$ as follows:

$$\dot{\mathbf{q}} = \lambda \dot{\mathbf{r}} + s \dot{\lambda} \mathbf{d} = \frac{l}{s \langle \mathbf{d}, \mathbf{v} \rangle} \left[\mathbf{I} - \frac{\mathbf{d} \mathbf{v}^T}{\langle \mathbf{d}, \mathbf{v} \rangle} \right] \dot{\mathbf{r}} = \mathbf{T} \dot{\mathbf{r}}, \quad (11)$$

where the 3×3 matrix \mathbf{T} is defined as

$$\mathbf{T} = \frac{l}{s \langle \mathbf{d}, \mathbf{v} \rangle} \left[\mathbf{I} - \frac{\mathbf{d} \mathbf{v}^T}{\langle \mathbf{d}, \mathbf{v} \rangle} \right]. \quad (12)$$

3.4 Relationship between $\ddot{\mathbf{q}}$ and $\ddot{\mathbf{r}}$, $\dot{\mathbf{r}}$

To relate the second-order derivative of $\mathbf{r}(t)$ to that of its image plane projection $\mathbf{q}(t)$, we differentiate Eq. (11) with respect to t , obtaining

$$\lambda \ddot{\mathbf{q}} = \lambda^2 \ddot{\mathbf{r}} + (\ddot{\lambda} \lambda - 2\dot{\lambda}^2) s \mathbf{d} + 2\dot{\lambda} \dot{\mathbf{q}}, \quad (13)$$

where λ and $\dot{\lambda}$ are defined in Eq. (10). By differentiating Eq. (9) twice with respect to t , we can compute $\ddot{\lambda}$ and then get an analytical formula for $\ddot{\mathbf{q}}$ from Eq. (13), that is

$$\ddot{\mathbf{q}} = \mathbf{T} \left[\ddot{\mathbf{r}} - \frac{2}{s \langle \mathbf{d}, \mathbf{v} \rangle} \langle \dot{\mathbf{r}}, \mathbf{v} \rangle \dot{\mathbf{r}} \right], \quad (14)$$

where the matrix \mathbf{T} is defined as in Eq. (12). It then follows from Eq. (5) that $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_1 + \ddot{\mathbf{q}}_2$, where

$$\ddot{\mathbf{q}}_1 = \mathbf{T} \left[\ddot{\mathbf{r}}_1 - \frac{2}{s \langle \mathbf{d}, \mathbf{v} \rangle} \langle \dot{\mathbf{r}}, \mathbf{v} \rangle \dot{\mathbf{r}} \right], \quad \ddot{\mathbf{q}}_2 = \mathbf{T} \ddot{\mathbf{r}}_2. \quad (15)$$

As we can see, the third-order surface parameters e, f, g, h only appear in $\ddot{\mathbf{q}}_2$.

4 Surface Reconstruction

In this section we shall show that by using *two* scene lines intersecting at a point \mathbf{p}_o , we are able to recover first order surface parameter s and higher order parameters a, b, c, e, f, g, h in close form using Eq. (11) and Eq. (15). For clarity, we indicate the quantities measured in the image plane with the superscript m and the quantities associated with different scene lines with a subscript. For example, the measurement of the first-order derivative of the i th curve $\mathbf{q}_i(t)$ in the image plane is indicated by $\dot{\mathbf{q}}_i^m$.

4.1 Recovering First and Second Order Parameters

It can be seen from Eq. (4) that we only need to determine two unknowns r and s to recover first and second-order surface parameters. Replacing a, b, c in Eq. (3) by Eq. (4), we obtain a novel expression for $\dot{\mathbf{r}}$ in terms of r and s (embedded in h_1, h_2, h_3, B_u, B_v) as:

$$\dot{\mathbf{r}} = -\frac{1}{r(h_1h_2 - h_3^2)} \begin{bmatrix} h_2 & h_3 & 0 \\ h_3 & h_1 & 0 \\ 0 & 0 & r(h_1h_2 - h_3^2) \end{bmatrix} \begin{bmatrix} B_u \\ B_v \\ 0 \end{bmatrix} = -\frac{1}{r} \begin{bmatrix} \mathbf{V} \mathbf{B} \\ 0 \end{bmatrix},$$

where we have defined

$$\mathbf{V} = \frac{1}{h_1h_2 - h_3^2} \begin{bmatrix} h_2 & h_3 \\ h_3 & h_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} B_u \\ B_v \end{bmatrix}.$$

Accordingly, it follows from Eq. (11) that the first-order derivative $\dot{\mathbf{q}}$ and its L_2 norm can also be expressed in terms of the two unknowns r and s :

$$\dot{\mathbf{q}} = -\frac{1}{r} \mathbf{T} \begin{bmatrix} \mathbf{V} \mathbf{B} \\ 0 \end{bmatrix}, \quad \|\dot{\mathbf{q}}\|^2 = \langle \dot{\mathbf{q}}, \dot{\mathbf{q}} \rangle = \frac{1}{r^2} [\mathbf{B}^T \mathbf{V}^T 0] \mathbf{T}^T \mathbf{T} \begin{bmatrix} \mathbf{V} \mathbf{B} \\ 0 \end{bmatrix} \quad (16)$$

where only the unknown s (not r) appears in \mathbf{T} , \mathbf{V} and \mathbf{B} .

Suppose that we are able to measure tangent directions ($\tan \phi_k$ and $\tan \phi_j$) and first-order derivatives ($\dot{\mathbf{q}}_k^m$ and $\dot{\mathbf{q}}_j^m$) of $\mathbf{q}_k(t)$ and $\mathbf{q}_j(t)$ at \mathbf{q}_o , respectively (see Section 5). By taking the ratio $\|\dot{\mathbf{q}}_k^m\|^2 / \|\dot{\mathbf{q}}_j^m\|^2$, we have

$$\frac{\|\dot{\mathbf{q}}_k^m\|^2}{\|\dot{\mathbf{q}}_j^m\|^2} = \frac{[\mathbf{B}_k^T \mathbf{V}^T 0] \mathbf{T}^T \mathbf{T} \begin{bmatrix} \mathbf{V} \mathbf{B}_k \\ 0 \end{bmatrix}}{[\mathbf{B}_j^T \mathbf{V}^T 0] \mathbf{T}^T \mathbf{T} \begin{bmatrix} \mathbf{V} \mathbf{B}_j \\ 0 \end{bmatrix}}, \quad (17)$$

where the matrix \mathbf{V} is expressed in terms of our tangent direction measurements $\tan \phi_k$ and $\tan \phi_j$. Notice that the matrix \mathbf{T} defined in Eq. (12) does not depend on a particular line. Equation (17) imposes a constraint for us to solve for s . Once s is computed, we can easily derive the closed-form solution for another unknown r up to a sign from Eq. (16):

$$r^2 = \frac{[\mathbf{B}_k^T \mathbf{V}^T 0] \mathbf{T}^T \mathbf{T} \begin{bmatrix} \mathbf{V} \mathbf{B}_k \\ 0 \end{bmatrix}}{\|\dot{\mathbf{q}}_k^m\|^2}. \quad (18)$$

4.2 Recovering Third Order Parameters

To recover the third-order surface parameters, we assume that we are able to estimate the second-order derivatives for the two reflection curves in the image plane, denoted by $\hat{\mathbf{q}}_k^m$ and $\hat{\mathbf{q}}_j^m$ respectively (see Section 5). Let $\hat{\mathbf{v}}$ denote a 2D vector consisting of the first two components of a 3D vector \mathbf{v} . In accordance with the decomposition of $\hat{\mathbf{q}}$ in Eq. (15), we can divide $\hat{\mathbf{q}}_k^m$ and $\hat{\mathbf{q}}_j^m$ into two parts, yielding:

$$(\hat{\mathbf{q}}_2)_k^m = \hat{\mathbf{q}}_k^m - (\hat{\mathbf{q}}_1)_k, \quad (\hat{\mathbf{q}}_2)_j^m = \hat{\mathbf{q}}_j^m - (\hat{\mathbf{q}}_1)_j, \quad (19)$$

where $(\hat{\mathbf{q}}_1)_k$ and $(\hat{\mathbf{q}}_1)_j$ (independent of e, f, g, h) are known from Eq. (6) once we have recovered s, a, b, c . On the other hand, we have analytical solutions for $(\hat{\mathbf{q}}_2)_k$ and $(\hat{\mathbf{q}}_2)_j$ from Eq. (15), that is,

$$(\hat{\mathbf{q}}_2)_k = \mathbf{T}_{22}(\hat{\mathbf{r}}_2)_k, \quad (\hat{\mathbf{q}}_2)_j = \mathbf{T}_{22}(\hat{\mathbf{r}}_2)_j, \quad (20)$$

where \mathbf{T}_{22} is the upper left 2×2 sub-matrix of \mathbf{T} , and $\hat{\mathbf{r}}_2$ can be expressed from Eq. (7) using Eq. (4) as

$$\begin{bmatrix} \ddot{u}_2 \\ \ddot{v}_2 \end{bmatrix} = \frac{2 \cos \theta}{r(h_1 h_2 - h_3^2)} \begin{bmatrix} h_2 & h_3 \\ h_3 & h_1 \end{bmatrix} \begin{bmatrix} \dot{u}^2 & 2\dot{u}\dot{v} & \dot{v}^2 & 0 \\ 0 & \dot{u}^2 & 2\dot{u}\dot{v} & \dot{v}^2 \end{bmatrix} \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix}. \quad (21)$$

Thus, Equating Eq. (19) and Eq. (20) gives rise to a constraint system for e, f, g, h :

$$\begin{bmatrix} (\hat{\mathbf{q}}_k^m - (\hat{\mathbf{q}}_1)_k) \\ (\hat{\mathbf{q}}_j^m - (\hat{\mathbf{q}}_1)_j) \end{bmatrix} = \frac{2 \cos \theta}{r(h_1 h_2 - h_3^2)} \mathbf{M}_1 \mathbf{M}_2 \mathbf{M}_3 \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix}, \quad (22)$$

where $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$ are defined as follows:

$$\mathbf{M}_1 = \begin{bmatrix} h_2 & h_3 & 0 & 0 \\ h_3 & h_1 & 0 & 0 \\ 0 & 0 & h_2 & h_3 \\ 0 & 0 & h_3 & h_1 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} \mathbf{T}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{22} \end{bmatrix}, \quad \mathbf{M}_3 = \begin{bmatrix} \dot{u}_k^2 & 2\dot{u}_k\dot{v}_k & \dot{v}_k^2 & 0 \\ 0 & \dot{u}_k^2 & 2\dot{u}_k\dot{v}_k & \dot{v}_k^2 \\ \dot{u}_j^2 & 2\dot{u}_j\dot{v}_j & \dot{v}_j^2 & 0 \\ 0 & \dot{u}_j^2 & 2\dot{u}_j\dot{v}_j & \dot{v}_j^2 \end{bmatrix}. \quad (23)$$

Equation (22) leads to the following closed-form solution for the third-order surface parameters, that is,

$$\begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix} = \frac{r(h_1 h_2 - h_3^2)}{2 \cos \theta} (\mathbf{M}_1 \mathbf{M}_2 \mathbf{M}_3)^{-1} \begin{bmatrix} (\hat{\mathbf{q}}_k^m - (\hat{\mathbf{q}}_1)_k) \\ (\hat{\mathbf{q}}_j^m - (\hat{\mathbf{q}}_1)_j) \end{bmatrix}, \quad (24)$$

where the existence of $(\mathbf{M}_1 \mathbf{M}_2 \mathbf{M}_3)^{-1}$ is based on the following proposition:

Proposition 1 The matrix $\mathbf{M}_1 \mathbf{M}_2 \mathbf{M}_3$ is invertible.

Proof: $\det(\mathbf{M}_1) \neq 0$ follows directly from $\Delta \neq 0$. It can be easily proved that $\det(\mathbf{M}_2) \neq 0$ since \mathbf{T} in Eq. (11) is associated with the projective transformation from a mirror surface (3 D.O.F.) into the image plane (2 D.O.F.). Let us prove that $\det(\mathbf{M}_3) \neq 0$. We first show that \mathbf{M}_3 is invertible when one of $\{\dot{u}_k, \dot{v}_k, \dot{u}_j, \dot{v}_j\}$ is zero. For example, if $\dot{u}_k = 0$, then $\det(\mathbf{M}_3) = (\dot{v}_k \dot{u}_j)^4 \neq 0$. Otherwise, either $\dot{v}_k = 0$ or $\dot{u}_j = 0$ will contradict to our observation of two curves with different orientations. Next we shall consider the case where none of $\dot{u}_k, \dot{v}_k, \dot{u}_j, \dot{v}_j$ is zero. The proof is performed by contradiction. With two differently-oriented image curves observed, we should have

$$\dot{v}_k / \dot{u}_k \neq \dot{v}_j / \dot{u}_j, \quad \dot{v}_k / \dot{u}_k \neq -\dot{v}_j / \dot{u}_j. \quad (25)$$

Suppose that \mathbf{M}_3 is singular. Its 4 row vectors U_1, U_2, U_3, U_4 are linearly dependent. Without loss of generality, we may assume that $U_4 = k_1 U_1 + k_2 U_2 + k_3 U_3$ (at least one k_i is nonzero), which can be expanded as

$$k_1 \dot{u}_k^2 + k_3 \dot{u}_j^2 = 0 \quad (26)$$

$$2k_1 \dot{u}_k \dot{v}_k + k_2 \dot{u}_k^2 + 2k_3 \dot{u}_j \dot{v}_j = \dot{u}_j^2 \quad (27)$$

$$k_1 \dot{v}_k^2 + 2k_2 \dot{u}_k \dot{v}_k + k_3 \dot{v}_j^2 = 2\dot{u}_j \dot{v}_j \quad (28)$$

$$k_2 \dot{v}_k^2 = \dot{v}_j^2 \quad (29)$$

By eliminating variables through substitutions, we get

$$(k_1 + k_2 k_3)(k_3 \dot{v}_k + \dot{u}_k)^2 = 0, \quad (k_1 + k_2 k_3)(k_1 \dot{v}_j - k_2 \dot{u}_j)^2 = 0. \quad (30)$$

If $k_1 + k_2 k_3 = 0$, then it follows from Eqs. (26) and (29) that $k_3 [\dot{v}_j^2 / \dot{v}_k^2 - \dot{u}_j^2 / \dot{u}_k^2] = 0$. To satisfy Eq. (25), we must have $k_3 = 0$, which leads to $k_1 = 0$ and then $\dot{v}_k / \dot{u}_k = \dot{v}_j / \dot{u}_j$ from Eqs. (27) and (28), contradictory to our assumption (25). Consequently, Equation (30) simplifies to

$$k_3 \dot{v}_k + \dot{u}_k = 0, \quad k_1 \dot{v}_j - k_2 \dot{u}_j = 0. \quad (31)$$

Eqs. (31), (26) and (29) present an over-constrained system for k_1, k_2, k_3 , which requires $\dot{v}_k / \dot{u}_k = \dot{v}_j / \dot{u}_j$, again contradictory to our assumption (25). Therefore, U_1, U_2, U_3, U_4 must be linearly independent, and \mathbf{M}_3 is invertible. $\square \square$

5 Numerical Measurement and Error Analysis

In Section 4 we have assumed that for a reflected curve $\mathbf{q}(t)$ observed on the image plane, we are able to measure its orientation $\tan \phi$, the first-order derivative $\dot{\mathbf{q}}^m$ and the second-order derivative $\ddot{\mathbf{q}}^m$ at \mathbf{q}_o . In this section we shall describe how to numerically compute these quantities, and analyze the reconstruction error due to such approximations.

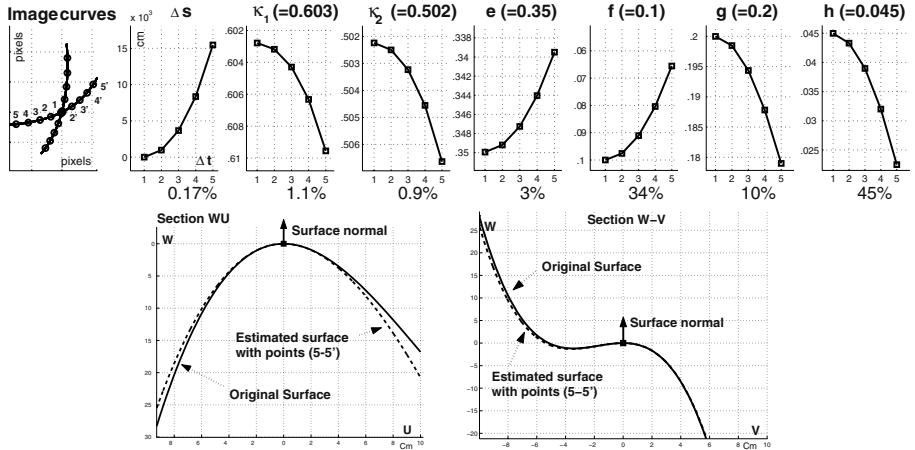


Fig. 3. Reconstruction error analysis by means of numerical simulations on a synthetic mirror surface. Given the center of the camera \mathbf{c} and 2 scene lines intersecting at \mathbf{p}_o , we observe two reflected image curves depicted in the top left panel. The synthetic mirror surface is positioned such that the distance s between the reflecting point \mathbf{r}_o and \mathbf{c} is 9cm. The surface principal curvatures at \mathbf{r}_o are $\kappa_1 = -0.603$ and $\kappa_2 = -0.502$, and the third-order surface parameters are $e = -0.35, f = -0.1, g = 0.2, h = -0.045$. By numerically measuring the first- and second-order derivatives at \mathbf{q}_o (i.e. point 1) using pairs of mark points located at increasing distance Δt from \mathbf{q} (i.e. mark point pair $(2, 2')$, \dots , $(5, 5')$), we recover the local surface at \mathbf{r}_o as described in Section 4. Each of the remaining plots in the top panel recovered surface parameter as a function of the mark gap Δt , with the maximum percentage error reported at the bottom. Notice that the error of recovered distance s increases as a quadratic function of Δt , the curvature error is one order of magnitude bigger than the distance error, and the third-order parameter error is one order of magnitude bigger than the curvature error. In the bottom panels the reconstructed surface (estimated by using the pair of mark points $(5, 5')$) is qualitatively compared to the original one in both $\mathbf{W} - \mathbf{V}$ and $\mathbf{W} - \mathbf{U}$ sections of the \mathbf{UVW} reference system. The numerical approximation (32) and (33) appears to give rise to reasonably good reconstruction results as long as the mark points are close enough to each other (i.e., Δt small enough).

Given a scene line $\mathbf{p}(t)$, we may accurately measure the orientation of its reflected image curve $\mathbf{q}(t)$ at \mathbf{q}_o using B-spline interpolation. In fact, by constructing a B-spline that interpolates image points along the curve, the direction of $\dot{\mathbf{q}}$ (i.e., $\tan \phi$) can be calculated by numerical differentiation of the resulting B-spline. To estimate a complete $\dot{\mathbf{q}}$ (with both direction and magnitude) and higher-order derivative $\ddot{\mathbf{q}}$, we can make use of mark points $\mathbf{p}_o = \mathbf{p}(t_o), \mathbf{p}_{-1} = \mathbf{p}(t_{-1}), \mathbf{p}_1 = \mathbf{p}(t_1), \dots$ (see Fig. 2) distributed along $\mathbf{p}(t)$ and use central finite difference approximation. Specifically, suppose that the mark points $\mathbf{p}(t_i)(i = \dots, -1, 0, 1, \dots)$ are mapped to corresponding image points $\mathbf{q}(t_i)$. Let the step size $\Delta t = t_i - t_{i-1}$. We may approximate $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ at \mathbf{q}_o by using 2 points and 3 points respectively, that is

$$\dot{\mathbf{q}} \approx (\mathbf{q}(t_1) - \mathbf{q}(t_{-1}))/(\Delta t) \quad (32)$$

$$\ddot{\mathbf{q}} \approx (\mathbf{q}(t_1) - 2\mathbf{q}(t_0) + \mathbf{q}(t_{-1})) / (\Delta t)^2. \quad (33)$$

The truncation error of the finite difference approximation (32) and (33) decays when Δt decreases. To analyze how this numerical approximation affects recovery of distance, curvature and third-order parameters of the mirror surface, we conducted numerical simulations on a synthetic mirror surface by implementing Eqs. (17), (18) and (24) in Matlab (see Fig. 3).

6 Experimental Results

In addition to numerical simulations on synthetic surfaces, we also validated our theoretical results by recovering local surface parameters of some real mirror objects. A Kodak DC290 digital camera with 1792×1200 resolution was used to take a picture of a mirror surface reflecting a checkerboard pattern of $2\text{cm} \times 2\text{cm}$ grid size. The mirror surface and camera were set about 30cm apart. The edges of the pattern grids acted as a pair of intersecting lines and corners served as mark points for our finite difference approximation. The pattern was placed such that both pattern and its specular reflection were clearly visible from the camera (see Fig. 1). The camera and pattern were calibrated using standard calibration routines. Our local surface reconstruction algorithm can be summarized as the following 8 steps:

1. Select a scene intersection point and its reflected point (e.g. 1 and 1' in Fig. 1).
2. Select four neighboring points from both checkerboard pattern (e.g. 2, 3, 4, 5) and corresponding reflected pattern (e.g. 2', 3', 4', 5').
3. From 1, 2, 3, 4, 5 compute \mathbf{p}_o and the direction of two scene lines $\Delta\mathbf{p}_1$ and $\Delta\mathbf{p}_2$.
4. From 1', 2', 3', 4', 5' estimate \mathbf{q}^m , $\dot{\mathbf{q}}_1^m$, $\dot{\mathbf{q}}_2^m$ and $\ddot{\mathbf{q}}_1^m$, $\ddot{\mathbf{q}}_2^m$ using Eqs. (32) and (33).
5. Recover the distance parameter s by Eq. (17) from $\dot{\mathbf{q}}_1^m$, $\dot{\mathbf{q}}_2^m$.
6. Recover the parameter r by Eq. (18) from $\dot{\mathbf{q}}_1^m$, $\dot{\mathbf{q}}_2^m$.
7. Recover curvature parameters (a, b, c) by Eq. (4).
8. Recover third-order surface parameters (e, f, g, h) by Eq. (24) from $\ddot{\mathbf{q}}_1^m$, $\ddot{\mathbf{q}}_2^m$.

We validated this algorithm with a specular teapot and a portion of car fender (see Fig. 5). The recovery of third-order surface parameters has been validated in Fig. 3 (bottom panel) using a synthetic mirror surface. In the teapot experiments we compared our reconstruction results with those obtained using three lines [2] in Fig. 4.

7 Conclusions

Under the assumption of an unknown mirror surface reflecting a known calibrated pattern (e.g. a checkerboard) onto the image plane of a calibrated camera, we demonstrated that surface position and shape up to third order can be

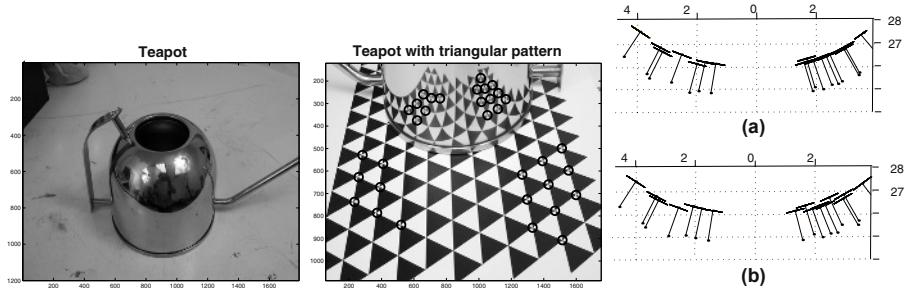


Fig. 4. Teapot Experiment. We compared our reconstruction results with those obtained using the 3-lines-approach [2] for a specular teapot (left panel). At that end, we used a special pattern composed of a tessellation of triangles in order to have a triplet of intersecting lines at each corner point. The bottom part of the teapot (a cylinder of diameter $d = 13.15$ cm) was reconstructed at each marked point (middle panel). The right panel compares reconstructed points obtained by 3-lines-method (a) and our method (b). Each point is represented by its tangent plane and its normal vector. Statistical analysis shows that these two methods exhibit similar performances in reconstructing position and surface normal at each intersecting point. Our approach, however, is more advantageous than the 3-lines-approach in that it can also estimate curvature parameters. Our recovered average principal curvatures are $\kappa_1 = -0.153 \pm 0.005$ and $\kappa_2 = 0.003 \pm 0.007$, which corresponds to an average estimated cylinder diameter of 13.097 cm.

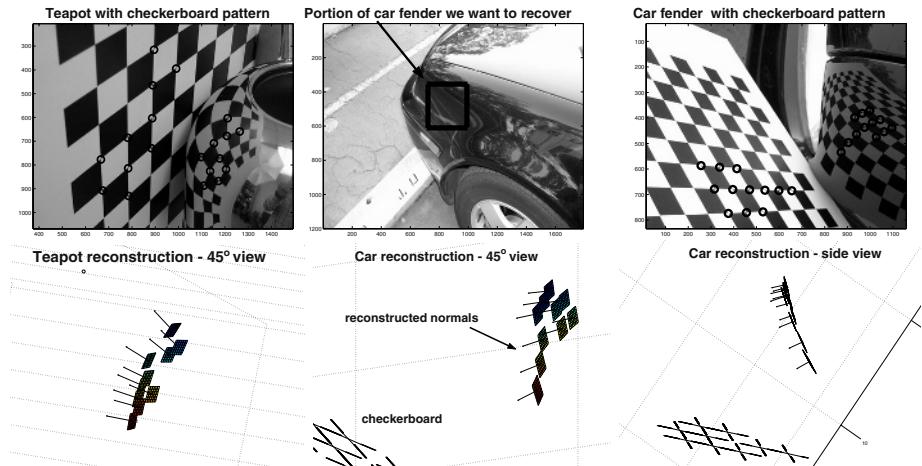


Fig. 5. Experimental results with real surfaces (teapot and car fender).

derived as a function of local position, orientation and local scale measurements in the image when two orientations are available at the same point of the image reflected pattern (e.g. a corner). We validated our theoretical results with both numerical simulations and experiments with real surfaces and found that the

method is practical and yields good quality surface reconstruction. Future work may be done to overcome the correspondence problem between pattern points and their reflected image points. Additional work may be also needed to remove the hypothesis of having a calibrated pattern, which will most likely require integrating additional cues (such as stereo views) and some form of prior knowledge on the likely statistics of the scene geometry.

References

1. S. Savarese and P. Perona: Local Analysis for 3D Reconstruction of Specular Surfaces. *IEEE Conf. on Computer Vision and Pattern Recognition*, II 738–745 (2001) *CVPR*, II 738–745 (2001).
2. S. Savarese and P. Perona: Local Analysis for 3D Reconstruction of Specular Surfaces - Part ii. *ECCV*, II 759–774 (2002).
3. S. Savarese, M. Chen, P. Perona: Second Order Local Analysis for 3D Reconstruction of Specular Surfaces. *3DPVT*, 356–360 (2002).
4. T. Binford: Inferring surfaces from images. *Artificial Intelligence*, **17** (1981) 205–244.
5. A. Blake: Specular stereo. *IJCAI* (1985) 973–976.
6. A. Blake and G. Brelstaff: Geometry from specularities. *ICCV Proc. of Int Conf. of Computer Vision* (1988) 394–403.
7. M. Chen and J. Arvo: Theory and Application of Specular Path Perturbation. *ACM Transactions on Graphics*. **19** (2000) 246–278.
8. R. Cipolla and P. Giblin: Visual motion of curves and surfaces. *Cambridge University Press* 2000.
9. M. Halsead, A. Barsky, S. Klein, and R. Mandell: Reconstructing curved surfaces from reflection patterns using spline surface fitting normals. *SIGGRAPH* (1996).
10. G. Healey and T. Binford: Local shape from specularity. *Computer Vision, Graphics, and Image Processing* **42** (1988) 62–86.
11. K. Ikeuchi: Determining surface orientation of specular surfaces by using the photometric stereo method. *IEEE PAMI* **3** (1981) 661–669.
12. J. Koenderink and A. van Doorn: Photometric invariants related to solid shape. *Optica Acta* **27** (1980) 981–996.
13. M. Oren and S. K.Nayar: A theory of specular surface geometry. *Trans. Int. Journal of Computer Vision* (1997) 105–124.
14. J. Zheng and A. Murata: Acquiring a complete 3d model from specular motion under the illumination of circular-shaped light sources. *IEEE PAMI* **8** (2000).
15. A. Zisserman, P. Giblin, and A. Blake: The information available to a moving observer from specularities. *Image and Video Computing* **7** (1989) 38–42.
16. D. Perard: Automated visual inspection of specular surfaces with structured-lighting reflection techniques. *PhD Thesis – VDI Verlag Nr. 869* (2001).
17. M. Tarini, H. Lensch, M. Goesele and H.P. Seidel: Shape from Distortion 3D Range Scanning of Mirroring Objects. *Proc. of SIGGRAPH, Sketches and Applications* (2002) 248
18. T. Bonfort and P. Sturm: Voxel Carving for Specular Surfaces. *Proceedings of the 9th IEEE International Conference on Computer Vision* (2003)

Structure of Applicable Surfaces from Single Views

Nail Gumerov, Ali Zandifar, Ramani Duraiswami, and Larry S. Davis

Perceptual Interfaces and Reality Lab, University of Maryland, College Park

{gumerov, alizand, ramani, lsd}@umiacs.umd.edu

Abstract. The deformation of *applicable* surfaces such as sheets of paper satisfies the differential geometric constraints of isometry (lengths and areas are conserved) and vanishing Gaussian curvature. We show that these constraints lead to a closed set of equations that allow recovery of the *full geometric structure* from a *single image* of the surface and knowledge of its undeformed shape. We show that these partial differential equations can be reduced to the Hopf equation that arises in non-linear wave propagation, and deformations of the paper can be interpreted in terms of the characteristics of this equation. A new exact integration of these equations is developed that relates the 3-D structure of the applicable surface to an image. The solution is tested by comparison with particular exact solutions. We present results for both the forward and the inverse 3D structure recovery problem.

Keywords: Surface, Differential Geometry, Applicable Surfaces, Shape from X

1 Introduction

When a picture or text printed on paper is imaged, we are presented with a problem of unwarping the captured digital image to its flat, fronto-parallel representation, as a preprocessing step before performing tasks such as identification, or Optical Character Recognition (OCR). In the case that the paper is flat, the problem reduces to one of undoing a projection of an initial shape such as a rectangle, and the rectification (or unwarping) can be achieved by computing a simple homography. A harder problem is when the piece of paper is itself deformed or bent. In this case the unwarping must undo both the effects of the three-dimensional bending of the surface, and the imaging process. The differential geometry of surfaces provides a very powerful set of relations for analysis of the unwarping. However, most quantitative use of differential geometry has been restricted to range data, while its use for image data has been primarily qualitative. The deformation of paper surfaces satisfies the conditions of isometry and vanishing Gaussian curvature. Here, we show that these conditions can be analytically integrated to infer the complete 3D structure of the surface from an image of its bounding contour.

Previous authors have attempted to enforce these conditions in 3D reconstruction. However, they essentially enforced these as *constraints* to a process of polynomial/spline fitting using data obtained on the surface [16]. In contrast, we *solve* these equations, and show that *information on the bounding contour is sufficient to determine structure completely*. Further, exact correspondence information along the bounding contour is not needed. We only need the correspondences of a few points, e.g., corners. Other than its

theoretical importance, our research can potentially benefit diverse computer vision applications, e.g. portable scanning devices, digital flattening of creased documents, 3D reconstruction without correspondence, and perhaps most importantly, OCR of scene text.

2 Previous Work

A seminal paper by Koenderink [7] addressed the understanding of 3D structure qualitatively from occluding contours in images. It was shown that the concavities and convexities of visual contours are sufficient to infer the local shape of a surface. Here, we perform quantitative recovery of 3D surface structure for the case of applicable surfaces. While we were not able to find similar papers dealing with analytical integration of the equations of differential geometry to obtain structure, the following papers deal with related problems of unwarping scene text, or using differential geometric constraints for reconstruction.

Metric rectification of planar surfaces: In [2,12,15] algorithms for performing metric rectification of planar surfaces were considered. These papers extract from the images, features such as vanishing lines and right angles and perform rectification. Extraction of vanishing lines is achieved by different methods; such as the projection profile method [2] and the illusory and non-illusory lines in textual layouts [15].

Undoing paper curl for non-planar surfaces knowing range data: A number of papers deal with correcting the curl of documents using known shape (e.g. cylinders) [11, 19]. These approaches all need 3D points on the surface to solve for the inverse mapping. In [16] sparse 3D data on the curled paper surface was obtained from a laser device. An approximate algorithm to fit an applicable surface through these points was developed that allowed obtaining dense depth data. The isometry constraint was approximately enforced by requiring that distances between adjacent nodes be constant. In [1] a mass-spring particle system framework was used for digital flattening of destroyed documents using depth measurements, though the differential geometry constraints are not enforced.

Isometric surfaces: In [10] an algorithm is developed to bend virtual paper without shearing or tearing. Ref. [13] considers the shape-from-motion problem for shapes deformed under isometric mapping.

3 Theory

3.1 Basic Surface Representation

A surface is the exterior boundary of an object/body. In a 3D world coordinate system, a surface $\mathbf{r} = \mathbf{r}(X, Y, Z)$, (where (X, Y, Z) is any point on the surface) is mathematically represented in explicit, implicit and parametric forms respectively as:

$$z = f(x, y), \quad F(x, y, z) = 0, \quad \mathbf{r}(u, v) = (X(u, v), Y(u, v), Z(u, v)). \quad (1)$$

Consider a smooth surface S expressed parametrically as:

$$\mathbf{r}(u, v) = (X(u, v), Y(u, v), Z(u, v)), \quad (2)$$

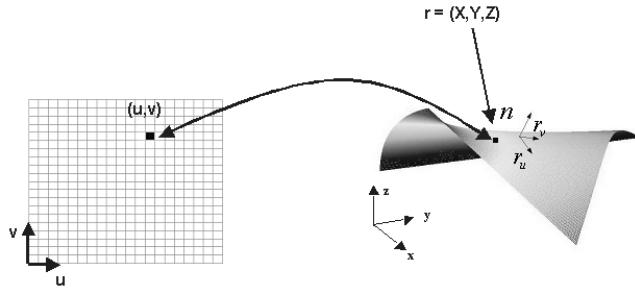


Fig. 1. Parametric representation of a surface

which is a mapping from any point (u, v) in the parametric (or undeformed) plane (uv -plane) to a point (X, Y, Z) on the surface in 3D (Figure 3). The sets $\{\mathbf{r}(u, v), v = \text{const}\}$ and $\{\mathbf{r}(u, v), u = \text{const}\}$ represent two families of curves on the surface, whose partial derivatives are tangent vectors to the curves $v = \text{const}$ and $u = \text{const}$ respectively. These derivatives are often called *tangent vectors* [9]. Let the second derivatives of \mathbf{r} with respect to u and v be \mathbf{r}_{uu} , \mathbf{r}_{uv} and \mathbf{r}_{vv} . The element of distance $ds = |\mathbf{dr}|$ on the surface is given at each surface point (u, v) by the *first fundamental form* of a surface

$$ds^2 = |\mathbf{dr}|^2 = \|\mathbf{r}_u\|^2 du^2 + 2\mathbf{r}_u \cdot \mathbf{r}_v dudv + \|\mathbf{r}_v\|^2 dv^2 = E du^2 + 2F dudv + G dv^2,$$

$$E(u, v) = \|\mathbf{r}_u\|^2, \quad F(u, v) = \mathbf{r}_u \cdot \mathbf{r}_v, \quad G(u, v) = \|\mathbf{r}_v\|^2.$$

The surface coordinates are orthogonal iff $F \equiv 0$. The surface normal \mathbf{n} and area element $d\mathbf{n}$ can be defined in terms of the tangent vectors as:

$$\mathbf{n} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{\|\mathbf{r}_u \times \mathbf{r}_v\|} = \sqrt{EG - F^2}, \quad d\mathbf{n} = \|\mathbf{r}_u \times \mathbf{r}_v\| dudv = \sqrt{EG - F^2} dudv. \quad (3)$$

The *second fundamental form* of a surface at a point (u, v) measures how far the surface is from being planar. It is given by

$$-d\mathbf{r} \cdot d\mathbf{n} = L(u, v)du^2 + 2M(u, v)dudv + N(u, v)dv^2, \quad (4)$$

where L , M and N are standard and defined e.g., in [9]. For every normal section through (u, v) there exist two principal curvatures (k_1, k_2) . The mean and Gaussian curvature; $H(u, v)$ and $K(u, v)$ are

$$H \equiv \frac{k_1 + k_2}{2} = \frac{1}{2} \frac{EN - 2FM + GL}{EG - F^2}, \quad K \equiv k_1 k_2 = \frac{LN - M^2}{EG - F^2}. \quad (5)$$

3.2 Special Surfaces

Let us assume that we have a mapping of a point in the parametric plane (u, v) to a point in 3D (X, Y, Z) . The mapping is *isometric* if the length of a curve or element of area is invariant with the mapping, i.e.

$$E(u, v) = \|\mathbf{r}_u\|^2 = 1, \quad F(u, v) = \mathbf{r}_u \cdot \mathbf{r}_v = 0, \quad G(u, v) = \|\mathbf{r}_v\|^2 = 1. \quad (6)$$

Lengths and areas are conserved in an isometric mapping

$$ds^2 = |d\mathbf{r}|^2 = E(u, v)du^2 + 2F(u, v)dudv + G(u, v)dv^2 = du^2 + dv^2,$$

$$dA = \sqrt{EG - F^2} dudv = dudv.$$

The mapping is *conformal* if the angle between curves on a surface is invariant of the mapping ($F = 0$). It is *developable* if the Gaussian curvature is zero everywhere.

$$K = 0 \implies LN - M^2 = 0. \quad (7)$$

It is *applicable* if the surface is isometric with a flat surface (Eq. 6) and the Gaussian curvature vanishes (Eq. 7) for every point on the surface.

3.3 Differential Equations for Applicable Surfaces

If we differentiate Eq. (6), we have:

$$\mathbf{r}_{uu} \cdot \mathbf{r}_u = \mathbf{r}_{uu} \cdot \mathbf{r}_v = \mathbf{r}_{uv} \cdot \mathbf{r}_u = \mathbf{r}_{uv} \cdot \mathbf{r}_v = \mathbf{r}_{vv} \cdot \mathbf{r}_u = \mathbf{r}_{vv} \cdot \mathbf{r}_v = 0. \quad (8)$$

This shows that $\mathbf{r}_{uu} = (X_{uu}, Y_{uu}, Z_{uu})$, $\mathbf{r}_{uv} = (X_{uv}, Y_{uv}, Z_{uv})$ and $\mathbf{r}_{vv} = (X_{vv}, Y_{vv}, Z_{vv})$ are perpendicular to \mathbf{r}_u and \mathbf{r}_v and consequently, are collinear with the normal vector to the surface.

$$\mathbf{n} \parallel (\mathbf{r}_u \times \mathbf{r}_v) \parallel \mathbf{r}_{uu} \parallel \mathbf{r}_{uv} \parallel \mathbf{r}_{vv}, \quad (9)$$

where \parallel denotes 'is parallel to'. We can thus express \mathbf{n} as

$$\mathbf{n} = a\mathbf{r}_{uu} = b\mathbf{r}_{uv} = c\mathbf{r}_{vv}. \quad (10)$$

We can rewrite (7) using (10) as:

$$LN - M^2 = 0 \implies a\|\mathbf{n}\|^2 c\|\mathbf{n}\|^2 - b^2\|\mathbf{n}\|^2\|\mathbf{n}\|^2 = 0 \implies ac - b^2 = 0, \quad (11)$$

where a, b , and c are scalars, and

$$\frac{\mathbf{r}_{uv}}{\mathbf{r}_{uu}} = \frac{a}{b} = \frac{b}{c} = \frac{\mathbf{r}_{vv}}{\mathbf{r}_{uv}}. \quad (12)$$

Therefore from (12) we have:

$$\frac{\partial^2 W}{\partial v^2} \frac{\partial^2 W}{\partial u^2} = \left(\frac{\partial^2 W}{\partial u \partial v} \right)^2, \quad \text{for } W = X, Y, Z. \quad (13)$$

Solving the set of nonlinear higher order partial differential equations (PDEs) (Eq. 13), we can compute the surface structure \mathbf{r} in 3D, given boundary conditions (curves) for an applicable surface. These equations may be solved by conventional methods of solving PDEs e.g. Finite Differences or FEM. However, we provide a much more efficient method, based on reducing the solution to integration of several simultaneous ODEs.

3.4 A First Integration: Reduction to ODEs

Let $W_u = \partial W / \partial u$, $W_v = \partial W / \partial v$. The functions $W_u(u, v)$ and $W_v(u, v)$ satisfy the consistency conditions

$$\frac{\partial W_u}{\partial v} = \frac{\partial W_v}{\partial u}, \quad W = X, Y, Z. \quad (14)$$

i.e. cross-derivatives are the same. From Eqs. (13) and (14) we have

$$\frac{\partial W_u}{\partial u} \frac{\partial W_v}{\partial v} - \frac{\partial W_u}{\partial v} \frac{\partial W_v}{\partial u} = \frac{\partial (W_u, W_v)}{\partial (u, v)} = 0. \quad (15)$$

Therefore Eq. (13) can be treated as a degeneracy condition for the Jacobian of the mapping from $(u, v) \mapsto (W_u, W_v)$. This degeneracy means that the functions W_u and W_v are functions of a single variable, t , which in turn is a function of (u, v) . In other words:

$$\exists t = t(u, v) \text{ such that } W_u(u, v) = W_u(t), \quad W_v(u, v) = W_v(t), \quad (16)$$

where $W = X, Y, Z$. In this case $t = \text{const}$ is a line in the parametric plane. Since W denotes any of X, Y and Z , Eq. (16) could hold separately for each component, with some different mapping functions $t_x(u, v)$, $t_y(u, v)$, and $t_z(u, v)$ specific to each coordinate. However, these functions must all be equal because all are functions of the single variable $t(u, v)$, which can be called the *mapping or characteristic function* for the surface S . Therefore,

$$\mathbf{r}_u = \mathbf{r}_u(t), \quad \mathbf{r}_v = \mathbf{r}_v(t), \quad (17)$$

where $t = t(u, v)$. Denoting by the superscript dot the derivative of a function with respect to t , we can write \mathbf{r}_{uu} and \mathbf{r}_{vv} as

$$\mathbf{r}_{uu} = \dot{\mathbf{r}}_u \frac{\partial t}{\partial u}, \quad \mathbf{r}_{vv} = \dot{\mathbf{r}}_v \frac{\partial t}{\partial v}. \quad (18)$$

From Eq. (9, 18), we see that $\dot{\mathbf{r}}_u$ and $\dot{\mathbf{r}}_v$ are collinear with the surface normal i.e. $\dot{\mathbf{r}}_u \parallel \mathbf{n}$, $\dot{\mathbf{r}}_v \parallel \mathbf{n}$. Let us define a new vector \mathbf{w} as :

$$\mathbf{w} = u\dot{\mathbf{r}}_u(t) + v\dot{\mathbf{r}}_v(t). \quad (19)$$

Also note that \mathbf{w} is a function of the characteristic variable t , since the Jacobian of a mapping from $(u, v) \mapsto (t, \mathbf{m} \cdot \mathbf{w})$ for a constant vector \mathbf{m} vanishes:

$$\begin{aligned} \frac{\partial (t, \mathbf{w} \cdot \mathbf{m})}{\partial (u, v)} &= \frac{\partial t}{\partial u} \frac{\partial \mathbf{w} \cdot \mathbf{m}}{\partial v} - \frac{\partial t}{\partial v} \frac{\partial \mathbf{w} \cdot \mathbf{m}}{\partial u} = \frac{\partial t}{\partial u} \dot{\mathbf{r}}_v(t) \cdot \mathbf{m} - \frac{\partial t}{\partial v} \dot{\mathbf{r}}_u(t) \cdot \mathbf{m} \\ &= \mathbf{r}_{uv} \cdot \mathbf{m} - \mathbf{r}_{uv} \cdot \mathbf{m} \implies \frac{\partial (t, \mathbf{w} \cdot \mathbf{m})}{\partial (u, v)} = 0. \end{aligned}$$

This means that \mathbf{w} is a function of t alone; $\mathbf{w} = \mathbf{w}(t)$. From collinearity of \mathbf{w} with $\dot{\mathbf{r}}_u$ and $\dot{\mathbf{r}}_v$ it follows that two scalar functions $h_u(t)$ and $h_v(t)$ can be introduced as

$$\dot{\mathbf{r}}_u(t) = h_u(t) \mathbf{w}(t), \quad \dot{\mathbf{r}}_v(t) = h_v(t) \mathbf{w}(t). \quad (20)$$

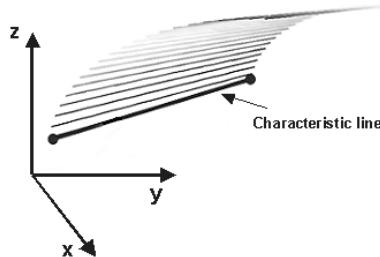


Fig. 2. Characteristics lines as generator lines

By (20), and from Eq. (19), we have

$$uh_u(t) + vh_v(t) = 1, \quad h_v(t)\dot{\mathbf{r}}_u(t) - h_u(t)\dot{\mathbf{r}}_v(t) = 0. \quad (21)$$

Therefore, Eq.(21) defines a characteristic line in the uv -plane for $t = const$. While the latter equation provides a relation between functions of t , the former implicitly determines $t(u, v)$. Since $h_u(t)$ and $h_v(t)$ are known, Eq. (21) gives $t(u, v)$. Note that t satisfies the equation

$$h_v(t)\frac{\partial t}{\partial u} - h_u(t)\frac{\partial t}{\partial v} = 0, \quad (22)$$

which is a *Hopf-type* equation, a common nonlinear hyperbolic equation in shock-wave theory [4]. The characteristics of this equation are $t(u, v)$ which satisfies

$$t(u, v) = t(u + c(t)v), \quad c(t) = \frac{h_u(t)}{h_v(t)}. \quad (23)$$

Therefore, for any $t = const$ the characteristic is a line in the uv -plane. The properties of the Hopf equation are well studied in the theory of propagation of shock waves in nonlinear media ([4]). Along the characteristics, $t = const$, all functions of t are constant, including $h_u(t)$ and $h_v(t)$. As follows from Eq. (21), in the (u, v) -plane these characteristics are straight lines. The lines corresponding to characteristics are also straight lines on the surface. In fact to generate an applicable surface, we can sweep a line in space and the generated envelope will be applicable. Through every point on the surface there is a straight line as shown (Figure 2) by:

$$\mathbf{r}(t) = u\mathbf{r}_u(t) + v\mathbf{r}_v(t) + \rho(t) \quad , \quad \dot{\rho}(t) = -\mathbf{w}(t), \quad (24)$$

The above equations are sufficient to solve the basic warping and unwarping problems for images based on information about the shapes of the image boundaries. The goal is to find for any characteristic line, the variables $\mathbf{r}_u(t)$, $\mathbf{r}_v(t)$, $\rho(t)$, $h_u(t)$ and $h_v(t)$ and, finally, $\mathbf{r}(t)$ from available information. To summarize the differential and algebraic relations for applicable surfaces, we have

$$\begin{aligned} \mathbf{r}(u, v) &= u\mathbf{r}_u(t) + v\mathbf{r}_v(t) + \rho(t), \quad \dot{\mathbf{r}}_u(t) = h_u(t)\mathbf{w}(t), \quad \dot{\mathbf{r}}_v(t) = h_v(t)\mathbf{w}(t), \\ \rho(t) &= -\mathbf{w}(t), \quad uh_u(t) + vh_v(t) = 1, \quad \|\mathbf{r}_u\|^2 = 1, \quad \mathbf{r}_u \cdot \mathbf{r}_v = 0, \quad \|\mathbf{r}_v\|^2 = 1. \end{aligned} \quad (25)$$

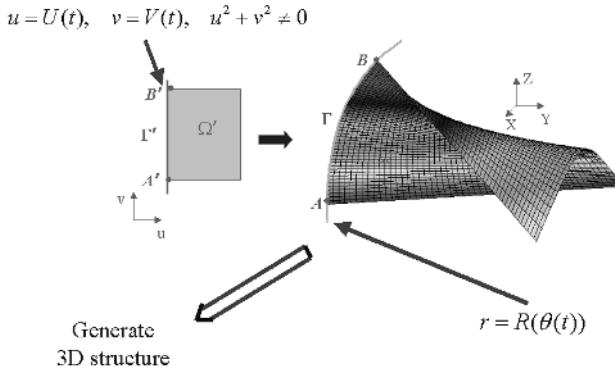


Fig. 3. Generation of an applicable surface with a 3D curve. In this example a straight line Γ' in the uv -plane is mapped on a given 3D curve Γ .

3.5 Forward Problem: Surface with a Specified Boundary Curve

Here, we specify the bending of a flat page in 3D so that one edge conforms to a given 3D curve. We call this the forward problem. We generate the warped surface to demonstrate the solution to Eq. (25).

Let Γ' be an open curve on a patch $\Omega' \subset P$ in the uv -plane, corresponding to an open curve Γ in 3D. To generate an applicable surface in 3D, knowledge of the corresponding curves Γ' and Γ and the patch boundaries in the uv -plane (Figure 3) are sufficient. We know that the curve Γ' starts from a point $A' = (u_0, v_0)$ and the corresponding curve Γ passes from $A = (X_0, Y_0, Z_0)$ and the point B corresponds to the point B' . Due to isometry, the length of the two curves are the same, and there is a one-to-one mapping from a domain $\Omega' \subset P$ to $\Omega \subset S$, which are respectively bounded by Γ' and Γ . For any point $(u^*, v^*) \in \Omega'$ there exists a characteristic, $t = t^*$, which also passes through some point on Γ' . Assume now that Γ' is specified by the parametric equations

$$u = U(t), \quad v = V(t), \quad u^2 + v^2 \neq 0.$$

Without loss of generality, we can select t to be a natural parameterization of Γ' , measured from point A' ; i.e. the arc length s along the curve Γ' , measured from the curve starting point $t = t_0$,

$$s \equiv \int_{t_0}^t ds \equiv \int_{t_0}^t \sqrt{dr \cdot dr}. \quad (26)$$

parametrizes the curve. Let $\Gamma' : (U(t), V(t))$ be in $[t_{\min}, t_{\max}]$. If we represent Γ in parametric form as $\mathbf{r} = \mathbf{R}(t)$, then due to isometry, t will also be a natural parameter for Γ' , and

$$\dot{U}^2 + \dot{V}^2 = 1, \quad \dot{\mathbf{R}} \cdot \dot{\mathbf{R}} = 1. \quad (27)$$

The surface equations for any $(u, v) \in \Omega'$ are

$$\begin{aligned} \mathbf{r}_u \cdot \mathbf{r}_u &= 1, \quad \mathbf{r}_u \cdot \mathbf{r}_v = 0, \quad \mathbf{r}_v \cdot \mathbf{r}_v = 1, \\ Uh_u + Vh_v &= 1, \quad h_v \dot{\mathbf{r}}_u - h_u \dot{\mathbf{r}}_v = \mathbf{0}, \quad U\mathbf{r}_u + V\mathbf{r}_v + \rho = \mathbf{R}. \end{aligned} \quad (28)$$

While the number of unknowns here is 11 ($\mathbf{r}_u, \mathbf{r}_v, \rho, h_u, h_v$) and the number of equations are 12 (Eqs. 27,28) but two of them are dependent (Eqs. including h_u and h_v). For unique solution of Eqs. (27,28), we differentiate Eq. (27) to obtain sufficient equations to solve the forward problem

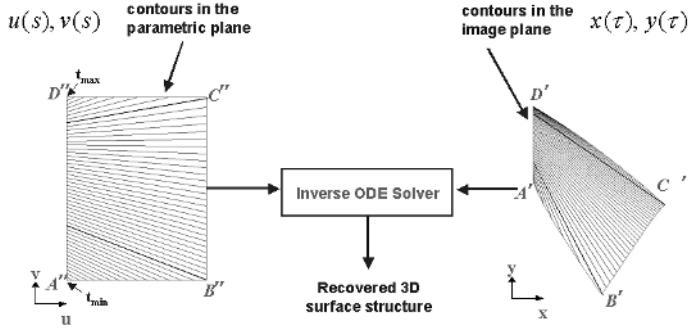
$$\begin{aligned} \dot{\mathbf{r}}_u &= \frac{h_u \mathbf{F}}{\dot{U}h_u + \dot{V}h_v}, \quad \dot{\mathbf{r}}_v = \frac{h_v \mathbf{F}}{\dot{U}h_u + \dot{V}h_v}, \quad h_u = \frac{g_u}{Vg_v + Ug_u}, \quad h_v = \frac{g_v}{Vg_v + Ug_u}, \\ \mathbf{F} &= \ddot{\mathbf{R}} - \ddot{U}\mathbf{r}_u - \ddot{V}\mathbf{r}_v, \quad g_u = \ddot{U} - \ddot{\mathbf{R}} \cdot \mathbf{r}_u, \quad g_v = \ddot{V} - \ddot{\mathbf{R}} \cdot \mathbf{r}_v. \end{aligned} \quad (29)$$

These equations must be integrated numerically using, e.g., the Runge-Kutta method [17]. To generate the structure of the applicable surface we need for any characteristic line, the functions $\mathbf{r}_u(t), \mathbf{r}_v(t)$ and $\rho(t)$; $(\mathbf{r}_u(t), \mathbf{r}_v(t))$ are obtained from the solution to ODEs, while $\rho(t)$ is computed from the fifth equation in (28). The solution to our problem is a two-point boundary value problem (bvp). Most software for ODEs is written for initial value problems. To solve a bvp using an initial value solver, we need to estimate $\mathbf{r}_{u0} = \mathbf{r}_u(0)$ and $\mathbf{r}_{v0} = \mathbf{r}_v(0)$. which achieves the correct boundary value. The vectors \mathbf{r}_{u0} and \mathbf{r}_{v0} are dependent, since they satisfy the first three equations (28), which describe two orthonormal vectors. Assuming that $(\mathbf{r}_u, \mathbf{r}_v, \mathbf{r}_u \times \mathbf{r}_v)$ is a right-handed basis, we can always rotate the reference frame of the world coordinates so that in the rotated coordinates we have $\mathbf{r}_{u0} = (1, 0, 0)$, $\mathbf{r}_{v0} = (0, 1, 0)$. Consistent initial conditions \mathbf{r}_{u0} and \mathbf{r}_{v0} for Eq. (28) can be obtained by application of a rotation matrix $Q(\alpha, \beta, \gamma)$ with Euler angles α, β and γ , to the vectors $(1, 0, 0)$ and $(0, 1, 0)$, respectively. We also can note that for some particular cases it may happen that both the functions g_v and g_u in Eq. (29) may be zero. In this case the equations for h_u and h_v can be replaced by the limiting expressions for $g_v \rightarrow 0, g_u \rightarrow 0$. In the special case (rectangular patch in the parametric plane), we can show that there is an analytical solution given by:

$$\mathbf{r}_u = \frac{\ddot{\mathbf{R}} \times \dot{\mathbf{R}}}{|\ddot{\mathbf{R}}|}, \quad \mathbf{r}_v = \dot{\mathbf{R}}. \quad (30)$$

3.6 Inverse Problem: 3D Structure Recovery of Applicable Surfaces

Here, we seek to estimate the 3D structure of an applicable surface from a single view (with known camera model) and knowledge of the undeformed uv plane boundary. For any point (x, y) in the image plane, we can estimate the corresponding point in the uv -plane and vice versa by solving the ODEs for the problem. The input parameters are the known camera model, the patch contours in the uv -plane and the image plane. Assume that the image of the patch (Ω') is bounded by two curves Γ'_1 and Γ'_2 , the corresponding

**Fig. 4.** Inverse Problem Schematic

patch (Ω) in the uv -plane is bounded by Γ_1 and Γ_2 and that the patch Ω bounded by the two characteristics, $t = t_{\min}$, and $t = t_{\max}$ (Fig. 4). We assume that Γ_1 and Γ_2 are piecewise continuous curves in the uv -plane, and not tangential to the characteristic lines $t_{\min} < t < t_{\max}$. For any point $(u_*, v_*) \in \Omega$ there exists a characteristic, $t = t_*$, which passes through some points on Γ_1 and some points on Γ_2 . In the uv -plane these curves can be specified by a natural parameterization $u = U_1(s_1)$, $v = V_1(s_1)$ for Γ_1 , and $u = U_2(s_2)$, $v = V_2(s_2)$ for Γ_2 , with $u^2 + v^2 \neq 0$. Here $s_1(t)$ and $s_2(t)$ are unknown and must be found in the process of solution.

Γ_1 and Γ_2 correspond to the 3D curves $\mathbf{r} = \mathbf{r}_1(t)$ and $\mathbf{r} = \mathbf{r}_2(t)$, which are unknown and found in the process of solution. Note that at the starting point or end point, Γ_1 and Γ_2 may intersect. At such a point the characteristic $t = t_{\min}$ or $t = t_{\max}$ is tangential to the boundary or the boundary is not smooth (e.g. we are at a corner). In case Γ_1 and Γ_2 intersect at $t = t_{\min}$ and $t = t_{\max}$ they completely define the boundary of the patch Ω . These cases are *not* special and can be handled by the general method described below. Assume that the camera is calibrated, and the relation between the world coordinates $\mathbf{r} = (X, Y, Z)$ and coordinates of the image plane (x, y) are known as $x = F_x(\mathbf{r})$ and $y = F_y(\mathbf{r})$. What is also known are the equations for Γ'_1 and Γ'_2 that are images of the patch boundaries Γ_1 and Γ_2 . These equations, assumed to be in the form $x = x_1(\tau_1)$, $y = y_1(\tau_1)$ for Γ'_1 ; and $x = x_2(\tau_2)$, $y = y_2(\tau_2)$ for Γ'_2 . Here τ_1 and τ_2 are the natural parameters of these curves; $\tau_1(t)$ and $\tau_2(t)$ are obtained from the solution. The specification of the curve parameters as 'natural' means:

$$U_i'^2 + V_i'^2 = 1, \quad x_i'^2 + y_i'^2 = 1, \quad i = 1, 2. \quad (31)$$

A complete set of equations describing the surface can be reduced then to

$$\begin{aligned} \mathbf{r}_u \cdot \mathbf{r}_u &= 1, \quad \mathbf{r}_u \cdot \mathbf{r}_v = 0, \quad \mathbf{r}_v \cdot \mathbf{r}_v = 1, \\ \mathbf{r}_2 &= (U_2 - U_1) \mathbf{r}_u + (V_2 - V_1) \mathbf{r}_v + \mathbf{r}_1, \quad \dot{\mathbf{r}}_i = \dot{s}_i (U_i' \mathbf{r}_u + V_i' \mathbf{r}_v), \\ F_x(\mathbf{r}_i) &= x_i(\tau_i), \quad F_y(\mathbf{r}_i) = y_i(\tau_i), \quad i = 1, 2. \end{aligned} \quad (32)$$

We have 16 equations relating the 15 unknowns $(\mathbf{r}_u, \mathbf{r}_v, \mathbf{r}_1, \mathbf{r}_2, s_1, s_2, \tau_1, \tau_2)$. As in the previous case, one equation depends the other 15 and so the system is consistent. After $s(t)$, $\mathbf{r}_1(t)$, $\mathbf{r}_u(t)$, and $\mathbf{r}_v(t)$ are found, h_u , h_v , and ρ can be determined as

$$h_u = \frac{V_2 - V_1}{U_1 V_2 - U_2 V_1}, \quad h_v = \frac{U_1 - U_2}{U_1 V_2 - U_2 V_1}, \quad \rho = \mathbf{r}_1 - U_1 \mathbf{r}_u - V_1 \mathbf{r}_v. \quad (33)$$

This enables determination of $t(u, v)$ and $\mathbf{r}(u, v)$, similar to the forward problem. Here too the vector \mathbf{w} is collinear to the normal to the surface (Eq. 19) and satisfies $\mathbf{w} = k\mathbf{n}$. Let the rate of change of s_1 be a constant, s_{10} . The ODEs containing the unknowns ($s_1, s_2, \tau_1, \tau_2, \mathbf{r}_u, \mathbf{r}_v, \rho$) can be written as follows:

$$\begin{aligned} s_1 &= \dot{s}_{10}t, \quad \dot{\tau}_1 = \dot{s}_{10}\mathbf{c}_1 \cdot \mathbf{a}_1, \quad \dot{s}_2 = -\frac{k\mathbf{f}_2 \cdot \mathbf{b}_2}{\mathbf{e}_2 \cdot \mathbf{b}_2 + \mathbf{c}_2 \cdot [(\mathbf{c}_2 \cdot \mathbf{a}_2)\mathbf{d}_2 + \mathbf{G}_2 \cdot \mathbf{c}_2]}, \\ \dot{\tau}_2 &= \dot{s}_2\mathbf{c}_2 \cdot \mathbf{a}_2, \quad k = -\frac{\mathbf{e}_1 \cdot \mathbf{b}_1 + \mathbf{c}_1 \cdot [(\mathbf{c}_1 \cdot \mathbf{a}_1)\mathbf{d}_1 + \mathbf{G}_1 \cdot \mathbf{c}_1]}{\mathbf{f}_1 \cdot \mathbf{b}_1} \dot{s}_{10}, \quad \dot{\mathbf{r}}_u = kh_u\mathbf{n}, \\ \dot{\mathbf{r}}_v &= kh_v\mathbf{n}, \quad \dot{\rho} = -k\mathbf{n}, \quad h_u = \frac{v_2 - v_1}{u_1 v_2 - u_2 v_1}, \quad h_v = \frac{u_1 - u_2}{u_1 v_2 - u_2 v_1}, \\ \mathbf{a}_i(\tau_i, \mathbf{r}_i) &= \frac{x'_i \nabla F_x(\mathbf{r}_i) + y'_i \nabla F_y(\mathbf{r}_i)}{x'^2_i + y'^2_i}, \quad \mathbf{b}_i(\tau_i, \mathbf{r}_i) = y'_i \nabla F_x(\mathbf{r}_i) - x'_i \nabla F_y(\mathbf{r}_i), \\ \mathbf{c}_i(s_i, \mathbf{r}_u, \mathbf{r}_v) &= u'_i \mathbf{r}_u + v'_i \mathbf{r}_v, \quad \mathbf{d}_i = y''_i \nabla F_x(\mathbf{r}_i) - x''_i \nabla F_y(\mathbf{r}_i), \quad \mathbf{e}_i = u''_i \mathbf{r}_u + v''_i \mathbf{r}_v, \\ \mathbf{f}_i &= (u'_i h_u + v'_i h_v) \mathbf{n}, \quad \mathbf{G}_i = y'_i \nabla \nabla F_x(\mathbf{r}_i) - x'_i \nabla \nabla F_y(\mathbf{r}_i). \end{aligned} \quad (34)$$

To start the integration of the inverse problem, we need initial conditions for $(s_1, s_2, \tau_1, \tau_2, \mathbf{r}_u, \mathbf{r}_v, \rho)$.

Solution to the Boundary Value Problem: While the equation above can be solved for a general camera model, we will consider the simple orthographic case here. We can show these initial values here are:

$$\begin{aligned} t_0 &= s_{10} = s_{20} = \tau_{10} = \tau_{20} = 0, \quad \mathbf{r}_{10} = \mathbf{r}_{20} = \mathbf{r}_0, \\ u_{10} &= u_{20} = u_0, \quad v_{10} = v_{20} = v_0, \quad x_{10} = x_{20} = F_x(\mathbf{r}_0), \quad y_{10} = y_{20} = F_y(\mathbf{r}_0), \end{aligned}$$

and for the starting point in 3D, $\mathbf{r}_0 = \mathbf{r}_0(x_0, y_0, z_0)$ where z_0 is some free parameter in the orthographic case. Note also that at the initial point the formulae for h_u and h_v

$$h_u = \frac{v_2 - v_1}{u_1 v_2 - u_2 v_1}, \quad h_v = \frac{u_1 - u_2}{u_1 v_2 - u_2 v_1}. \quad (35)$$

are not acceptable, since the numerators and denominators are zero. However, we can find h_{u0} and h_{v0} from

$$u_0 h_{u0} + v_0 h_{v0} = 1, \quad \dot{s}_{10}(u'_{10} h_{u0} + v'_{10} h_{v0}) = \dot{s}_{20}(u'_{20} h_{u0} + v'_{20} h_{v0}). \quad (36)$$

The solution of this linear system specifies h_{u0} and h_{v0} as a function of \dot{s}_{20} , which can be estimated from the free parameter, and is in fact one of the Euler angles γ_0 . Recalling that $(\mathbf{r}_u, \mathbf{r}_v, \mathbf{r}_u \times \mathbf{r}_v)$ is a right-handed basis, we can rotate the reference frame of the world coordinates by Euler angles $(\alpha_0, \beta_0, \gamma_0)$ so that we have $\mathbf{r}_{u0} = (1, 0, 0)$, $\mathbf{r}_{v0} = (0, 1, 0)$. Further:

$$\begin{aligned} \dot{s}_{10} \mathbf{e}_{10} \cdot \mathbf{b}_{10} + k_0 \mathbf{f}_{10} \cdot \mathbf{b}_{10} + \dot{s}_{10} \mathbf{c}_{10} \cdot [(\mathbf{c}_{10} \cdot \mathbf{a}_{10}) \mathbf{d}_{10} + \mathbf{G}_{10} \cdot \mathbf{c}_{10}] &= 0, \\ \dot{s}_{20} \mathbf{e}_{20} \cdot \mathbf{b}_{20} + k_0 \mathbf{f}_{20} \cdot \mathbf{b}_{20} + \dot{s}_{20} \mathbf{c}_{20} \cdot [(\mathbf{c}_{20} \cdot \mathbf{a}_{20}) \mathbf{d}_{20} + \mathbf{G}_{20} \cdot \mathbf{c}_{20}] &= 0, \\ \mathbf{c}_{10} \cdot \mathbf{b}_{10} = 0, \quad \mathbf{c}_{20} \cdot \mathbf{b}_{20} = 0. \end{aligned} \quad (37)$$

These 4 relations can be treated as equations relating the 10 unknowns k_0 , \mathbf{r}_{u0} , \mathbf{r}_{v0} , \mathbf{n}_0 (\mathbf{r}_{u0} , \mathbf{r}_{v0} and \mathbf{n}_0 are 3D vectors). Also \mathbf{r}_{u0} , \mathbf{r}_{v0} , and \mathbf{n}_0 form an orthonormal basis, which therefore can be completely described by the three Euler angles $(\alpha_0, \beta_0, \gamma_0)$:

$$\mathbf{r}_{u0} = Q_0 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{r}_{v0} = Q_0 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{n}_0 = Q_0 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

where Q_0 is the Euler rotation matrix. This shows that \mathbf{r}_{u0} , \mathbf{r}_{v0} , and \mathbf{n}_0 a three-parameter set depending on $(\alpha_0, \beta_0, \gamma_0)$. Thus the relations Eq. (37) can be treated as 4 equations with respect to the unknowns $k_0, \alpha_0, \beta_0, \gamma_0$, for given \dot{s}_{20} or $k_0, \alpha_0, \beta_0, \dot{s}_{20}$ for given γ_0 , and can be solved. Then

$$\rho_0 = \mathbf{r}_0 - u_0 \mathbf{r}_{u0} - v_0 \mathbf{r}_{v0}. \quad (38)$$

determines ρ_0 as soon as \mathbf{r}_{u0} , \mathbf{r}_{v0} , and \mathbf{r}_0 are specified. Furthermore, we can reduce the four equations above to one nonlinear equation, whose roots can be determined by conventional numerical methods [17].

We found that this equation has two solutions, and so the Euler angles have four possible values. By choosing the free parameter γ_0 (Orthographic case), we can set all the initial conditions needed for the inverse problem. The challenge is to get the best estimate of γ_0 so that the boundary condition specifying correspondence points (such as the corners) is achieved. This is called the *shooting method*. We do this by minimizing a cost function J :

$$J = \arg \min_{\gamma_0} \|(x_e, y_e) - \mathbf{F}(\mathbf{r}(t_{\max}; \gamma_0, \Gamma_1, \Gamma_2, \Gamma'_1, \Gamma'_2))\|, \quad (39)$$

where (x_e, y_e) is the image coordinates of the 3D surface ending point (X_e, Y_e, Z_e) and $\mathbf{r}(t_{\max}; \gamma_0, \Gamma_1, \Gamma_2, \Gamma'_1, \Gamma'_2)$ is the last step of the 3D structure solution and \mathbf{F} is the camera model function. It is clear that $\mathbf{F}(\mathbf{r}(t_{\max}; \gamma_0, \Gamma_1, \Gamma_2, \Gamma'_1, \Gamma'_2))$ is the ending point of 3D surface calculated by the ODE solver. Therefore, we change the free parameter γ_0 until we can hit the ending corner or are within a specified tolerance of the ending point in the image plane. If the number of the correspondence points on the edge available exceeds the number of shooting parameters (say the 4 corners) a least-square approach can be used.

Ambiguities: As stated in the inverse problem, the method relies on the boundary information of the patch in the image plane. So, since some deformations can lead us to the same images of the boundary, we have ambiguities. In these cases we need to extract other useful cues such as texture or shading to resolve the ambiguities. This is the subject for future work.

4 Discussion and Results

4.1 Simple Validation of the Forward Problem

The purpose of this paper is to present and validate the new method. For this purpose we implemented the solution in algorithms. In the validation stage, we compared the results for warping to a 3D curve with the following analytical solution corresponding to a cylindrical surface

$$X = u - u_{\min}, \quad Y = N \cos \varphi(v), \quad Z = N \sin \varphi(v), \quad \varphi(v) = v/N. \quad (40)$$

To reproduce this surface we started our algorithm for warping with a 3D curve with the condition that in the (u, v) -plane the curve is a straight line, $u = u_{\min}$, and the fact that the corresponding 3D curve is

$$X(t) = 0, \quad Y(t) = N \cos \varphi(t), \quad Z(t) = N \sin \varphi(t). \quad (41)$$

For this surface we have the initial conditions for integration as $\mathbf{r}_{u0} = (-1, 0, 0)$, $\mathbf{r}_{v0} = (0, -\sin \varphi_0, \cos \varphi_0)$ with $\varphi_0 = v_{\min}/N$. We integrated the forward problem Eq. (29) numerically using an ODE solver from MATLAB, which was based on the 4th order Runge-Kutta method. The results were identical to the analytical solution within the tolerance specified to the solver. We also checked that solution (30) is correct.

4.2 Forward Problem: Implementation Issues and Results

After initial tests we used the method of warping with 3D curves for generation of more complex applicable surfaces. The tests were performed both by straightforward numerical integration of ODE's (29) and using the analytical solution for rectangular pathces (30). Both methods showed accurate and consistent results. To generate an example curve $\mathbf{R}(t)$ parametrized naturally, we specified another function $\tilde{\mathbf{R}}(\theta)$ where θ is an arbitrary parameter and then used transform

$$\mathbf{R}(t) = \tilde{\mathbf{R}}(\theta), \quad \frac{dt}{d\theta} = \left| \frac{d\tilde{\mathbf{R}}(\theta)}{d\theta} \right|, \quad (42)$$

which provides $|\dot{\mathbf{R}}| = 1$, and guarantees that t is the natural parameter. The function $\tilde{\mathbf{R}}(\theta)$ used in tests was

$$\tilde{\mathbf{R}}(\theta) = (P(\theta), N \cos \theta, N \sin \theta), \quad P(\theta) = a_1\theta + a_2\theta^2 + a_3\theta^3 + a_4\theta^4, \quad (43)$$

and some other than polynomial dependencies $P(\theta)$ were tested as well. One of the examples of image warping with a 3D curve is presented in Figure 5.

For this case the boundary curve were selected in the form (43), with parameters $N = 200$, $a_1 = 20$, $a_2 = 10$, $a_3 = 10$, $a_4 = -10$ and we used Eqs (31) and (34) to generate the 3D structure and characteristics. In this example the characteristics for this surface are not parallel, which is clearly seen from the graph in the upper right corner of Fig. 5. The image of the portrait of Ginevra dé Bencia by Leonardo da Vinci, was fit into a rectangle in the uv -plane and warped with the generated surface. Further its orthographic projection was produced using pixel-by-pixel mapping of the obtained transform from the (u, v) to the (x, y) . These pictures are also shown in Figure 5.

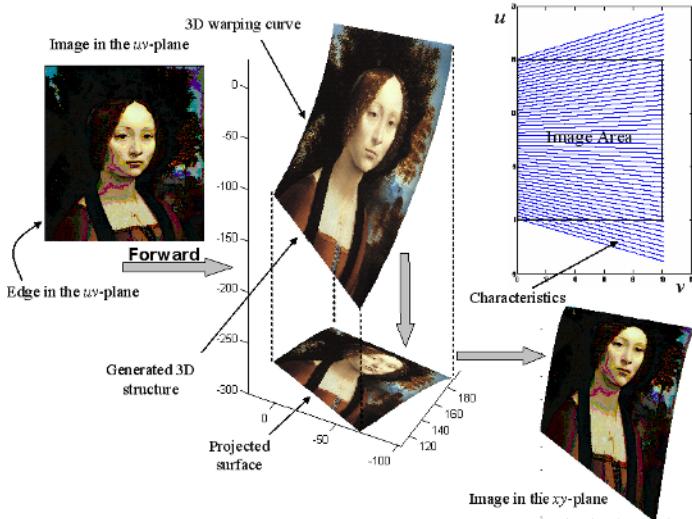


Fig. 5. ‘Forward’ problem: given a plane sheet of paper, and a smooth 3-D open curve in Cartesian XYZ space. Our goal is to bend the paper so that one edge conforms to the specified curve. Using the analytical integration of the differential geometric equations specifying applicability we are able to achieve this. We can also achieve the same result not only for the straight line edge, but for an arbitrary 2-D curve in the uv -plane. The picture shown are actual computations.

4.3 Inverse Problem: Implementation Issues and Results

To check the validity of the unwarping procedure, we ran the 2D unwarping problem with synthetic input data on the patch boundaries and corner correspondence points obtained by the warping procedure. The output of the solver providing h_u , h_v , \mathbf{r}_u , \mathbf{r}_v , and ρ as functions of t coincided with these functions obtained by the 3D curve warping program within the tolerance specified for the ODE solver. The unwarped pixel-by-pixel images are shown in Figure 6 as the end point of the unwarping process in the xy -plane. We ran the algorithm for small fonts. The original image has the same font size everywhere and with the forward algorithm we warp the image. The unwarped image has uniform font size everywhere, lines are parallel and right angles are preserved. The output is noisy at the top of the output image, since in the image this information was lost. We make the following remarks about the implementation of the inverse problem:

Global Parametrization: In the inverse problem, we march the ODE’s with respect to the bounding contours in uv -plane and xy -plane. Therefore, for simplicity and modularity, we use a global parameter η for bounding contours that runs from η in $[0,1]$ on the first boundary to $\eta = [3, 4]$ on the last. This parameterization gives us a simple and exact way of tracking the edges at the boundary contours and the correspondence between them.

ODE solver: To solve the ODE, we applied the Runge-Kutta 4th and 5th order in MATLAB, except for the last edge of the ODE, where the problem was computationally stiff. For this, we solved the ODE by Gear’s method [17].

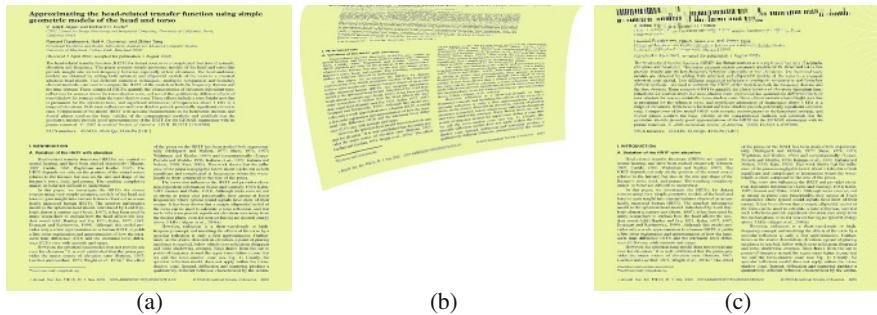


Fig. 6. Inverse Problem for small font: a) original image b) warped by the forward $\tilde{\mathbf{R}}(\theta) = (a\theta(b - \theta^3), N\cos\theta, N\sin\theta)$ where $a = 10, b = 2, N = 200$ c) unwarped by the inverse problem

Automatic Corner Detection by ODE solver: We need the corners in the image plane for the boundary of the patch to solve the inverse problem. As stated, the global natural parameterization of the curve in image plane, gives us an easy and reliable feature for corner detection. Basically, the corner is reached when s_2 and τ_2 (global parameters of Γ'_2 and Γ_2) are 1, 2 and 3, respectively.

5 Conclusion and Future Work

This paper presents, to our knowledge, the first occasion that differential geometry has been used quantitatively in the recovery of structure from images. A theory and method for warping and unwarping images for applicable surfaces based on patch boundary information and solution of nonlinear PDEs of differential geometry was developed. The method is fast, accurate and correspondence free (except for a few boundary points).

We see many useful applications of this method for virtual reality simulations, computer vision, and graphics; e.g. 3D reconstruction, animation, object classification, OCR, etc. While the purpose of this study was developing and testing of the method itself, ongoing work is related both to theoretical studies and to development of practical algorithms. This includes more detailed studies of the properties of the obtained equations, problems of camera calibration, boundary extraction, sensitivity analysis, efficient minimization procedures, and unwarping of images acquired by a camera, where our particular interest is in undoing the curl distortion of pages with printed text.

References

1. M. S. Brown and W. B. Seales. Document restoration using 3D shape: A general deskewing algorithm for arbitrarily warped documents In *ICCV 2001*, 2001
2. P. Clark and M. Mirmehdi. Estimating the orientation and recovery of text planes in a single image In *Proceedings of the British Machine Vision Conference*, 2001
3. D.A. Forsyth and J. Ponce. Computer Vision: A Modern Approach *Prentice Hall*, 2003
4. G.B. Whitham, *Linear and Nonlinear Waves*, New-York: Wiley, 1974

5. R. Hartley and A. Zissermann. Multiple View Geometry in Computer Vision In *Cambridge Press*, 2000
6. J. Garding. Surface orientation and curvature from differential texture distortion In *International Conference on Computer Vision, ICCV*, 1995
7. J.J. Koenderink. What Does the Occluding Contour Tell us About Solid Shape *Perception*, 13: 321-330, 1984
8. J.J. Koenderink, *Solid Shape*, MIT Press, 1990.
9. G. A. Korn and T.M. Korn. Mathematical Handbook for scientists and engineers In *Dover Publications, Inc.*, 1968
10. Y. L. Kergosien, H. Gotoda and T. L. Kunii. Bending and creasing virtual paper In *IEEE Computer graphics and applications*, Vol. 14, No. 1, pp 40-48, 1994
11. T. Kanungo, R. Haralick, and I. Phillips. Nonlinear Local and Global Document Degradation Models In *Int'l. J. of Imaging Systems and Tech.*, Vol. 5, No. 4, pp 220-230, 1994
12. D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes In *IEEE Computer Vision and Pattern Recognition Conference*, pp 482-488, 1998
13. M. A. Penna. Non-rigid Motion Analysis: Isometric Motion In *CVGIP: Image Understanding*, Vol. 56, No. 3, pp 366-380, 1992.
14. M. Do Cormo. Differential Geometry of Curves and Surfaces. *Prentice Hall*, 1976
15. M. Pilu. Extraction of illusory linear clues in perspectively skewed documents In *IEEE Computer Vision and Pattern Recognition Conference*, 2001
16. M. Pilu. Undoing Page Curl Distortion Using Applicable Surfaces In *Proc. IEEE Conf Computer Vision Pattern Recognition*, 2001
17. W.H. Press, S.A. Teukolsky, W.T. Vetterling and B. P. Flannery. *Numerical Recipes in C*, Cambridge University Press, 1993
18. R.I. Hartley. Theory and practice of projection rectification In *International Journal of Computer Vision*, Vol.2, No. 35, pp 1-16, 1999
19. Y. You, J. Lee and Ch. Chen. Determining location and orientation of a labeled cylinder using point pair estimation algorithm In *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 8, No. 1, pp 351-371, 1994

Joint Bayes Filter: A Hybrid Tracker for Non-rigid Hand Motion Recognition

Huang Fei and Ian Reid

Department of Engineering Science,
University of Oxford, Parks Road, Oxford, OX1 3DP, UK

Abstract. In sign-language or gesture recognition, articulated hand motion tracking is usually a prerequisite to behaviour understanding. However the difficulties such as non-rigidity of the hand, complex background scenes, and occlusion etc make tracking a challenging task. In this paper we present a hybrid HMM/Particle filter tracker for simultaneously tracking and recognition of non-rigid hand motion. By utilising separate image cues, we decompose complex motion into two independent (non-rigid/rigid) components. A generative model is used to explore the intrinsic patterns of the hand articulation. Non-linear dynamics of the articulation such as fast appearance deformation can therefore be tracked without resorting to a complex kinematic model. The rigid motion component is approximated as the motion of a planar region, where a standard particle filter method suffice. The novel contribution of the paper is that we unify the independent treatments of non-rigid motion and rigid motion into a robust Bayesian framework. The efficacy of this method is demonstrated by performing successful tracking in the presence of significant occlusion clutter.

1 Introduction

Many computer vision applications such as surveillance, sports, and human-computer interfacing require robust estimation and understanding object motion. Tracking of rigid and non-rigid objects such as vehicles [2] and humans [1] has been under extensive investigation in recent years. In this paper, we address the existing problems of single-view tracking and recognition of articulated hand motion in complete occlusion scenes. In these situations, simultaneous estimation and recognition of articulated hand poses can be challenging but is crucial for gesture recognition. By utilising separate image cues, we decompose complex motion into two independent (non-rigid/rigid) components. Rigid motion is approximated as the motion of a planar region and approached using a *Particle filter* while non-rigid dynamical motion is analysed by a *Hidden Markov Model (HMM) filter*. Due to its generative learning ability, hand articulation is correctly estimated even under significant occlusions.

A considerable body of work exists in hand tracking and gesture recognition. All tracking methods have some associated tracker representations, either kinematic model-based [8], [11], [5] or appearance model-based [12]. Kinematic

model-based methods construct a geometrical model before tracking. Although it can provide more information about hand configurations than 2D appearance, tracking is usually made possible with a careful initialisation and tedious model fitting/searching process, and could fail to maintain tracking where there is fast appearance deformation and/or severe occlusions; in gesture recognition, significant change point in appearance deformation is usually important for semantics interpretation. PCA appearance models [12] have the advantage of the ability to generate a new appearance using a small training set, but linear correlations impose a limit to its applications. Complex scenes and occlusion clutter pose serious distractions to all these representations. In recent years, exemplars [16] have become a popular choice for tracker representations because object models can be extracted from the original data, and the non-linear nature of articulations can be conveniently represented by a sequence of examples which exhibits first-order Markov dependence.

In addition to representation, motion estimation algorithm also contributes significantly to a tracker's performance. Besides the successful application of Particle filter[4], Dynamic Bayesian Network (DBN) [10] in visual tracking, the Hidden Markov Model (HMM) considered in this paper is another statistical time-series modelling tool [15]. Since it was firstly introduced for sign-language analysis and gesture recognition [3], there have been some research work in human motion classification using HMMs. Bregler proposed a probabilistic compositional framework [22] to classify dynamic human representation (coherence blobs), Brand [19] makes a further study in this direction by identifying kinematic structure evolution as a controlling dynamic process in 3D human motion synthesis given a set of motion capture data. Despite these efforts, the powerfulness of *dynamic* Markov model as a tracker to analyse the non-rigid motion has not been recognised until recent years [16]. In the *Metric Mixture* tracker proposed by Toyama and Blake [16], exemplars are assumed aligned in the center of probabilistic mixtures. The object tracking problem then transforms to the filtering and recognition of the representative examples in the spatial-temporal domain.

In [16], two dynamic processes (global motion and shape changes) share the same joint observation density provided by the chamfer distance. This leads to an attractive homogeneity of description and implementation. However it necessitates the use of a large particle set, which must *simultaneously* represent hypotheses of both shape and position. In many applications these processes can (or even should) be decoupled, potentially leading to a more economical use of particles, and therefore greater efficiency and reliability.

We propose such a decoupling for the analysis of complex hand motion. In this application, the hand motion is separated into two components assumed independent: cyclic shape variations and hand region motion. Unlike [16], each has its own dynamic process and observation density. The former is modelled by a probabilistic discrete-state system (using a Hidden Markov Model) while the latter is achieved via a particle-based region tracker. Each maintains a full pdf of its respective component of the motion, and they interact via an importance sampling function. We term this the **Joint Bayes filter (JBF)**.



Fig. 1. A video sequence of articulated hand motion.

In JBF, *monte-carlo* approximation provides an optimal localization of the hand region regardless of its articulation and clutter distraction, which reduces the distraction to the non-rigid shape inference. On the other hand, the HMM filter's output model the dynamics of the particle set and provides the importance weights of the particles, thus improves the accuracy and efficiency of the region tracking. The overall benefit of our approach is clear: Tracking and recognition of hand articulations can be performed simultaneously; due to the independence of two observation processes, JBF tracker can withstand significant occlusion distractions and perform well in cluttered scenes.

2 The Problem

Figure 1 shows consecutive example frames from a video sequence of articulated hand motion. Changing appearances between successive frames can be significant, and thus rule out the standard assumption of constant intensity underlying optical flow based tracker [12]. Although an edge-based tracker [18] [11] [21] may perform well in normal situations, the strong occlusion clutter introduced later in the paper will damage a tracker without prior dynamics [12], or with only weak prior dynamics [18]. In this situation, even dense sampling will not help tracking articulations because of the lack of a proper *dynamical model of hand articulations*. These difficulties call for compact hand representations and stronger dynamical model based on the representations.

To deal with fast appearance deformation, strong occlusion clutter and complex scene, combination of both shape and colour as tracker representation [18], [23] becomes a natural choice. Although a global shape region provides a more reliable measure of the object identity than sparse boundary edges, in the presence of heavy occlusion clutter, a strong dynamic model of hand shape variation is still needed to infer what is happening behind the scene. We believe that such shape dynamics learned from regions are more reliable than their edge-based counterparts. Colour is another useful cue, because it can not only provide task-specific object representation (for example, skin colour can segment the hand from the shadows and form a silhouette sequence), but also provide a good measure of the moving region when we need to approximate 'rigid region' motion. In this paper, we exploit the fact that colour-histogram of the region of interest is invariant to geometrical distortions provided that it is correctly tracked. This rigid colour appearance has been studied in [6], [14]. A multiple hypothesis based particle filter together with colour representations [14] has been demonstrated as a good basis for region-based tracking.

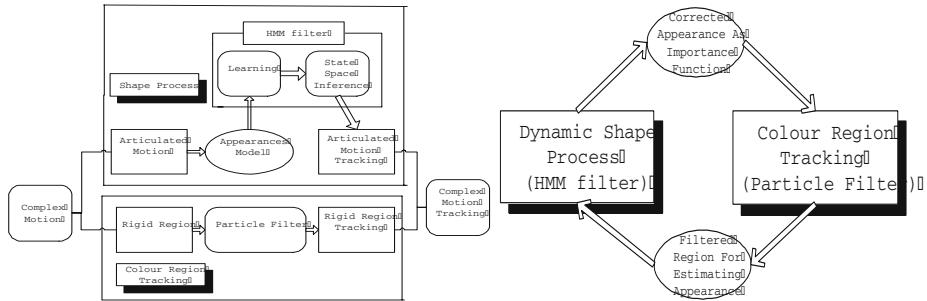


Fig. 2. (a)The flow chart of new tracking system; (b)The relationship between the two independent components. In each video frame, the Particle filter first locate the hand, and then the HMM filter infers about its' appearance. The appearance is used to update the Particle filter for the next video frame.

Although a colour and shape based tracker was proposed in [18] [23], our approach differs significantly from theirs. No dynamic models of hand shape variations are learned in [18] [23], yet this dynamics together with relatively robust features are crucial for tracking non-rigid motion under severe occlusions. A compact global silhouette representation as image moments [19] can avoid the tedious need to find the hand shape from multiple observations at local regions, and thus save computational resources for the colour-histogram computation and non-rigid shape tracking. In our Joint Bayes Filter (JBF) method, a colour-based particle filter provides a robust estimation of non-rigid object translation and localizes the most likely hand location for the HMM filter. In turn, the shape output from the HMM filter provides the importance weighting for the particle set before the resampling stage and the particle set updating in the prediction stage. This combination distinguishes our method from others. For illustrative purposes, we introduce the overall tracking system in Figure 2. The relationship between the two independent Bayesian filters, the HMM filter and the Particle filter, is also summarized.

3 Discrete Shape Tracking

In this section, we discuss the first component (the discrete-state shape tracking) of our system. Like a speech signal, we assume the articulation of hand motion as a time sequential process and can provide time-scale invariance in recognition and tracking. In most situations non-rigid motion periodically causes appearance changes. The underlying motion patterns of the articulations are often intractable, while the appearance changes often observe statistical constraints. Though a silhouette of the hand is one of the weak cues and barely preserve 3D structure of the hand, it could provide a reasonable amount of information about the articulated shape changes without resorting to a complex hand model. Image moments ($X = \{m_0, \dots, m_n\}$ where m_i is the i^{th} moment of the shape)

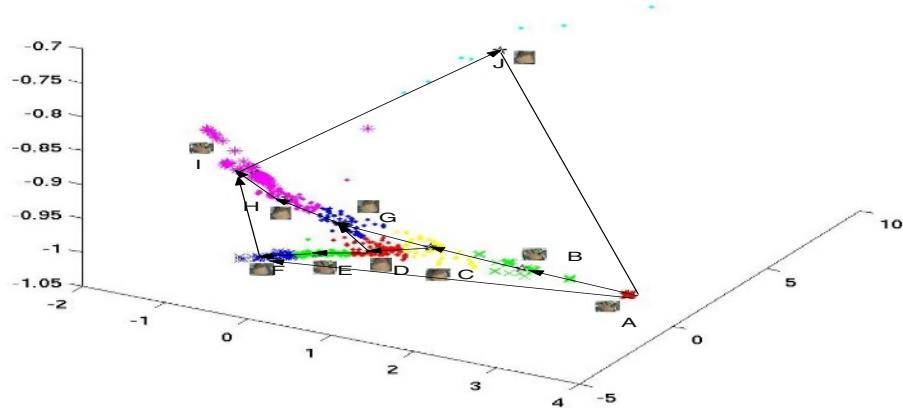


Fig. 3. Articulated hand motion embedded into a 3D metric space using Local Linear Embedding (LLE) [9] algorithm.

of the silhouettes are computed to estimate the shape class. Although a single silhouette and its image moments cannot reveal the truth about the underlying motion patterns, when we accumulate this weak evidence given sufficient amount of training data, a useful dynamic appearance manifold embedded in the training data can be discovered (Figure 3).

In line with recent advances in manifold learning [9], we embed our image moments sequence in a metric space. Figure 3 shows the distributions of articulated hand appearances. In the vector-quantized 3d metric space, not only are the spatial-temporal continuity of the non-rigid hand motion (i.e. the hand shape changes) well preserved, but also smooth trajectories (which approximate the linear Markov chains, confirming the Markov assumption underlying HMM) in the feature space are identifiable. This preliminary examination validates in a certain degree that the image features (silhouette moments in this case) is sufficient to account for non-rigid motion.

3.1 Learning and Tracking

Having assumed that non-rigid motion causes a dynamic appearance manifold, and verified that a sequence of image moments can actually replicate such dynamics (Figure 3), we concentrate on the essential learning and inference aspects for our tracking objective. Similar as the statistical learning in HMMs, dynamic motion data are aligned to discrete states and motion dynamics are approximated from the states. During tracking, in order to estimate what is going to be next most likely appearance correctly, the best underlying state sequence has to be decoded from current observation and a prior dynamic appearance model.

A classical VQ algorithm [13] is used in the learning stage to group the hand appearance data into clusters. Although complex gaussian mixtures are usually used in approximating motion data, in this paper simple L_2 distance measure is

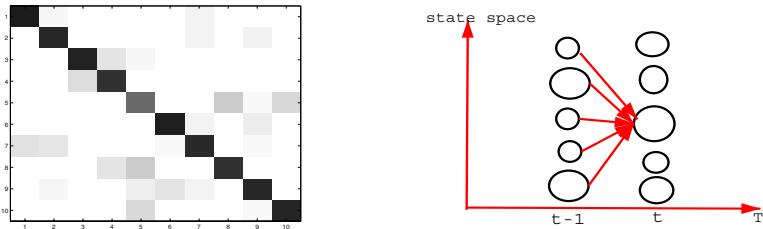


Fig. 4. (a) Dynamic model provides strong motion prior; (b) The size of the circles represent the distribution of shape identities from the observation process, together with the dynamic model (weighted along the trajectories), determine the MAP estimation results in each frame.

used without strong parametric assumption. Thus we not only obtain the tracker representations, but also ‘embed’ articulated human motion into discrete states. Having obtained the codebook, the essential aspect of the HMM tracker: *spatial-temporal filtering using shape dynamics*, is straightforward. HMM provides such strong motion dynamics $P(X_t|X_{t-1})$ (Figure 4),

$$P(X_t|X_{t-1}) = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \quad (1)$$

where $\xi_{ij}(t)$ denotes the probability of being in state i at time t and at state j at time $t+1$ given the model and the observation, $\gamma_i(t)$ be defined as the probability of being in state i at time t , given the entire observation sequence and the model. This can be related to $\xi_{ij}(t)$ by summing as $\gamma_i(t) = \sum_{j=1}^N \xi_{ij}(t)$, N is the number of latent states.

We argue that this predictive model $P(X_t|X_{t-1})$ is typically stronger than those of *Kalman filter* or *Particle filter* tracker (Figure 7). In those trackers, the dynamic motion is usually built through a rather *ad-hoc* prediction (usually an AR model) of local feature measurements. In *HMM*, the dynamical model $P(X_t|X_{t-1})$ is built through the statistics of latent states evolution. After the alignment of the human appearances into discrete samples, latent states have been associated with meaningful example appearances (either deterministic or probabilistic). Estimating the expected number of transitions from state s_i to state s_j and the expected number of transitions from state s_i in the training set determines the likelihood that one appearance evolves to another appearance sequentially, thus approximate the prior probability distribution for non-rigid motion.

On the other hand, object tracking can be viewed as feature matching or object registration in the time domain, it shares similarities with object recognition. In discrete appearance tracking, we are interested in more than simply recognizing static patterns at a particular time t , dynamic appearance patterns are more important for tracking. We define X_t as the discrete shape tracker state (the associated exemplars) at time t , and Z_t represents image observations (image moments of the silhouettes in this case) at time t . We emphasize that the

trajectories between different states A, B, C etc observe Markov assumptions. The visual tracking problem can be interpreted as making inference about the ‘random walk’ on the appearance manifold, however such ‘randomness’ must observe Markov independence property. For example, a first order Markov model observe $P(X_t|X_{1:t-1}, Z_{1:t-1}) = P(X_t|X_{t-1})$, its geometrical interpretation is, for instance, when object appearances evolve from $A \rightarrow B \rightarrow C \rightarrow G \rightarrow H \rightarrow I$ on the manifold. At state I , the example appearance is more closely related to state H than the previous appearances. This assumption is plausible because state H is closest to state I , agreeing with the general energy (distance) minimal principle as used in *Kass Snakes* tracker [21].

The *Viterbi* algorithm [20] is adapted for decoding the best underlying state sequence as well as tracking non-rigid hand motion. In order to find the single best state sequence, $Q = (q_1, q_2, \dots, q_t)$ (also known as the motion trajectory such as $A \rightleftharpoons B \rightleftharpoons C \rightleftharpoons G \rightleftharpoons H$), for the given observation $O = (o_1, o_2, \dots, o_t)$ (the measurements such as $\hat{A}, \hat{B}, \hat{C} \dots$ etc), we first define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1 q_2 \dots q_{t-1}, q_t = i, o_1 o_2 \dots o_t | \lambda] \quad (2)$$

where λ refers to a particular HMMs for hand motion analysis. $\delta_t(i)$ is the best score (highest probability) along a single path, at time t , which accounts for the first t observations and ends in state i . Since during visual tracking, making the right predictions at each time instants is the major objective, we come to the *Bayesian* tracking formula.

$$P(X_t|Z_{1:t}) = \max_{1 \leq i \leq N} [\max_{1 \leq i \leq N} [\delta_{t-1}(X) \cdot P(X|X_{t-1})] \cdot P(Z_t|X_t)] \quad (3)$$

4 Colour-Region Tracking

Now we briefly introduce the second component (colour-region tracking) of our system. Tracking non-rigid hand motion cannot be successful without a robust global motion estimator, and this is an equally important issue when occlusion occurs. A particle filter is now the standard tool to handle such multimodal nature distractions. Colour-histogram is a relative robust and efficient region descriptor invariant to non-rigid object translation, rotation and occlusion. In this paper, only when the hand region is correctly localized, can colour segmentation provide an accurate silhouette input to the HMM filter.

Traditional colour-based particle filter tracker has some drawbacks. First it lacks a sophisticated mechanism for updating the region’s scale changes. This difficulty can cause troubles for deterministic methods [6]. In [6] and [14], no clear solutions for updating the scales are given. A recent work attacks this problem by utilizing scale-space concepts [7]. In fact, the adaptive scale corresponds to the non-rigid shape changes. In our JBF framework, we explicitly model the dynamics of the particle set as a first-order AR process, updated by output from the HMM filter. A second problem with the traditional particle filter is that *factored sampling* often generate many lower-weighted samples which have

little contribution to the posterior density estimation. Accuracy and efficiency are sacrificed accordingly. However, the HMM filter in the JBF tracker provides an additional sensor which can reweigh the particles and form an ‘important’ region for the particle filter.

5 Joint Bayes Filter

The essential aspect of Joint Bayes Filter is that at every video frame, each of the process (either shape changes or global hand motion) maintains an independent motion prior for the shape/position of the hand. The likelihood of each process is evaluated separately (nearest neighbour classification in the shape feature space, histogramming similarity measure in the colour space), but the posterior of each process is “resampled” to maintain accuracy and efficiency (the HMM filter provides importance sampling for the Particle filter, Particle filter determines the most likely hand region¹), therefore, complementary Bayesian beliefs are reinforced and propagated through the Markov chain.

For the sake of clarity, we summarize the JBF algorithm in detail. In the HMM filter, let X_t represent the shape tracker state (associated with exemplars), and Z_t denote the image observations (image moments of the silhouette in this case) at time t . $d(X_t, Z_t)$ refers to the distance measure in feature space. The state vector of the Particle filter is defined as $x_t = (x, y, s_x, s_y)$, where x, y, s_x, s_y refer to the rectangle location $L(x, y)$ in the image plane and scales along x, y coordinates. $R(x_t)$ is the candidate region thus defined, M is the number of particles used. $b_t(u) \in \{1, \dots, N\}$ is the bin index associated with the colour vector $y_t(u)$ at pixel location u in frame t . Assume we have a reference colour histogram: $q^* = q^*(n)_{n=1, \dots, N}$ obtained at initial frame. $q_t(x_t)$ denotes the current observation of the colour histogram.

$$q_t(x_t) = C \sum_{L \in R(x_t)} \omega(|u - L|) \delta[b_t(L) - n]. \quad (4)$$

where C is a normalization constant ensuring $\sum_{n=1}^N q_t(X_t) = 1$, ω is a weighting function. $D([q^*, q_t(x_t)])$ represents the Bhattacharyya distance.

The $g_t(X_t)$ used is similar to the one proposed in *ICondensation* [18], $g_t(X_t) \sim \exp(-\lambda(C(S_t) + \Delta x_t))$ where $C(S_t)$ denotes the centroid of the shape, and Δx_t is the offset between the centroid of the shape and the colour region. In the following, $A_H(x_t)$ denotes the most likely hand region, which is a rectangle area. $A_S(X_t)$ refers to the shape tracker output from the HMM filter.

¹ Although it seems plausible to directly update the posterior of the HMM filter from the posterior of the Particle filter (i.e. estimate the mean shape in the feature space from the sample shape output from the whole distribution of colour regions), we believe such treatment has no significant improvement and would like to investigate this topic further in the future.

Joint Bayes Filter Algorithm

1. Initialization.

Particle Filter: Select the hand region, obtain the reference colour-histogram q^* . For $i = 1, 2, \dots, M$, select the initial particle set $x_0^{(i)}$.

HMM Filter: Obtain $A_H(x_0)$ from the tracker initialization. Perform colour segmentation in $A_H(x_0)$ to obtain the silhouette.

2. Prediction.

Particle Filter: For $i = 1, 2, \dots, M$, draw new sample set $\tilde{x}_t^{(i)} \sim p(x_t | x_{t-1}^{(i)})$, here the dynamics process is a first order AR model. Calculate the colour-histogram distribution $q_t(\tilde{x}_t)$. Evaluate the importance weights

$$\tilde{\omega}_t^{(i)} = \frac{p(x_t | x_{t-1}^{(i)})}{g_t(X_t)} p(z_t | \tilde{x}_t^{(i)}), \text{ where } p(z_t | \tilde{x}_t^{(i)}) \sim \exp(-\lambda D^2[q^*, q_t(x_t)]),$$

and normalize the importance weights.

HMM Filter: Generate the new prior $P(X_t | Z_{1:t-1})$ by propagating $P(X_{t-1} | Z_{t-1})$ through the markov chain.

3. Update.

Particle Filter: Resample with replacement N particles $(x_t^{(i)}; i = 1, 2, \dots, N)$ from the set $(\tilde{x}_t^{(i)}; i = 1, \dots, N)$ according to the importance weights. Output the $A_H(x_t)$ from the particle filter.

HMM filter: Obtain the $A_H(x_t)$ from the Particle filter, perform colour segmentation, get the observation density $P(Z_t | X_t) \sim \exp(-\lambda d(X_t, Z_t))$. Combine with the prior $P(X_{t-1} | Z_{t-1})$ to estimate $P(X_t | Z_{1:t})$ which is the most likely appearance at time t .

6 Experiment

Several experiments are designed to examine the performance of the JBF tracker.

1. **Tracking dynamic appearances using JBF.** We obtain a long video sequence of cyclic hand motion. 70% of the data is used for training the dynamic appearance model $P(X_t | X_{t-1})$ and selecting the exemplar set, the rest for tracking. 300 particles are used to approximate the distribution of the candidate regions, each is associated with the colour-histogram density in YUV colour-space (8 bins for each channel). Near real-time performance has been achieved for the overall tracking system. The result is shown in Figure 5. Small non-rigid appearance deformations and varying changing speed between successive frames are well captured. In fact, the gait of the articulated hand motion is encoded in the strong appearance dynamics which is built in the learning stage. We also notice that even using the weak cue of image moments alone, tracking non-rigid hand poses in the JBF framework can achieve rather good performance.
2. **Coping with occlusion.** In Figure 6, we demonstrate that the region tracker provides relatively robust object detection/localization, and therefore reduces the distractions to the HMM filter. Of most interest is whether the performance of the HMM filter tracker will degenerate under several frames

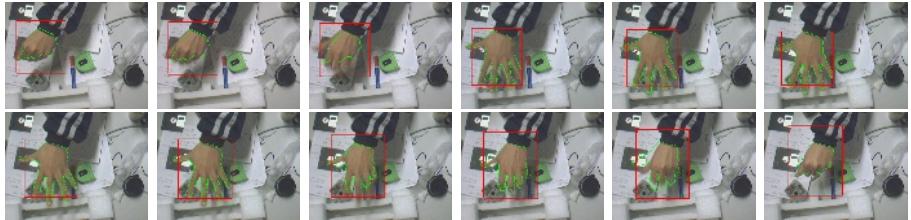


Fig. 5. Tracking results of the JBF tracker, the Particle filter determines the most likely hand region (the red rectangle), the HMM filter produce the most likely hand shapes (the green contours of the example shapes are drawn to show results).

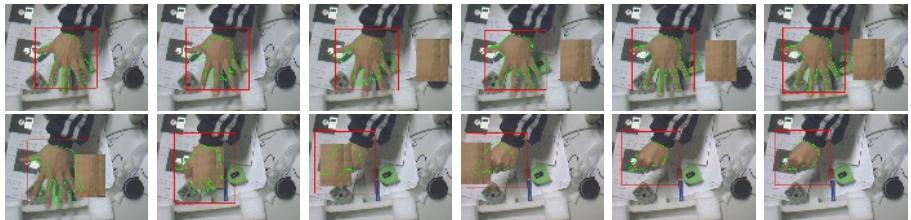


Fig. 6. The Particle filter in JBF reduces the distractions to the HMM filter.

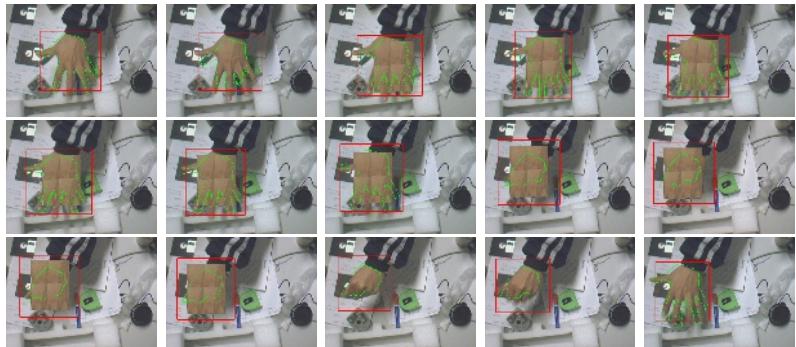


Fig. 7. The HMM filter in JBF withstands several frames of occlusion clutter.

of significant occlusion. In experiment (Figure 7), we clutter the hand regions with skin colour patch for several frames, and observe that the tracker is not only able to recover sufficient information of the hand shape, but also can correctly predict the significant change point in hand articulation. This suggests that our HMM component in JBF framework is relatively robust to significant occlusion clutter.

Here we summarize the mechanism of the HMM filter to handle occlusion clutter as demonstrated in (Figure 7) : $P(X_t|X_{t-1})$ represents the strong appearance dynamics of hand motion learned, $P(Z_t|X_t)$ represents the observation density.

In discrete appearance tracking, the most likely shape appearance estimation is given by $P(X_t|Z_t) \sim [P(X_{t-1}|Z_{t-1}) \cdot P(X_t|X_{t-1})] \cdot P(Z_t|X_t)$.

Suppose up to frame t , there are no occlusion or unreliable observations. From frame $t+1$, a significant occlusion is introduced into the video sequence. Then the observation density $P(Z_{t+1}|X_{t+1})$ contributes little to the shape appearance tracking with the first two components corresponding to the dynamic prior being most influential. A strong dynamic appearance model $P(X_t|X_{t-1})$ obtained during the learning stage, and a correct initial estimate $P(X_0|Z_0)$ in the tracking stage, are two important factors which enable the HMM filter tracker to give an optimal estimate even under harsh conditions.

7 Conclusions

This paper presents an unifying framework for non-rigid motion tracking and understanding. The contributions of our approach are as follows:

1. Explicitly separate the articulated hand motion into two independent observation processes: non-rigid motion and rigid region motion. Different dynamic models in JBF (dynamic appearance model in the HMM filter modelling the shape changes, auto-regressive process in the Particle filter updating the particle set) are complementary for articulated hand motion tracking.
2. Demonstrate the probabilistic inference mechanism of the HMM filter in visual tracking is Bayesian (MAP). In contrast to the multiple hypothesis in particle filter, we show that state-based inference is also robust to occlusion clutter and unreliable measurements. Both methods are fully Bayesian and therefore this combination (JBF filter) gives robust tracking results in real-world applications.
3. In contrast to the previous work, we associate shape descriptors with the HMM filter, colour-histogram appearance model with the Particle filter, independent target representations are closely related to the motion estimation objective(non-rigid/rigid motion), in a hand tracking and recognition application.

References

1. Aggarwal. J, Cai.Q: Human Motion Analysis: A Review. Computer Vision and Image Understanding (1999)
2. Koller. D, Weber. J and Malik. J: Robust Multiple Car Tracking with Occlusion Reasoning. Proc. European Conference on Computer Vision (1994)
3. Starner.T, Pentland.A: Visual Recognition of American Sign Language Using Hidden Markov Model. Proc. International Workshop on Automatic Face and Gesture Recognition (1995)
4. Isard. M, Blake. A: Active Contour. Springer-Verlag (1998)
5. Morris.D and Rehg.J: Singularity Analysis for Articulated Object Tracking. Proc. Computer Vision and Pattern Recognition (1998)

6. Comaniciu.D, Ramesh.V and Meer.P: Real-Time Tracking of Non-Rigid Objects using Mean Shift. Proc. Computer Vision and Pattern Recognition (2000)
7. Collins.R: Mean-shift Blob Tracking through Scale Space. Proc. Computer Vision and Pattern Recognition (2003)
8. Rehg.J and Kanade.T: Model-Based Tracking of Self-Occluding Articulated Objects. Proc. International Conference on Computer Vision (1995)
9. Roweis.S and Saul.L: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science (2000)
10. Palovic.V, Rehg.J, Cham.T, Murphy.K: A Dynamic Bayesian Network Approach to Figure Tracking Using Learned Dynamic Models. Proc. International Conference on Computer Vision (1999)
11. Stenger.B, Thayananthan.A, Torr.P, and Cipolla.R: Filtering Using a Tree-Based Estimator. Proc. International Conference on Computer Vision (2003)
12. Black.M ,Jepson.A: Eigen tracking: Robust matching and tracking of an articulated objects using a view based representation. Proc. European Conference on Computer Vision (1996)
13. Linde.A , Gray.R: An algorithm for vector quantization design. IEEE.Trans. on Communications (1980)
14. Pérez.P, Hue.C, Vermaak.J and Gangnet.M: Color-based probabilistic tracking. Proc. European Conference on Computer Vision (2002)
15. Rabiner.R: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc.IEEE (1989)
16. Toyama.K and Blake.A: Probabilistic Tracking with Exemplars in a Metric Space. Proc. International Conference on Computer Vision (2001)
17. Sidenbladh.H, Black.M: Learning Image Statistics for Bayesian Tracking Proc. International Conference on Computer Vision (2001)
18. Isard.M and Blake.A: ICondensation: Unifying low-level and high-level tracking in a stochastic framework Proc. European Conference on Computer Vision (1998)
19. Brand.M: Shadow Puppetry. Proc. International Conference on Computer Vision (1999)
20. Viterbi.A: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE. Transaction on Information Theory (1967)
21. Kass.M, Witkin.A, Terzopoulos.D: Snakes: Active contour models. International Journal of Computer Vision (1987)
22. Bregler.C:1997: Learning and Recognizing Human Dynamics in Video Sequences. Proc. Computer Vision and Pattern Recognition (1997)
23. Birchfield.S: Elliptical Head Tracking using Intensity Gradients and Colour Histogram. Proc. Computer Vision and Pattern Recognition (1998)
24. Gavrila.D, Philomin.V: Real-time Object Detection for ‘smart’ Vehicles International Conference on Computer Vision (1999)

Iso-disparity Surfaces for General Stereo Configurations

Marc Pollefeys and Sudipta Sinha

Department of Computer Science
University of North Carolina
Chapel Hill, N.C. 27599-3175
{marc,ssinha}@cs.unc.edu

Abstract. This paper discusses the iso-disparity surfaces for general stereo configurations. These are the surfaces that are observed at the same resolution along the epipolar lines in both images of a stereo pair. For stereo algorithms that include smoothness terms either implicitly through area-based correlation or explicitly by using penalty terms for neighboring pixels with dissimilar disparities these surfaces also represent the implicit hypothesis made during stereo matching. Although the shape of these surfaces is well known for the standard stereo case (i.e. fronto-parallel planes), surprisingly enough for two cameras in a general configuration to our knowledge their shape has not been studied. This is, however, very important since it represents the discretisation of stereo sampling in 3D space and represents absolute bounds on performance independent of later resampling. We prove that the intersections of these surfaces with an epipolar plane consists of a family of conics with three fixed points. There is an interesting relation to the human horopter and we show that for stereo the retinas act as if they were flat. Further we discuss the relevance of iso-disparity surfaces to image-pair rectification and active vision. In experiments we show how one can configure an active stereo head to align iso-disparity surfaces to scene structures of interest such as a vertical wall, allowing better and faster stereo results.

1 Introduction

In stereo matching pixels of one images are compared with pixels of another image to identify corresponding pixels. The accuracy at which this can be done is limited by the resolution of the images. Since the matching ambiguity is limited to the epipolar lines, it is the resolution along the epipolar lines that is relevant. Therefore, we argue that iso-disparity surfaces (where disparities are defined along the epipolar line) characterize the uncertainty and discretization in stereo reconstruction. While the geometry of these surfaces is very well known and understood in the standard stereo case (i.e. fronto-parallel planes located at distances inversely proportional to the disparity), only very little is known for general stereo configurations. However, as we will see later, the shape of these curves –and therefore what is achievable by stereo– depends dramatically on the geometric configuration (and the internal cameras settings). It is also important to be aware that image pair rectification can only affect negatively (or at best keep unchanged) the intrinsic uncertainty of the stereo reconstruction for a certain configuration.

In addition, many stereo algorithms make assumptions about the scene that are related to the iso-disparity surfaces. Although many different approaches have been proposed

to compute dense stereo correspondences (we refer the readers to [22] for a recent review), most of these algorithms share common properties. Many algorithms evaluate potential matches by comparing pixels within a support region located around the points of interest. This pixel-by-pixel comparison of the region makes an implicit assumption about the 3D geometry of the observed surface. In fact, in this case the stereo algorithm “sweeps” over the iso-disparity surfaces. These correspond to the shape hypothesis being considered by the stereo algorithm.

Besides this, many stereo algorithm perform an optimization over the epipolar line or over the whole image where not only the matching cost is minimized, but also the surface smoothness is taken into account. This allows to reduce problems with ambiguous matches. Often, a cost term of the form $\|\nabla d\|^2$ is used where d is the disparity (along the epipolar line) and ∇ represents the gradient operator. A similar term can also be used for computing dense disparity maps using optical flow [3]. This results in a bias towards 3D surfaces for which $\|\nabla d\| = 0$. In other words, we can conclude that in addition to characterize the uncertainty iso-disparity surfaces represent the implicit assumptions made by many stereo algorithms.

In Section 2 and 3 we will discuss the geometry of the iso-disparity surfaces. In Section 4 related work is discussed. In Section 5 the relevance to the human visual system is discussed and in Section 6 the impact on image-pair rectification is discussed. Section 7 discusses the application to active vision and the paper is concluded in Section 8.

2 The Standard Stereo Case

Before analysing general stereo configurations, we will review the standard stereo case. The standard stereo configuration consists of two identical cameras with a relative displacement being a pure translation along the cameras’ x-axes. In this configuration the same scanlines from both images are corresponding epipolar lines. Therefore the stereo search can be limited to corresponding horizontal scanlines.

In this case the implicit stereo surfaces take on a particularly simple form. For a 3D point (X, Y, Z) and two cameras in the standard stereo configuration with focal length f and baseline b (and with the image plane aligned with the XY-plane), the observed disparity is

$$d = -\frac{fb}{Z} . \quad (1)$$

Therefore in this case the implicit stereo surfaces are fronto-parallel planes. This is a well-known result. An example is shown in Figure 1. The synthetic cameras are 20 cm apart, have parallel optical axis and a virtual image size and focal length of 1000 pixels (roughly equivalent to 35mm lense or 53 degrees field-of-view). To avoid clutter only one in ten iso-disparity curves is plotted and a dot is placed for one in ten pixels on those curves. The same camera and baseline are also used further in the paper.

3 The General Case

Because the ambiguity caused by an unknown depth is limited to image displacements along the epipolar lines we define disparities in those terms. First we define a coordinate

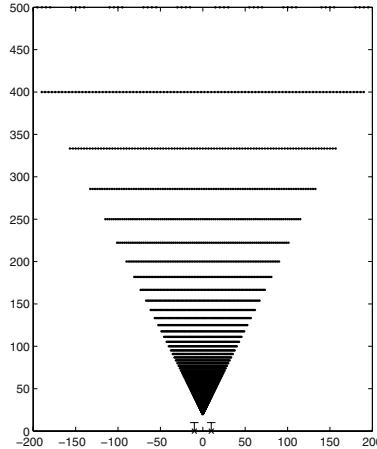


Fig. 1. Implicit stereo surfaces for the standard setup.

λ on the epipolar line where $\lambda(\mathbf{m}) = |\mathbf{m} - \mathbf{e}| - l_0$ where \mathbf{e} is the epipole and with $l_0 = \min |\mathbf{m} - \mathbf{e}|$ over all image points \mathbf{m} . Note that λ is bound to $[0, \sqrt{w^2 + h^2}]$ (with w and h being the image width and height respectively). Now the disparity can be defined as $d = \lambda'(\mathbf{m}') - \xi\lambda(\mathbf{m})$ with $\xi \in \{-1, 1\}$ a term that takes into account a possible different orientation of the epipolar lines (see [13] for more details on this issue)¹. A similar disparity measure was for example proposed in [3]². Note that this is different from the typical definition where disparities would be defined directly in difference of image coordinates, leading to both horizontal and vertical disparities, e.g. [23]. In the case where epipolar lines are horizontal both measures coincide.

Given the nature of the problem it is most relevant to start with the analysis of the shape of the iso-disparity curves within epipolar planes. Let us call such a curve ϕ . Let us also define the point M_∞ as the point that is imaged at infinity in both images, i.e. the point that is on the intersection of both planes parallel with the image planes and passing through the centers of projection (and the considered epipolar plane). The following theorem can be proven:

Theorem 1. *The iso-disparity curves ϕ are conics that pass through both centers of projection and the point M_∞ .*

Proof: Consider two centers of projection C_1 and C_2 and the intersections I_1 and I_2 of an epipolar plane with the two image planes (see Figure 2). Take four image points $\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_i, \mathbf{m}_j$ with coordinates $0, 1, i, j$ along the line and the point at infinity \mathbf{m}_∞ on the epipolar line I_1 and similarly $\mathbf{m}'_0, \mathbf{m}'_1, \mathbf{m}'_i, \mathbf{m}'_j$ and \mathbf{m}'_∞ on the corresponding epipolar line

¹ ξ should be chosen so that the visible part of the iso-disparity surface is observed from the same side by both cameras, e.g. $\xi = -1$ for verging cameras where the epipoles are on different sides of the image.

² Note that the definition proposed in [3] has the undesirable implicitly assumption that corresponding epipolar lines have similar orientations.

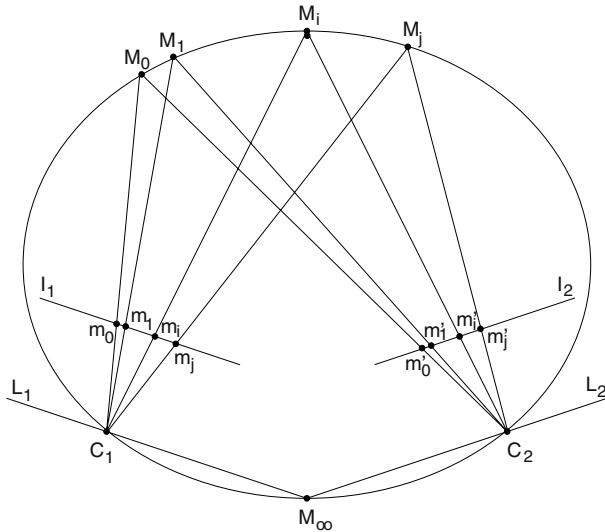


Fig. 2. Illustration of theorem

I_2 . Let us name the corresponding 3D points M_0, M_1, M_i, M_j and M_∞ respectively. Observe that the cross-ratio

$$\{M_0C_1, M_1C_1; M_iC_1, M_\infty C_1\} = \{m_0, m_1, m_i, m_\infty\}$$

is equal to

$$\{M_0C_2, M_1C_2; M_iC_2, M_\infty C_2\} = \{m'_0, m'_1, m'_i, m'_\infty\} .$$

Therefore, according to Chasles' theorem [7] both centers of projection C_1 and C_2 are on the same conic. The same analysis can be made when replacing M_i by M_j and since both conics share five common points M_0, M_1, M_∞, C_1 and C_2 they are equal. Note that since the cross-ratio is invariant to translation the proof is also valid for m'_0, m'_1, m'_i, m'_j having coordinates $d, d+1, d+i$ and $d+j$ respectively. \square

Note that with 3 fixed points, i.e. M_∞, C_1, C_2 , there is a two degree family of possible conics left. One of the degrees of freedom corresponds to the actual disparity value that is considered, the other degree of freedom corresponds to the ratio of focal lengths for both images (i.e. the size of pixels).

It is also interesting to extend the analysis of the iso-disparity curves out of the epipolar plane. Let us for example consider all the image points with equal λ . By definition these form a circle with the epipole as center (and radius $l_0 + \lambda$). The corresponding points for a disparity d are located on a circle³ with center e' and radius $l'_0 + \xi\lambda + d$. In general this type of iso-disparity curves will therefore be the intersection of two elliptic cones. Such a cone would be circular when the baseline is orthogonal to the image plane

³ Depending on ξ it might be necessary to consider two separate half-circles. The separating line would be a line through the epipole and parallel with the other image plane. This is again related to the orientation of the epipolar lines [13].

and would become a plane when the baseline is parallel with the image plane. Note that –if we would desire to do so– the intersection of two cones is relatively simple to compute [14]. Since in our case the vertex of each cone is comprised within the other cone, a single connected intersection is obtained. If the epipoles are far from the images, the radii of the circles passing through the images are large and therefore the iso-disparity surfaces would have low curvatures (at least in the direction out of the epipolar planes). Some specific cases will be discussed further.

4 Related Work

The special case for $d = 0$ is very much related with the concept of the *horopter* [1]. This curve is the locus of all 3D points that have identical image projections in both images of a stereo pair. Note that this is different from our definition of disparities since we are concerned with distance along the epipolar lines, but it can (partially) coincide in a number of cases. For the typical case of two cameras fixating a point, the horopter corresponds to an iso-disparity surface at least for the part contained within the horizontal plane. This part of the horopter is an ellipse similar to the one shown in Figure 2. In this case the vertical part of the horopter is a vertical line in the bisecting plane.

In fact, computer vision researchers have already used the concept of the horopter –or iso-disparity surfaces– in the context of stereo matching. Burt et al. [6] proposed to warp the images so that the horopter would better correspond to the expected scene content. By shifting scanlines the horopter of the standard stereo setup was transformed from the plane at infinity to the ground plane which is very useful in the context of robot navigation. We will show in Section 7 how this can also be achieved for a vertical wall by reconfiguring the camera. Others have also studied the possible use of the horopter in stereo vision, especially in the context of active stereo heads with verging [18] and torsion [12] degrees of freedom, or with asymmetrically skewed stereo cameras [8]. We will explore this possibility more in detail in the Section 7.

In [23] Völpel and Theimer also discuss iso-disparity loci and their relation to the reconstruction uncertainties in area-based stereo. However, instead of considering disparities along the epipolar lines, they consider independent disparities along the x - and y -direction. While our analysis is relevant to stereo algorithms that search for matches along epipolar lines, theirs is relevant to algorithms that look for corresponding pixels with similar coordinates without considering epipolar geometry. It is important to notice that this is fundamentally different and that both analyses only overlap when corresponding epipolar lines are aligned with corresponding image scanlines, i.e. for the standard stereo configuration and for a single scanline for some other stereo configurations. The analysis given in [23] is only meaningful for stereo configurations that do not deviate too much from the standard stereo case.

5 Human Vision

The horopter has also been the subject of a lot of study in the area of human vision. Under the assumption that the retina is spherical and that corresponding points are spread symmetrically around the fovea, the horopter becomes a circle known as the Vieth-Müller

circle [17]. It has, however, been observed that the empirical horopter deviates from the theoretical horopter by what is known as the Hering-Hillebrand deviations [11]. The empirical horopter is always flatter than the Vieth-Müller circle. It is concave towards the viewer for close distances, becomes flat at a specific distance known as the abathical distance and then becomes convex. Ogle [16] proposed to model the empirical horopter using a conic section.

Using the concepts discussed earlier in this paper some interesting observations can be made. Since the horopter is flat at the abathical distance, the eye (at least along the eye's equator) can be modeled as a camera with a planar retina with the plane offset by an angle corresponding to the vergence required to fixate a point at the abathical distance. *In other words, the retinas acts as if they were flat!* As far as we could tell, this seems to be a new result [11]. Given a typical abathical distance of 1m and an interocular distance of 6.5cm, the normal of the plane would deviate 1.86 degrees outwards from the visual axis. In this case Theorem 1 allows us to predict the shape of the horopter for any gaze direction and distance under the assumption that corresponding optical directions remain in correspondence (which is only an approximation given accommodation and other effects). If our eyes fixate a point straight in front of us, the 3 fixed points of Theorem 1, the fixation point and the need for symmetry are sufficient to determine uniquely the shape of the horizontal horopter. For excentric fixation, it can easily be shown that in the point of fixation the horopter should be tangent to the Vieth-Müller circle, i.e. the circle through both nodal points and the fixation point (given that the resolution of both fovea are the same), which in addition to the 4 known points also completely determines the shape of the horopter. This is illustrated with a few examples in Fig. 3.

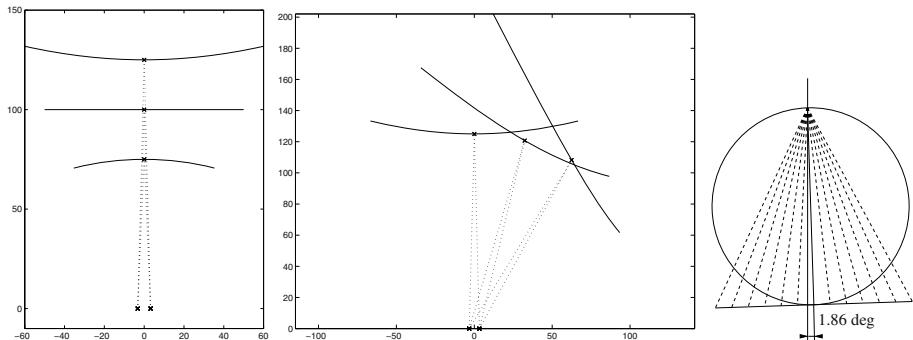


Fig. 3. Human horopter, as predicted by Theorem 1 (assuming an abathical distance of 1m and fixed corresponding points) for fixation at 75, 100 and 125cm (left) and for fixation at 125cm with 0, 15 and 30 degrees excentricity (middle). Simplified model of left eye with respect to stereo.

6 Rectification

To apply a stereo algorithm to a stereo pair that was not recorded using the standard stereo geometry, it is necessary to warp the images to make corresponding epipolar lines coincide with the scan-lines. This process is called *image-pair rectification*. In the past many different approaches have been proposed [4,10,15,19,9,21,20]. All these approaches somehow try to minimize distortions. Most of the approaches are homography based. For such approaches the epipole has to be warped to infinity to achieve parallel epipolar lines. In this case the resulting iso-disparity surfaces correspond to a set of planes, as in the standard stereo case or as the case with zoom discussed in Section 7.2. Clearly, when the original iso-disparity surfaces were far from planar, this type of rectification is bound to cause large distortions, at least for some parts of the images and of the iso-disparity surfaces.

Some recent approaches have abandoned the limitations imposed by homographies and have resorted to more general warping functions [21,20]. Both approaches preserve distances along epipolar lines. Therefore, these approaches preserve the iso-disparity surfaces. A disadvantage is that these rectification approaches are computationally more expensive and do not preserve lines. Note, however, that nowadays arbitrary warping functions can efficiently be performed on graphics hardware using texturing functionality.

By using an iso-disparity preserving rectification and a stereo algorithm that computes integer disparity values, the reconstructed points will have to be located on a finite set of surfaces corresponding to the iso-disparity surfaces for integer disparities. Ideally, one would expect every scene point to be projected on the closest iso-disparity surface by the stereo reconstruction. In Figure 4 a point reconstruction is shown that was obtained using a rectification approach that preserves distances along epipolar lines. The reconstruction was obtained from the two castle images shown at the top of the figure using a stereo algorithm without subpixel matching. Note the clearly visible elliptic patterns in the top view of the reconstruction. Our second example consists of a forward moving camera. In this case it can be verified that the iso-disparity surfaces are predicted to be surfaces of revolution around the optical axis with parabola passing through both centers of projection as generators. This example is illustrated in Fig.5. The reconstruction was obtained from the two beguinage images shown on the left. The observed structures correspond to the prediction, i.e. the intersection of the 3D scene with the iso-disparity surfaces.

While one can of course use an algorithm with subpixel precision, this doesn't change the fact that the depth precision of a stereo algorithm at a certain location will be proportional to the distance between the iso-disparity curves and that the algorithm will perform better if its priors (i.e. constant disparity) is aligned with the geometry of the observed scene.

7 Active Vision

It has been established long ago [5,2] that vision systems that would adapt themselves to the task at hand and/or to the observed environment can perform much better. The insights developed in this paper coupled with rectification algorithms that preserved the

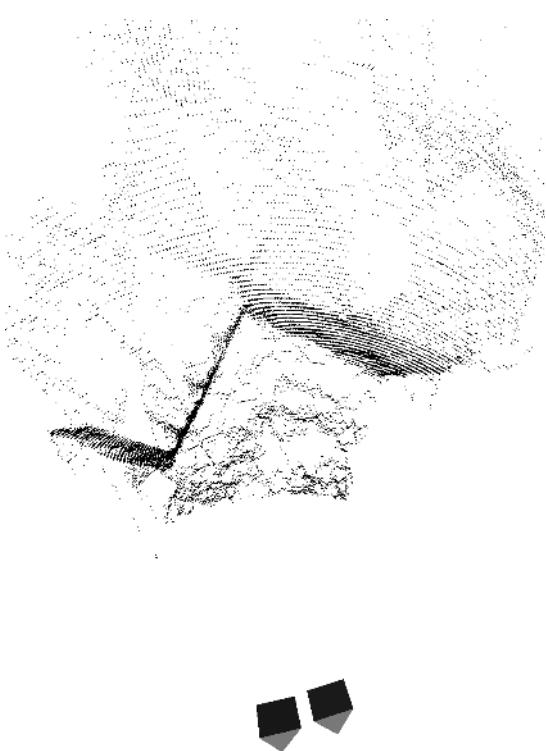


Fig. 4. Illustration of isodisparity surfaces for sideways motion (with some convergence). Image pair from the *castle* sequence (left) and 3D point reconstruction (right).

disparities can be very useful to provide the most optimal configuration for an active stereo head. Below we discuss the case of a verging stereo head and the case of a stereo head where the zoom of both cameras can be modified independently. Note that as mentioned earlier on todays hardware the expensive step for advanced rectification approaches is not the image warping anymore (since this can easily be achieved in real-time on graphics hardware), but the computation of the warping function. For an active vision scenario, one could compute the warping function once for many frames by keeping the configuration fixed for a number of consecutive frames, or, alternatively, precompute the warping function for a number of preset camera configurations. Those camera configurations would be matched to solve specific tasks, such as *reconstruct-left-wall*, *reconstruct-right-wall*, *reconstruct-object-in-front*, etc.

7.1 Vergence

Here we consider a typical active stereo head with verging capabilities. We study a synthetic configuration similar to the one shown in Figure 1, i.e we use the same virtual camera (width, height and focal length equal to 1000 pixels) and baseline (20cm). It is interesting to observe how small variations of the vergence angle cause important

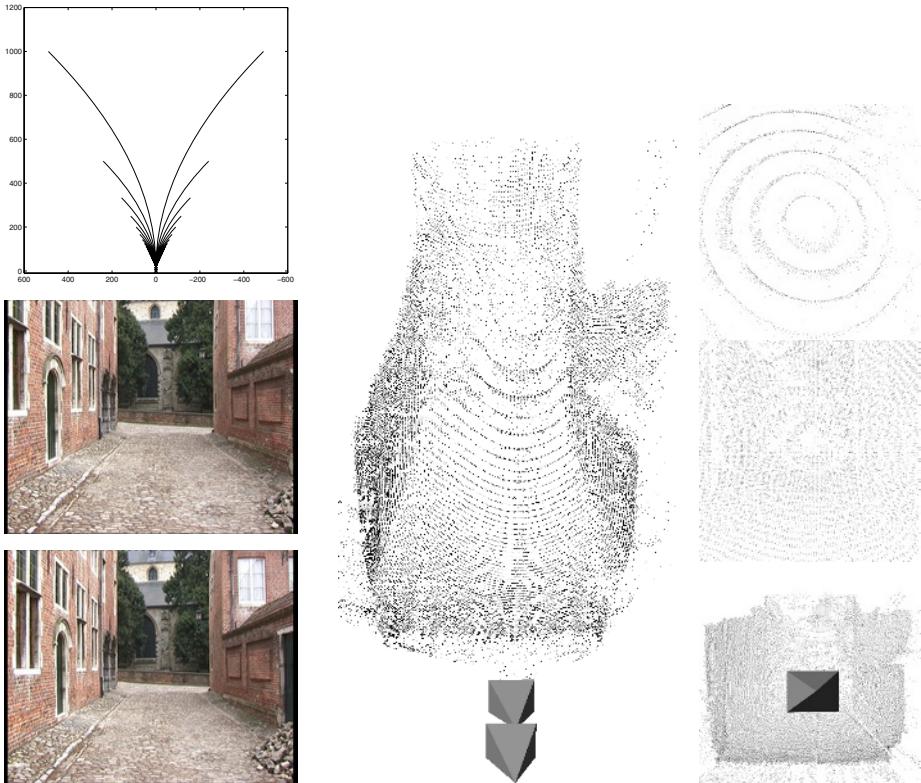


Fig. 5. Iso-disparity surfaces for forward motion. Top view of theoretical iso-disparity curves (upper-left). Notice that as expected reconstruction uncertainty is very large close to the direction of motion. Image pair from the *beguinage* sequence (lower-left) and different views of the 3D point reconstruction (middle/right). The images on the right are taken at different locations along the optical axis of the cameras.

changes for the iso-disparity surfaces. In Figure 6 iso-disparity curves are plotted for the plane containing both optical axes with vergence going from 10 degrees divergent to 15 degrees convergent.

It seems that divergent setups are better suited for observing objects (as the iso-disparity contours would more closely follow the expected object surface), convergent setups are better for exploring an environment. Note for example how some iso-disparity curves would be nicely aligned with a 2m wide corridor in the 10° convergent case. A possible limitation for divergent configurations is the more limited overlap. However, as can be seen in this figure, for cameras with a field of view 53 degrees, a stereo field of view of 33 resp. 43 degrees is obtained for 5 and 10 degrees divergence. Note also that, by shifting the CCD in the camera similarly to the optical keystone found in many projectors, it would be possible to obtain a much larger overlap for divergent configurations.

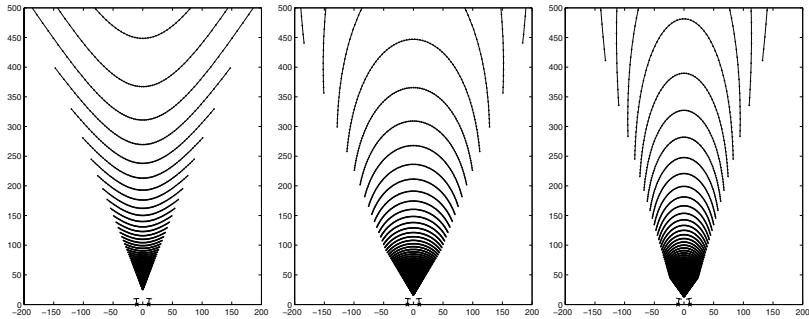


Fig. 6. Iso-disparity curves for different vergences, i.e. -5, 5 and 10 degrees for each camera (units are in cm).

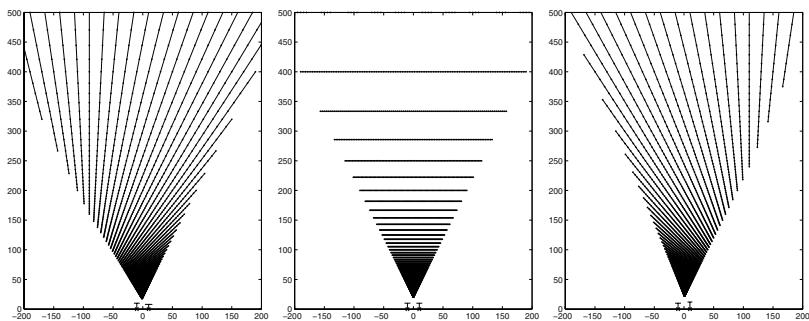


Fig. 7. Iso-disparity curves for different focal lengths (and parallel optical axis). The focal length of the left camera is 1000 and for the right camera 800, 1000 and 1200 (units are in cm).

7.2 Zooming

For active stereo zooming can also be used to change the shape of the iso-disparity surfaces depending on the task at hand. If the camera are kept parallel the iso-disparity surfaces remain a bundle of planes, but not fronto-parallel anymore. In Figure 7 a few examples are shown. Note how well zoomed in/out configurations are suited to reconstruct or follow a wall. In Figure 8 two stereo pairs are given recorded using a pair of pan-tilt-zoom cameras. The stereo pair is interlaced into a single image so that the disparities can be verified visually. The disparity images were computed using a standard stereo algorithm. The first stereo pair corresponds to the standard fronto-parallel stereo setup. The second stereo pair corresponds to the case shown on the left of Figure 7. One can clearly notice how for the second case the whole left wall fits within a very small disparity range. This benefits both the efficiency of the algorithm (less disparities to search) and the quality of the results (assumptions for area-based stereo are better met). Note that the same effect could be achieved by distorting standard stereo images along epipolar lines, but in this case this would imply a loss of information or efficiency (to the contrary of the groundplane rectification case [6] where one could shift epipolar lines with respect to each other). More in general, this paper makes it possible for a robot



Fig. 8. Stereo pair (interlaced to form single image) and corresponding disparity map for fronto-parallel case (top) and right-camera zoomed-out case (bottom, see Fig. 7, left).

equipped with a pair of pan-tilt-zoom cameras that explores an environment to actively control its cameras to 'focus in' on areas of interest. This should be understood in the stereo sense, meaning bringing the structure of interest within a small disparity range. Note that small changes in intrinsics and/or orientation will not affect much the depth discretisation of a particular stereo configuration. However, as we have shown it can have a great impact on the shape of the iso-disparity curves or in other words determine if the assumptions made by the stereo algorithm will be satisfied or not.

8 Summary and Conclusions

In this paper we have derived the shape of iso-disparity surfaces for general camera configuration. It was shown that the intersection of these implicit surfaces with epipolar planes have a particularly simple form. Comparing this with the empirical human horopter, we could conclude that the retinas act as if they were flat. This, as far as we were able to verify, is a new result and might lead to interesting further research in the area of human vision. The main goal of this paper was to provide more insight in the iso-disparity geometry for general camera configurations. We have discussed and illustrated the impact on rectification and advocate the use of disparity preserving rectification. Finally, we have shown how the insights developed in this paper can be exploited to 'focus' an active stereo head on a part of the scene. This allows to bring structures of interest within a small disparity range by controlling vergence and zooming, and thus allows to achieve better reconstruction of those structures faster.

Acknowledgments. The financial support of the NSF CAREER award IIS 0237533 and the (Darpa) NBC/DOI grant NBCH-1-03-0015 are gratefully acknowledged.

References

1. F. Aguilionius, *Opticorum Libri Sex*, Ex off. Plantiniana apud viduam et filios Jo. Moreti, Antwerpen 1613.
2. Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision", *Proc. ICCV*, pp. 35-54, 1987.
3. L. Alvarez, R. Deriche, J.Sánchez and J.Weickert, "Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach." *Journal of Visual Communication and Image Representation*. Vol. 13, No. 1/2, pp 3-21, March/June 2002.
4. N. Ayache, C. Hansen, "Rectification of images for binocular and trinocular stereovision", *Proc. International Conference on Pattern Recognition*, pp. 11-16, 1988.
5. R. Bajcsy, "Active Perception," *Proc. of the IEEE*, 76:996-1005, 1988.
6. P. Burt, L. Wixson, G. Salgian, "Electronically Directed 'Focal' Stereo", *Proc. ICCV*, pp. 97-101, 1995.
7. M. Chasles, *Traité des Sections Coniques.*, Gauthier-Villars, Paris, 1865.
8. A. Fransisco, F. Bergholm, "On the Importance of Being Asymmetric in Stereopsis—Or Why We Should Use Skewed Parallel Cameras", *IJCV* 29(3), 181-202 (1998).
9. J. Gluckman, S. Nayar, "Rectifying Transformations That Minimize Resampling Effects", *Proc. CVPR*, Vol. 1, pp. 111-117, 2001.
10. R. Hartley, "Theory and practice of projective rectification", *IJCV*, 35(2), pp. 115–127, November 1999.
11. I. Howard and B. Rogers, "Depth perception", *Seeing in Depth*, Vol. 2, Porteous, 2002.
12. M. Jenkin, J. Tsotsos, "Active stereo vision and cyclotorsion", *Proc. CVPR*, pp. 806—811, 1994.
13. S. Laveau and O. Faugeras. "Oriented projective geometry for computer vision". *Computer Vision - ECCV'96*, LNCS, Vol. 1064, Springer-Verlag, pp. 147-156, 1996.
14. J. Levin, "Mathematical models for determining the intersections of quadric surfaces", *Computer Graphics and Image Processing* 11(1):73-87, 1979.
15. C. Loop and Z. Zhang. "Computing Rectifying Homographies for Stereo Vision". *Proc. CVPR*, Vol.I, pages 125-131, 1999.
16. K. Ogle, "An analytic treatment of the longitudinal horopter; its measurements and the application to related phenomena especially to the relative size and shape of the ocular images", *Journal of Optical Society of America*, Vol. 22, pp. 665-728, 1932.
17. K. Ogle, *Researches in Binocular Vision*, W.B. Saunders company, Philadelphia & London, 1950.
18. T. Olson, "Stereopsis of Verging Systems", *Proc. CVPR*, pp. 55-60, 1993.
19. D. Papadimitriou and T. Dennis. "Epipolar line estimation and rectification for stereo images pairs", *IEEE Transactions on Image Processing*, 3(4):672-676, April 1996.
20. M. Pollefeys, R. Koch, L. Van Gool, "A simple and efficient rectification method for general motion", *Proc. ICCV*, pp. 496–501, 1999.
21. S. Roy, J. Meunier, I. Cox. "Cylindrical rectification to minimize epipolar distortion", *Proc. CVPR*, pp. 393–399, 1997.
22. D. Scharstein, R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", *IJCV*, Volume 47, Issue 1-3, pp. 7-42, April - June 2002.
23. B. Völpel, W.M. Theimer, "Localization Uncertainty In Area-Based Stereo Algorithms", *T-SMC(25)*, 1995, pp. 1628-1634.

Camera Calibration with Two Arbitrary Coplanar Circles

Qian Chen, Haiyuan Wu, and Toshikazu Wada

Faculty of Systems Engineering, Wakayama University,
Wakayama City, Wakayama, 640-8510, Japan.

{wuhy, chen, twada}@sys.wakayama-u.ac.jp, <http://www.wakayama-u.ac.jp/~chen>

Abstract. In this paper, we describe a novel camera calibration method to estimate the extrinsic parameters and the focal length of a camera by using only one single image of two coplanar circles with arbitrary radius.

We consider that a method of simple operation to estimate the *extrinsic parameters* and the *focal length* of a camera is very important because in many vision based applications, the position, the pose and the zooming factor of a camera is adjusted frequently.

An easy to use and convenient camera calibration method should have two characteristics: 1) the calibration object can be produced or prepared easily, and 2) the operation of a calibration job is simple and easy. Our new method satisfies this requirement, while most existing camera calibration methods do not because they need a specially designed calibration object, and require multi-view images. Because drawing beautiful circles with arbitrary radius is so easy that one can even draw it on the ground with only a rope and a stick, the calibration object used by our method can be prepared very easily. On the other hand, our method need only one image, and it allows that the centers of the circle and/or part of the circles to be occluded.

Another useful feature of our method is that it can estimate the focal length as well as the extrinsic parameters of a camera simultaneously. This is because zoom lenses are used so widely, and the zooming factor is adjusted as frequently as the camera setting, the estimation of the focal length is almost a must whenever the camera setting is changed. The extensive experiments over simulated images and real images demonstrate the robustness and the effectiveness of our method.

1 Introduction

Calibration of the extrinsic camera parameters is an indispensable preparation for computer vision tasks such as environment recognition, 3D shape acquirement and so on. In many real vision based applications, camera setting is adjusted frequently, and whenever the camera setting has been altered, the extrinsic camera parameters have to be estimated again. In recent years, zoom lenses have being widely used, and zooming factor is adjusted as frequently as other camera parameters. Thus when the position or pose of a camera has been adjusted, the focal length might also have been altered in most cases. Therefore, we consider that a method of simple operation to estimate the *extrinsic parameters* and the *focal length* of a camera simultaneously is highly desired. Such a

method should have two characteristics: 1) the calibration object can be produced or prepared easily, and 2) the operation of a calibration job is simple and easy.

One of the conventional methods to calibrate the extrinsic parameters and the focal length of a camera is to use point correspondence data. In order to get precise results, point correspondence data spreading over an image plane is necessary. However, it is difficult to prepare a lot of points with known 3D coordinates and to find the correspondence between the 3D points and their projection in the image. Although specially designed calibration objects can ease this job, producing such an object itself and setting it properly for the calibration is still complicated and time consuming, and sometime becomes impossible or impractical, e.g. in the case of wide observing area, such as a baseball stadium or a football playground.

The usage of point corresponding data can be avoided by using geometrical patterns such as straight lines and circles instead. Several researches using circular patterns, or conic patterns [2]-[6] have been reported so far. These camera calibration methods are for estimating intrinsic camera parameters, and they all use some special patterns and multi-view images.

Meng *et al.*[2] proposed a method using a pattern that consists of a circle and straight lines passing through its center. It needs at least three different views. Kim *et al.*[4] proposed a method that makes use of planar con-centric circles. It requires two views. Yang *et al.*[5] proposed a similar method except that con-centric ellipses are used instead of con-centric circles.

Other methods are about motion analysis or 3D interpretation of conic[7]-[16]. Although some of them can be used as a calibration method, they have some or all of the following disadvantages, 1) multi-view images are required, 2) only part of the extrinsic camera parameters can be estimated, 3) the focal length can not be estimated, 4) a specially designed calibration object is required.

Long[7] proposed a method to find the correspondence between conics in two views, and to estimate the relative orientation of the optical axis of two views. Dhome *et al.*[8] proposed a method to estimate the attitude and the position of a circle from an image assuming known focal length and radius. Kanatani and Wu[15],[16] reported methods to extract 3D information from conics in images. The intrinsic and extrinsic camera parameters are supposed to be known.

In this paper, we describe a novel camera calibration method to estimate the extrinsic parameters and the focal length of a camera by using only one single image of two co-planar circles with arbitrary radius. Because drawing beautiful circles with arbitrary radius is so easy that one can even draw it on the ground with only a rope and a stick, the calibration object used by our method can be prepared very easily. On the other hand, our method need only one image, and it allows that the centers of the circle and/or part of the circles to be occluded. These features make the operation of camera calibration using our method becoming very simple and easy. Another useful feature of our method is that it can estimate the focal length as well as the extrinsic parameters of a camera simultaneously.

The extensive experiments over simulated images and real images demonstrate the robustness and the effectiveness of our method.

2 Elliptical Cone and Circular Cross Section

In this section we describe the problem of estimating the direction and the center of a circle from one perspective view. M.Dhome[8] addressed this problem in a research about the pose estimation of an object of revolution. We give a rigorous description here, which is then used in the estimation of the focal length and the extrinsic parameters of the camera in the succeeding section.

2.1 Ellipses and Conic Surfaces

If a circle is projected on to the image plane with perspective projection, it shows an ellipse in general case. Considering a camera coordinate system that the origin is the optical center and the Z -axis is the optical axis, then the ellipse in the image can be described by the following equation,

$$Ax_e^2 + 2Bx_e y_e + Cy_e^2 + 2Dx_e + 2Ey_e + F = 0, \quad (1)$$

or in quadratic form as following,

$$\begin{pmatrix} x_e & y_e & 1 \end{pmatrix} \begin{pmatrix} A & B & D \\ B & C & E \\ D & E & F \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ 1 \end{pmatrix} = 0. \quad (2)$$

A bundle of straight lines passing through the optical center and the ellipse defines an oblique elliptical cone. Assuming that the focal length of the camera is f , the image plane can be expressed by $z = -f$. Then the oblique elliptical cone can be described by,

$$\mathbf{P} = k(x_e \ y_e \ -f)^T, \quad (3)$$

where k is a scale factor describing the distance from the origin to \mathbf{P} . From Eq.(2) and Eq.(3) the equation to describe the oblique elliptical cone is derived,

$$\mathbf{P}^T \mathbf{Q} \mathbf{P} = 0, \quad (4)$$

where

$$\mathbf{Q} = \begin{pmatrix} A & B & -\frac{D}{f} \\ B & C & -\frac{E}{f} \\ -\frac{D}{f} & -\frac{E}{f} & \frac{F}{f^2} \end{pmatrix}. \quad (5)$$

Considering a supporting plane coordinate system that the origin is also the optical center, but the Z -axis is defined by the unit normal vector of the supporting plane of the circle to be viewed. Let z_0 be the Z coordinate of points on the plane, the points on the circle can be described by the following expression,

$$\begin{cases} (x - x_0)^2 + (y - y_0)^2 = r^2 \\ z = z_0 \end{cases} \quad (6)$$

where (x_0, y_0, z_0) is the center and r is the radius of the circle. A bundle of straight lines passing through the optical center and the circle defines an oblique circular cone described by the following equation,

$$\mathbf{P}_c^T \mathbf{Q}_c \mathbf{P}_c = 0, \quad (7)$$

where

$$\mathbf{Q}_c = \begin{pmatrix} 1 & 0 & -\frac{x_0}{z_0} \\ 0 & 1 & -\frac{y_0}{z_0} \\ -\frac{x_0}{z_0} & -\frac{y_0}{z_0} & \frac{x_0^2 + y_0^2 - r^2}{z_0^2} \end{pmatrix}. \quad (8)$$

Since the camera coordinate system and the supporting plane coordinate system have a common origin at the optical center, the transform between the two coordinate systems is a rotation. Because the oblique circular cone and the oblique elliptical cone are the same cone surface, there exists a rotation matrix \mathbf{R}_c that transforms \mathbf{P}_c to \mathbf{P} as following,

$$\mathbf{P} = \mathbf{R}_c \mathbf{P}_c. \quad (9)$$

Since $k\mathbf{Q}_c$ for any $k \neq 0$ describes the same cone as of \mathbf{Q}_c , from Eq.(9), Eq.(7) and Eq.(4) we have,

$$k\mathbf{R}_c^T \mathbf{Q} \mathbf{R}_c = \mathbf{Q}_c. \quad (10)$$

In order to determine \mathbf{R}_c and \mathbf{Q}_c so that the unit normal vector of the supporting plane and the center of the circle can be obtained, we want to convert \mathbf{Q} to a diagonal matrix first.

Let $\lambda_1, \lambda_2, \lambda_3$ be the eigen-values, and $\mathbf{v}_1 = (v_{1x} v_{1y} v_{1z})^T$, $\mathbf{v}_2 = (v_{2x} v_{2y} v_{2z})^T$, $\mathbf{v}_3 = (v_{3x} v_{3y} v_{3z})^T$ be the normalized eigen-vectors of \mathbf{Q} respectively, \mathbf{Q} can be expressed by the following equation,

$$\mathbf{Q} = \mathbf{V} \Lambda \mathbf{V}^T, \quad (11)$$

where

$$\begin{cases} \Lambda = \text{diag}\{\lambda_1, \lambda_2, \lambda_3\} \\ \mathbf{V} = (\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3) \end{cases}. \quad (12)$$

Substituting Eq.(11) for \mathbf{Q} in Eq.(10), the following equation is obtained,

$$k\mathbf{R}^T \Lambda \mathbf{R} = \mathbf{Q}_c, \quad (13)$$

where

$$\mathbf{R} = \begin{pmatrix} r_{1x} & r_{2x} & r_{3x} \\ r_{1y} & r_{2y} & r_{3y} \\ r_{1z} & r_{2z} & r_{3z} \end{pmatrix} = \mathbf{V}^T \mathbf{R}_c. \quad (14)$$

From Eq.(13) we obtain following equations,

$$\begin{cases} \lambda_1(r_{1x}^2 - r_{2x}^2) + \lambda_2(r_{1y}^2 - r_{2y}^2) + \lambda_3(r_{1z}^2 - r_{2z}^2) = 0 \\ \lambda_1 r_{1x} r_{2x} + \lambda_2 r_{1y} r_{2y} + \lambda_3 r_{1z} r_{2z} = 0 \end{cases}. \quad (15)$$

Without losing generality, we assume that

$$\begin{cases} \lambda_1\lambda_2 > 0 \\ \lambda_1\lambda_3 < 0 \\ |\lambda_1| \geq |\lambda_2| \end{cases} . \quad (16)$$

By simplifying Eq.(15) and $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, we obtain,

$$\mathbf{R} = \begin{pmatrix} g \cos \alpha & S_1 g \sin \alpha & S_2 h \\ \sin \alpha & -S_1 \cos \alpha & 0 \\ S_1 S_2 h \cos \alpha & S_2 h \sin \alpha & -S_1 g \end{pmatrix}, \quad (17)$$

where α is a free variable, S_1 and S_2 are undetermined signs, and

$$\begin{cases} g = \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}}, \\ h = \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \end{cases}, \quad (18)$$

By substituting Eq.(17) for \mathbf{R} in Eq.(13), k , x_0/z_0 , y_0/z_0 and r/z_0 are determined,

$$\begin{cases} k = \lambda_2 \\ \frac{x_0}{z_0} = -S_2 \frac{\sqrt{(\lambda_1 - \lambda_2)(\lambda_2 - \lambda_3)} \cos \alpha}{\lambda_2} \\ \frac{y_0}{z_0} = -S_1 S_2 \frac{\sqrt{(\lambda_1 - \lambda_2)(\lambda_2 - \lambda_3)} \sin \alpha}{\lambda_2} \\ \frac{r^2}{z_0^2} = -\frac{\lambda_1 \lambda_3}{\lambda_2^2} \end{cases} \quad (19)$$

Because the Z -axis of the supporting plane coordinate system is the unit normal vector of the plane (denoted by \mathbf{N}), from Eqs.(9), (14) and (19), \mathbf{N} and the center of the circle (denoted by \mathbf{C}) described in the camera coordinate system can be computed by the following expression,

$$\begin{cases} z_0 = S_3 \frac{\lambda_2 r}{\sqrt{-\lambda_1 \lambda_3}} \\ \mathbf{C} = z_0 \mathbf{VR} \begin{pmatrix} x_0/z_0 \\ y_0/z_0 \\ 1 \end{pmatrix} = z_0 \mathbf{V} \begin{pmatrix} S_2 \frac{\lambda_3}{\lambda_2} \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ 0 \\ -S_1 \frac{\lambda_1}{\lambda_2} \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{pmatrix}, \\ \mathbf{N} = \mathbf{VR} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{V} \begin{pmatrix} S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ 0 \\ -S_1 \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{pmatrix} \end{cases}, \quad (20)$$

where S_3 is also an undetermined sign.

Since a plane has two side, we let \mathbf{N} be the normal vector indicating the side faced to the camera. Also, since the center of the circle is in front of the camera, the following stands,

$$\begin{cases} \mathbf{N} \cdot (0 \ 0 \ 1)^T > 0 \\ \mathbf{C} \cdot (0 \ 0 \ 1)^T < 0 \end{cases}, \quad (21)$$

from which two of the three undetermined signs in Eq.(20) can be determined, and we have two sets of possible answers about \mathbf{N} and \mathbf{C} . In the case of unknown radius, the r is left a scale factor in Eq.(20).

2.2 Estimating the Extrinsic Parameters and the Focal Length Simultaneously

In this section, we describe a method to estimate the extrinsic parameters and the focal length of a given camera by using two coplanar circles with arbitrary radius. As described in the section 2.1, the unit normal vector of the supporting plane and the center of the circle can be determined from one perspective image, if the focal length is known.

In the case of unknown focal length, the symbol f in Eq.(5) leaves a variable, and it will remain in all the answers.

In order to determine the focal length so that the unit normal vector of the supporting plane and the center of the circle can be determined, we let the camera to view a scene consists of two coplanar circles. In this case, two ellipses will be detected, and according to section 2.1, two oblique elliptical cones can be formed from the detected ellipses if we give a focal length. From each of them, the normal vector of the supporting plane can be estimated independently.

If we have given a wrong focal length, each of the formed cones will be deformed in different ways and will not be similar to the real cone surfaces. In this case, the estimated unit normal vectors of the supporting plane from each of the two cones will not only be different from the real one, but will not be parallel to each other too. Only if we give the correct focal length, the unit normal vectors estimated from each of the detected ellipses will be the same.

Let $\mathbf{N}_1(f)$ denote the normal vector estimated from one of the two ellipses and $\mathbf{N}_2(f)$ denote the normal vector from the other one. Because the two circles are coplanar, $\mathbf{N}_1(f)$ and $\mathbf{N}_2(f)$ should be same. This constraint can be expressed by the following equation,

$$\mathbf{N}_1(f) \cdot \mathbf{N}_2(f) = 1. \quad (22)$$

Then by minimizing the following expression, the focal length f and the unit normal vector $\mathbf{N} (= \mathbf{N}_1 = \mathbf{N}_2)$ can be determined, and the ambiguity cased by the undetermined signs remained in Eq.(20) can be eliminated,

$$(\mathbf{N}_1(f) \cdot \mathbf{N}_2(f) - 1)^2 \rightarrow \min. \quad (23)$$

The centers of the two circles can also determined with Eq.(20). If the radius of the circles are unknown, then z_0 in Eq.(20) leaves a variable. Let \mathbf{C}_1 and \mathbf{C}_2 denote the centers of the two circles respectively, from Eq.(20) they can be expressed by,

$$\begin{cases} \mathbf{C}_1 = z_0 \mathbf{C}_{10} \\ \mathbf{C}_2 = z_0 \mathbf{C}_{20} \end{cases}, \quad (24)$$

where \mathbf{C}_{10} and \mathbf{C}_{20} can be computed from the detected ellipses. The distance between the two circle centers (d_{12}) can be calculated by the following expression,

$$d_{12} = |\mathbf{C}_1 - \mathbf{C}_2| = |z_0| |\mathbf{C}_{10} - \mathbf{C}_{20}|. \quad (25)$$

A world coordinate system $O-XYZ$ can be defined by using the two circle centers as reference points and the unit normal vector of the supporting plane as a reference direction as following. Let \mathbf{C}_1 be the origin, \mathbf{N} define the Z axis and the vector $\mathbf{C}_2 - \mathbf{C}_1$

define the direction of the X axis of $O-XYZ$ respectively, the origin \mathbf{O} , the unit vectors of X, Y, Z axes \mathbf{i}, \mathbf{j} and \mathbf{k} can be obtained by the following equation,

$$\begin{cases} \mathbf{O} = \mathbf{C}_1 \\ \mathbf{i} = \frac{\mathbf{C}_{20} - \mathbf{C}_{10}}{|\mathbf{C}_{20} - \mathbf{C}_{10}|} \\ \mathbf{k} = \mathbf{N} \\ \mathbf{j} = \mathbf{k} \times \mathbf{i} \end{cases} . \quad (26)$$

If one of the radius of the two circles, the distance between the two circle center or the distance between the optical center and the supporting plane is known, or if we use one of them as the unit length of $O-XYZ$, then z_0 can be determined from Eq.(20) or Eq.(25), thus \mathbf{O} and all other parameters related to length will be determined.

Then the optical center \mathbf{O}' , the unit vectors $\mathbf{i}', \mathbf{j}',$ and \mathbf{k}' that define the X, Y and Z axis of the camera coordinate system described in the world coordinate system $O-XYZ$ can be computed by the following equation,

$$\begin{cases} \mathbf{O}' = \begin{pmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{pmatrix} \left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \mathbf{O} \right] \\ \mathbf{i}' = \begin{pmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ \mathbf{j}' = \begin{pmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \\ \mathbf{k}' = \begin{pmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \end{cases} . \quad (27)$$

Therefore, by taking an image of a scene of two coplanar circles, the unit normal vector of the supporting plane containing the circles and the focal length can be estimated. And if the centers of the circle can be used as two reference point, then the full translation and rotation of the camera relative to the world coordinate system defined by the two circle centers and the normal vector of the supporting plane can be also determined. If neither of the radii of the circles or the distance between the two circle centers is available, the rotation and the translation of the camera can also be determined except that a scale factor remains undetermined. In both cases, the centers of the circles need not to be viewable in the image.

3 Experimental Results

In order to exam the usefulness and the effectiveness of the proposed algorithm; we first tested our method using some simulated images, then using some real images.

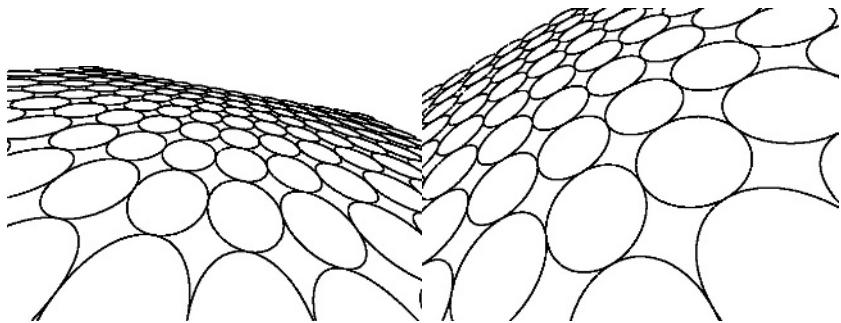


Fig. 1. Two sets of synthesized scenes of circles by CG: *case-1* and *case-2*.

Table 1. Estimated camera parameters

	<i>case-1</i>	RMS error	Standard deviation		<i>case-2</i>	RMS error	Stand deviation
$f(\text{pixel})$	5.52		9.21	$f(\text{pixel})$	7.19		11.89
$\beta(\text{degree})$	0.36		0.47	$\beta(\text{degree})$	0.11		0.15
$\theta(\text{degree})$	0.57		0.97	$\theta(\text{degree})$	0.51		0.85

3.1 Simulation Results

We used computer graphics to synthesize images of many different coplanar circle scenes. We first set the image resolution to 640×480 [pixel], the focal length to 200 [pixel], the tilt angle θ indicating the angle between the optical axis and the supporting plane to be 40 [degree], the roll angle β indicating the rotation about the optical axis to 10 [degree], and the distance between the optical center and the supporting plane to 3.0 [meter]. We called this camera setting as “*case-1*” hereafter. We use this camera setting to synthesize images of circles with a radius of 1.0 [meter].

Figure 1 shows an image containing all the circles used in the experiment of the “*case-1*” camera setting. We used 32 images containing two circles randomly selected from the ones shown in Figure 1.

We also have done a similar experiment using the camera setting called “*case-2*” of which the image resolution is same as the “*case-1*”, the focal length is 300 [pixel], $\theta = 50$ [degree], $\beta = 30$ [degree], the distance between the optical center and the supporting plane and the radius of the circles are same as “*case-1*”.

Figure 1 shows an image containing all the circles used in the experiment of the “*case-2*” camera setting. We used 17 images containing two circles randomly selected from the ones shown in Figure 1.

From each of the images, two ellipses were detected, which were used to estimate the unit normal vector of the supporting plane and the focal length. Then the estimated focal length and the tilt angle and the roll angle calculated from the estimated unit normal vector of the supporting plane were compared to the ground truth, which is the camera setting used to synthesize images with CG. The experimental results are summarized in Table 1 with suffixes 1 and 2.



Table scen1



Manhole on road



Table scene 2



Wave rings



Table scene 3

Fig. 2. Some images of real scene used the experiment.**Table 2.** Experimental results estimated from images shown in Figure 2

Image name	Resolution[pixel]	Focal length[pixel]	Unit Normal vector
Table scene 1	640×480	901.0	(0.03, 0.83, 0.56)
Table scene 2	640×480	1140	(0.13, 0.84, 0.51)
Table scene 3	1600×1200	2209	(-0.04, 0.83, 0.56)
Manhole on road	1600×1200	4164	(0.05, 0.97, 0.26)
Wave rings	275×412	2615	(0.03, 0.92, 0.39)

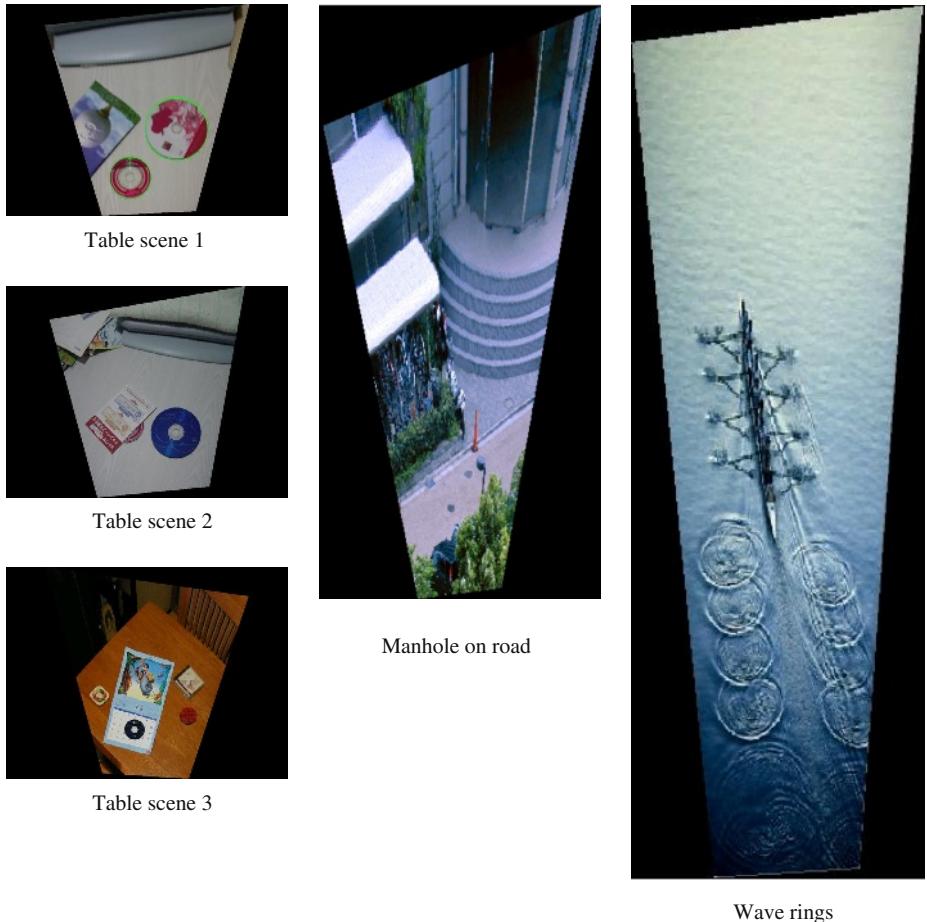


Fig. 3. Vertical views synthesized using estimated camera parameters.

3.2 Experiments with Real Images

We tested our method by applying it to many real images containing coplanar circle shape objects e.g., manholes on roads, CD discs on a table, widening rings on the water surface, and so on. Some of the images used in the experiment is shown in Figure 2).

The images were taken with two digital still cameras of different kind, and with a digital video camera. All of them are equipped with a zoom lens.

For each image, we detected the ellipses and used them to estimate the focal length and the unit normal vector of the supporting plane. The results are summarized in Table 1.

Since the ground truth of the camera setting including focal length, position and pose is not available, we used the estimated camera parameters to convert the image to a vertical view to the supporting plane by assuming a planar scene, and to see if it resembles the real scene. Figure 3.

In some of the converted images, the circular object does not show a perfect circle. The considerable reasons are, 1) in our method, the intrinsic parameters except the focal length are assumed to be calibrated, but uncalibrated cameras were used in the experiment, 2) the radial distortion of the cameras are not compensated.

4 Conclusion

This paper has presented a new camera calibration method for estimating the focal length and the extrinsic camera parameters using circular patterns. This method allows us to estimate the extrinsic parameters and focal length simultaneously using one single view of two coplanar circles with arbitrary radius. Moreover, it does not require the whole circles or the centers to be viewable. These features make a very convenient calibration method because both the preparation of the calibration pattern and the operation of taking picture are quite easy.

Compared with existing method, our method can determine the focal length of camera as well as extrinsic camera parameters. Even in the case that the position and the size of the circles are not available, our method can still give the focal length and the normal vector of the supporting plane.

We will extend our method estimating image center and the radial distortion parameters in the future work.

References

1. O.Faugeras, “Three-Dimensional Computer Vision: A Geometric Viewpoint”, *MIT Press* , 1993.
2. X.Meng and Z.Hu, “A New Easy Camera Calibration Technique Based on Circular Points”, *Pattern Recognition* , Vol. 36, pp. 1155-1164, 2003.
3. G.Wang, F.Wu and Z.Hu, “Novel Approach to Circular Points Based Camera Calibration”,
4. J.S.Kim , H.W.Kim and I.S.Kweon, “A Camera Calibration Method using Concentric Circles for Vision Applications”, *ACCV* pp. 23–25, 2002.
5. Yang, C., Sun, F. Hu, Z.: “Planar Conic Based Camera Calibration”, *ICPR*, 2000.
6. P.Gurdjos, A.Grouzil and R.Payrissat, “Another Way of Looking at Plane-Based Calibration: the Center Circle Constraint”, *ECCV*, 2002.
7. Long, Q.: “Conic Reconstruction and Correspondence From Two Views”, *PAMI* , Vol.18, No.2, pp. 151–160, 1996.
8. M.Dhome, J.T.Lapreste, G.Rives and M.Richetin, “Spatial Localization of Modelled Objects of Revolution in Monocular Perspective Vision”, *ECCV 90*, pp. 475–485, 1990.
9. S.Avidan and A.Shashua, “Trajectory Triangulation: 3D Reconstruction of Moving Points from a Monocular Image Sequence”, *PAMI* , Vol.22, No.4, pp. 348–357, 2000.
10. J.Heikkila and O.Silven, “A Four-step Camera Calibration Procedure with Implicit Image Correction”.
11. D.Forsyth, etc., “Invariant descriptors for 3-D object recognition and pose”, *IEEE. Trans. PAMI*, 13:971-991, 1991.
12. C.A.Rothwell, etc., “Relative motion and pose from arbitrary plane curves”, *Image Vis. Comput.* 10(4): 250-262, 1992.
13. R.Safaei-Red, etc., “Constraints on quadratic-curved features under perspective projection”, *Image Vis. Comput.* 19(8): 532-548, 1992.

14. R.Safaei-Red, etc, “Three-dimensional location estimation of circular features for machine vision”, *IEEE Trans. Robot Autom.* 8(5):624-640, 1992.
15. K.Kanatani and L.Wu, “3D Interpretation of Conics and Orthogonality”, *Image Understanding*, Vol.58, Nov, pp. 286-301, 1993.
16. L.Wu and K.Kanatani, “Interpretation of Conic Motion and Its Applications”, *Int. Journal of Computer Vision* , Vol.10, No.1, pp. 67–84, 1993.
17. P.Sturm, “A Case Against Kruppa’s Equations for Camera Self-Calibration”, *PAMI* , Vol.22, No.10, pp. 348–357, 2000.
18. R.Sukthankar, R.Stockton and M.Mullin, “Smarter Presentations: Exploiting Homography in Camera-Projector Systems”, *ICCV* pp. 247-253, 2001.
19. R.J. Holt and A.N.Netravali, “Camera Calibration Problem: Some New Result”, *CVIU* , No.54, Vol.3, pp. 368–383, 1991.
20. Z.Zhang, “A Flexible New Technique for Camera Calibration”, *PAMI* , Vol.22, No.11, pp. 1330–1334, 2000.
21. P.Sturm and S.Maybank, “On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications”, *CVPR* , pp. 432–437, 1999.
22. T.Wada, “Color-Target Detection Based on Nearest Neighbor Classifier: Example Based Classification and its Applications”, *JPJS SIG Notes, 2002-CVIM-134*, pp. 17–24, 2002.
23. Barreto, J.P. and Araujo H., “Issues on the Geometry of Central Catadioptric Image Formation”, *CVPR* , 2001.
24. A. Fitzgibbon, M.Pilu, and R.B.Fisher, “Direct Least Square Fitting of Ellipses”, *IEEE Trans. on Pattern Analysis and Machine Intelligence* , Vol.21, No.5, pp. 476–480, (1999).
25. R.Halir, and J.Flusser, “Numerically Stable Direct Least Squares Fitting of Ellipses”, *WSCG*, 1998.

Reconstruction of 3-D Symmetric Curves from Perspective Images without Discrete Features

Wei Hong¹, Yi Ma¹, and Yizhou Yu²

¹ Department of Electrical and Computer Engineering

² Department of Computer Science

University of Illinois at Urbana-Champaign

Urbana, IL, 61801, USA

{weihong,yima,yyz}@uiuc.edu

Abstract. The shapes of many natural and man-made objects have curved contours. The images of such contours usually do not have sufficient distinctive features to apply conventional feature-based reconstruction algorithms. This paper shows that both the shape of curves in 3-D space and the camera poses can be accurately reconstructed from their perspective images with unknown point correspondences given that the curves have certain invariant properties such as symmetry. We show that in such cases the minimum number of views needed for a solution is remarkably small: one for planar curves and two for nonplanar curves (of arbitrary shapes), which is significantly less than what is required by most existing algorithms for general curves. Our solutions rely on minimizing the L^2 -distance between the shapes of the curves reconstructed via the “epipolar geometry” of symmetric curves. Both simulations and experiments on real images are presented to demonstrate the effectiveness of our approach.

1 Introduction

One of the main thrusts of research in computer vision is to study how to reconstruct the shapes of 3-D objects from (perspective) images. Depending on the choice of the models for the shape (e.g., surfaces, volumes, or points), the reconstruction methods differ. On one hand, surface- and volume-based approaches are excellent for reconstructing free-form objects, but typically require precise precalibrated camera poses and sufficient variations (texture or shading) on the surfaces to achieve accurate reconstruction. On the other hand, when the objects (or scene) have discrete feature points (or lines), camera poses, calibration and the 3-D structures can be simultaneously recovered from the images via the well-established methods in multiple-view geometry [1,2].

Nevertheless, free-form natural or man-made objects with curved contours are ubiquitous in the real world, and very often their images do not have sufficient texture or feature points for us to apply conventional algorithms to accurately recover the surfaces and camera poses. So what can we do when we need to recover such curved shapes as well as camera poses from images without enough feature points to start with? In general, this is a very daunting task and in fact an impossible one unless certain constraints are imposed on either the shapes or the camera motions.

In this paper, we show that if the shape boundary consists of *symmetric curves*, the shape of the curves and the camera poses are strongly encoded in the perspective images

and can be recovered accurately from as few as two images – via the “epipolar geometry” of symmetric curves. All that is needed is the correspondence between entire curves, and no prior correspondences between points on the curves are required. Furthermore, if the symmetric curves in space are planar, both the shape of the curves and the camera pose can be recovered uniquely from a single view.

Relation to prior work. While multiple-view geometry of points, lines, and planes have been extensively studied and well-understood, recent studies have gradually turned to use curves and surfaces as basic geometric primitives for modeling and reconstructing 3-D shapes. The difficulty in reconstruction of curves is that the point correspondences between curves are not directly available from the images because there are no distinct features on curves except the endpoints. An algorithm in [3] was proposed to automatically match individual curves between images using both photometric and geometric information. The techniques introduced in [4,5] aimed to recover the motion and structure for arbitrary curves from monocular sequences of images. It was realized that it is not possible to uniquely and accurately reconstruct both the shape and camera poses for an arbitrary 3-D curve from an arbitrary set of views. Therefore, some restrictions need to be put on the views and curves themselves to guarantee a unique reconstruction. It was shown that, under circular motions, curved objects can be recovered from their silhouettes or contours [6,7,8,9]. The algorithm given in [10] can reconstruct symmetric shapes made of generalized cylinders. Reconstruction of curves from multiple views based on affine shape method was studied in [11,12]. The reconstruction of algebraic curves from multiple views has been proposed by [13,14].

This paper introduces symmetry as a very effective constraint to solve the reconstruction problem for 3-D curves. Such methods are very useful in practice since symmetric curves are ubiquitous in a wide range of natural and man-made scenes (e.g., leaves, signs, containers). Symmetry has long been exploited as a very effective cue in feature-based reconstruction methods [15,16,17,18,19]. Our work generalizes such methods to feature-less smooth curves.

Contribution of this paper. In this paper, we propose a novel approach for the reconstruction of curves in space and the recovery of camera poses by imposing global symmetry constraints on the original shapes of the curves. As part of the derivation, we establish the precise conditions and minimal number of views required for a unique solution: a) there is always a two-parameter family of ambiguous solutions in reconstructing general symmetric curves from a single view; b) nevertheless if the curves are planar (in space), the solution becomes unique; c) for general symmetric curves, two generic views are sufficient to give a unique solution (summarized in Table 1).

2 Symmetric Curve Representation

2.1 Perspective Image of a Curve

A parameterized curve $\gamma(t)$ in \mathbb{R}^n with parameter $t \in [t_a, t_b]$ is a continuous map

$$\gamma(\cdot) : t \mapsto \gamma(t) \in \mathbb{R}^n, \quad t \in [t_a, t_b]. \quad (1)$$

A curve γ is an equivalence class of parameterized curves because a curve can be arbitrarily parameterized. Two parameterized curves γ_1, γ_2 are said to be equivalent if there exists a continuous, monotonically increasing reparameterization function $\sigma : t \mapsto t'$ such that

$$\gamma_1(t) = \gamma_2(\sigma(t)). \quad (2)$$

The image of a 3-D parameterized curve $\Gamma(t)$ in \mathbb{R}^3 , $t \in [t_a, t_b]$ taken at $g_0 = (R_0, T_0)$ is a 2-D parameterized curve $\gamma(s)$ in \mathbb{R}^2 with parameter $s \in [s_a, s_b]$. $s = \sigma(t)$ is a reparameterization of t . If the camera is calibrated, the image curve $\gamma(s)$ in homogeneous image coordinates and the space curve $\Gamma(t)$ in spatial coordinates are related by

$$\lambda(s)\gamma(s) = \Pi_0 g_0 \Gamma(t), \quad (3)$$

where $s = \sigma(t)$, $s_a = \sigma(t_a)$, $s_b = \sigma(t_b)$, and $\Pi_0 = [I, 0]$ is the standard projection matrix. The parameter s, t may be the same, i.e. the reparameterization σ can be an identity function, which however we do not assume to know at this point.

2.2 Image of Symmetric Curves

Now we consider a pair of curves Γ, Γ' that are reflectively symmetric to each other with respect to a central plane P_r .¹ Without loss of generality, we assume that the two curves are symmetric with respect to the yz -plane of a selected canonical coordinate frame. The reflection can be represented by a Euclidean motion $g_r = (R_r, 0)$, where

$$R_r \doteq \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in O(3) \subset \mathbb{R}^{3 \times 3}. \quad (4)$$

Then $\Gamma'(t) = g_r \Gamma(t)$. If one image of the symmetric curves is taken at $g_0 = (R_0, T_0)$, the images of the two curves are:

$$\lambda(s)\gamma(s) = \Pi_0 g_0 \Gamma(t), \quad \lambda'(s')\gamma'(s') = \Pi_0 g_0 g_r \Gamma(t), \quad (5)$$

where $s = \sigma(t)$, $s_a = \sigma(t_a)$, $s_b = \sigma(t_b)$ and $s' = \sigma'(t)$, $s'_a = \sigma'(t_a)$, $s'_b = \sigma'(t_b)$.

2.3 Corresponding Points and Epipolar Geometry of Symmetric Curves

Notice that we can rewrite the equation for the image of the second curve as

$$\lambda'(s')\gamma'(s') = \Pi_0 g_0 g_r g_0^{-1}(g_0 \Gamma(t)).$$

Therefore, the image of the two symmetric curves can be interpreted as two images of the same curve taken with a relative camera pose

$$g_0 g_r g_0^{-1} = (R, T) \doteq (R_0 R_r R_0^T, R_0 R_r T_0).$$

¹ In this paper, we work primarily with reflective symmetry. But the same method, with some modification, can be applied to curves with translational or rotational symmetry.

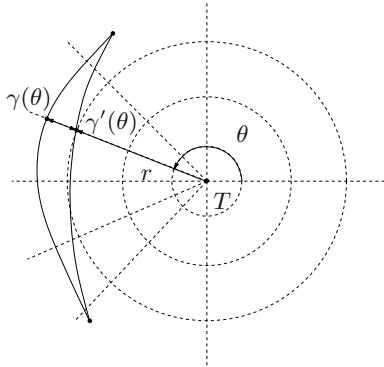


Fig. 1. Corresponding points can be easily obtained for a pair of symmetric curves in polar coordinates (r, θ) .

Under this interpretation, the two images $(\gamma(s), \gamma'(s'))$ of the pair of points $(\Gamma(t), \Gamma'(t))$ become the two images of a single point $\Gamma(t)$ in space taken from two different views. These corresponding image points should satisfy the epipolar constraint

$$\gamma'(s')^T \hat{T} R \gamma(s) = 0, \quad \forall s \in [s_a, s_b], s' \in [s'_a, s'_b], \sigma^{-1}(s) = \sigma'^{-1}(s'), \quad (6)$$

where we use \hat{T} to indicate the skew-symmetric matrix associated to T such that $\hat{T}v = T \times v$ for all $v \in \mathbb{R}^3$. From the definition of R_r , R and T , it is not difficult to show that $\hat{T}R = \hat{T}$. Hence the above epipolar constraint is simplified to:

$$\gamma'(s')^T \hat{T} \gamma(s) = \gamma'(s')^T \widehat{\gamma(s)} T = 0, \quad \sigma^{-1}(s) = \sigma'^{-1}(s'). \quad (7)$$

We call T the “vanishing point” for the pair of symmetric curves since it is parallel to the line defined by each pair of corresponding points $(\Gamma(t), \Gamma'(t))$.

The two reparameterization $\sigma(\cdot)$ and $\sigma'(\cdot)$ can be made the same in the following *polar coordinates* for the pair of curves. If the vanishing point $T \in \mathbb{R}^3$ is known², the intersection of the two image curves γ and γ' with any ray through T are two corresponding image points³. The angle of the ray θ and the distance to the vanishing point T establish the polar coordinates. The angle θ becomes a suitable parameter for the two curves so that $\gamma(\theta)$ and $\gamma'(\theta)$ are always two corresponding symmetric points. The epipolar constraint in polar coordinates becomes

$$\gamma'(\theta)^T \widehat{\gamma(\theta)} T = 0, \quad \forall \theta \in [\theta_a, \theta_b]. \quad (8)$$

² T is in homogeneous coordinates and $\|T\| = 1$.

³ Using polar coordinates is convenient when there is only one intersection between a ray and an image curve. However, when there are multiple intersections, the correspondences among the intersections with a known T should be set up as follows. Suppose the sequence of intersections with curve γ is $\{p_0, p_1, \dots, p_m\}$, and the sequence of intersections with curve γ' is $\{q_0, q_1, \dots, q_n\}$. Then $m = n$, and p_i corresponds to q_{m-i} .

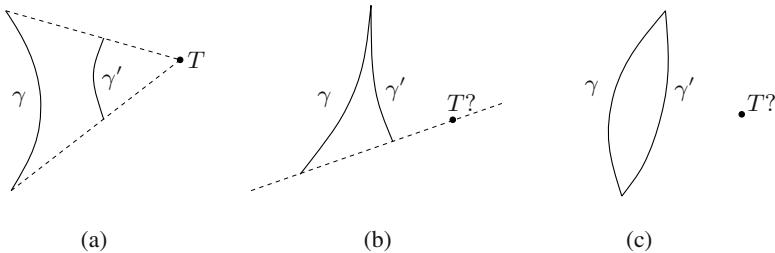


Fig. 2. (a) T is uniquely determined by the two pairs of end points; (b) T is determined by one pair of end points up to a line; (c) T is a two-parameter family.

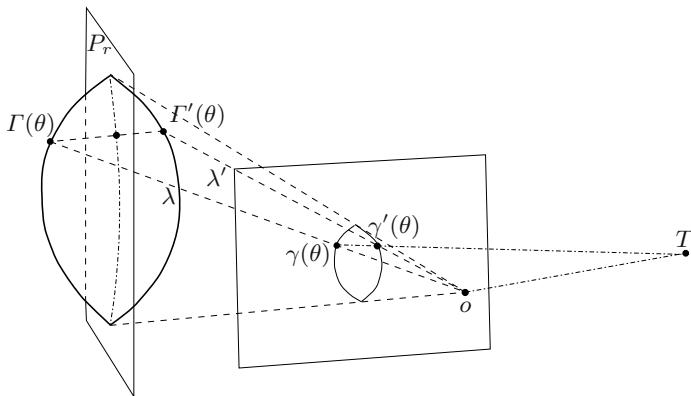


Fig. 3. If the vanishing point T is given, the 3-D depths for each pair of corresponding points in image can be uniquely determined.

3 Reconstruction of Symmetric Curves from Images

3.1 Ambiguity in Reconstruction of Symmetric Curves from a Single View

There are essentially three different cases for the image of a pair of symmetric curves that we illustrate in Figure 2. When the two pairs of end points of γ and γ' are separate as in case (a), the vanishing point T is uniquely determined and so is the correspondence between points on the two image curves. In the remaining cases (b) and (c), extra constraints need to be imposed on the symmetric curves sought in order to have a unique solution. We will focus on case (c) which is the most general case.

As illustrated in Figure 3, if the actual vanishing point T is given, the 3-D depths for each pair of corresponding points in image can be uniquely determined via the following “triangulation” equation [15]:

$$\begin{bmatrix} \widehat{T}\gamma(\theta) & -\widehat{T}\gamma'(\theta) \\ T^T\gamma(\theta) & T^T\gamma'(\theta) \end{bmatrix} \begin{bmatrix} \lambda(\theta) \\ \lambda'(\theta) \end{bmatrix} = \begin{bmatrix} 0_{3 \times 1} \\ 2d \end{bmatrix}, \quad (9)$$

where d is the distance from the camera center to the symmetry plane P_r . The first row of this equation means that T is the vanishing point of the family of parallel lines between

$\Gamma(\theta)$ and $\Gamma'(\theta)$. The line oT is parallel to the family of lines $\Gamma(\theta)\Gamma'(\theta)$ in 3-D. The second row of the equation implies that the distances from the points $\Gamma(\theta)$, $\Gamma'(\theta)$ to the symmetry plane P_r are equal. However, if the correct T is not known, the above equation always has a unique solution for $\lambda(\theta)$ and $\lambda'(\theta)$ for an arbitrarily chosen T . Furthermore the recovered 3-D curves $\Gamma(\theta) = \lambda(\theta)\gamma(\theta)$ and $\Gamma'(\theta) = \lambda'(\theta)\gamma'(\theta)$ are indeed symmetric. We hence have proven the following lemma:

Lemma 1. *Given a pair of curves γ and γ' on the image plane and an arbitrary feasible vanishing point T , there exists a pair of curves in space Γ and Γ' such that γ and γ' are their images, respectively.*

Lemma 1 states an unfortunate fact: for the pair of 2-D image curves shown in Figure 1, almost any choice of the vanishing point T results in a valid 3-D interpretation of the curves as the image of some pair of symmetric curves in 3-D. Therefore, for case (c) of Figure 2, there is a two-parameter family of pairs of symmetric curves in 3-D that give rise to the same pair of image curves; for case (b), there is a one-parameter family.

3.2 Reconstruction of Planar Symmetric Curves from a Single View

From the above discussions, the reconstruction of a pair of general symmetric curves from a single view is in general an *ill-posed* problem,⁴ unless some additional conditions are imposed upon the class of curves of interest. Most symmetric curves in practice are planar curves, and we first examine if this additional information may lead to a unique solution.

For the pair of planar symmetric curves in case (c) of Figure 2, the central line l_c of the curves in the image is determined by connecting the two end points.⁵ As before, the true vanishing point T leads to a correspondence of the two curves. Let $\gamma(\theta)$, $\gamma'(\theta)$ and $l_c(\theta)$ be corresponding points on the two curves and the central line. Also, l_c should lie on the central plane P_r . With the additional planar constraints, Equation (9) gives rise to

$$\begin{bmatrix} \widehat{T}\gamma(\theta) & -\widehat{T}\gamma'(\theta) & 0_{3 \times 1} \\ T^T\gamma(\theta) & T^T\gamma'(\theta) & 0 \\ \gamma(\theta) & \gamma'(\theta) & -2l_c(\theta) \end{bmatrix} \begin{bmatrix} \lambda(\theta) \\ \lambda'(\theta) \\ \lambda_c(\theta) \end{bmatrix} = \begin{bmatrix} 0_{3 \times 1} \\ 2d \\ 0_{3 \times 1} \end{bmatrix}, \quad \forall \theta \in [\theta_a, \theta_b]. \quad (10)$$

After eliminating $\lambda_c(\theta)$ by multiplying $\widehat{l}_c(\theta)$ on both sides of the third row, the equation becomes

$$\begin{bmatrix} \widehat{T}\gamma(\theta) & -\widehat{T}\gamma'(\theta) \\ T^T\gamma(\theta) & T^T\gamma'(\theta) \\ \widehat{l}_c(\theta)\gamma(\theta) & \widehat{l}_c(\theta)\gamma'(\theta) \end{bmatrix} \begin{bmatrix} \lambda(\theta) \\ \lambda'(\theta) \end{bmatrix} = \begin{bmatrix} 0_{3 \times 1} \\ 2d \\ 0_{3 \times 1} \end{bmatrix}, \quad \forall \theta \in [\theta_a, \theta_b]. \quad (11)$$

⁴ Except for the special case (a) of Figure 2.

⁵ For case (b), the image of the central line can also be determined from γ and γ' and we here do not elaborate due to the limit of space.

This equation can be rewritten as

$$\begin{bmatrix} \widehat{T}\gamma(\theta) & -\widehat{T}\gamma'(\theta) & 0_{3 \times 1} \\ T^T\gamma(\theta) & T^T\gamma'(\theta) & 2d \\ \widehat{l}_c(\theta)\gamma(\theta) & \widehat{l}_c(\theta)\gamma'(\theta) & 0_{3 \times 1} \end{bmatrix} \begin{bmatrix} \lambda(\theta) \\ \lambda'(\theta) \\ 1 \end{bmatrix} \doteq M(T, \theta)\Lambda(\theta) = [0_{7 \times 1}], \quad \forall \theta \in [\theta_a, \theta_b]. \quad (12)$$

The *necessary condition* for a valid solution of T for planar symmetric curves is

$$\text{rank}[M(T, \theta)] = 2, \quad \forall \theta \in [\theta_a, \theta_b]. \quad (13)$$

Since only the correct vanishing point T can satisfy the above rank condition, a criterion for finding the correct T is

$$T = \arg(\text{rank}[M(T, \theta)] = 2), \quad \forall \theta \in [\theta_a, \theta_b]. \quad (14)$$

Once T is found, the depth vector Λ can also be obtained as

$$\Lambda(\theta) = \text{null}(M(T, \theta)). \quad (15)$$

In practice, the rank condition will not be exactly satisfied by any T due to noise. We may choose $\Lambda(\theta)$ to be the eigenvector of $M(T, \theta)$ that is associated with the smallest eigenvalue. So we want to find the T such that $\int_{\theta_a}^{\theta_b} \|M(T, \theta)\Lambda(\theta)\|^2 d\theta$ is minimized. Let the singular value decomposition (SVD) of the matrix M be $M(T, \theta) = U(T, \theta)\Sigma(T, \theta)V(T, \theta)^T$,

$$T = \arg \min \left(\int_{\theta_a}^{\theta_b} \|M(T, \theta)\Lambda(\theta)\|^2 d\theta \right) = \arg \min \left(\int_{\theta_a}^{\theta_b} \Sigma_{3,3}(T, \theta)^2 d\theta \right), \quad (16)$$

where $\Sigma_{3,3}$ is the smallest singular value of M . Once T is found, the depth vector Λ is recovered as the third column of $V(T, \theta)$. In Section 4, we will show how this simple criterion gives accurate reconstruction of planar symmetric curves.

3.3 Reconstruction of General Symmetric Curves from Two Views

For a pair of general symmetric curves, according to Lemma 1, a single view is not enough for a unique recovery. In this subsection, we show that how an extra view may resolve this problem.

As an example, Figure 4 illustrates two images of a pair of (reflectively) symmetric curves. If T_1 and T_2 are the correct vanishing points in the two image planes, the images of the curves satisfy the two equations:

$$\begin{bmatrix} \widehat{T_1}\gamma_1(\theta_1) & -\widehat{T_1}\gamma'_1(\theta_1) \\ T_1^T\gamma_1(\theta_1) & T_1^T\gamma'_1(\theta_1) \end{bmatrix} \begin{bmatrix} \lambda_1(\theta_1) \\ \lambda'_1(\theta_1) \end{bmatrix} = \begin{bmatrix} 0_{3 \times 1} \\ 2d_1 \end{bmatrix}, \quad \forall \theta_1 \in [\theta_a, \theta_b]. \quad (17)$$

$$\begin{bmatrix} \widehat{T_2}\gamma_2(\theta_2) & -\widehat{T_2}\gamma'_2(\theta_2) \\ T_2^T\gamma_2(\theta_2) & T_2^T\gamma'_2(\theta_2) \end{bmatrix} \begin{bmatrix} \lambda_2(\theta_2) \\ \lambda'_2(\theta_2) \end{bmatrix} = \begin{bmatrix} 0_{3 \times 1} \\ 2d_2 \end{bmatrix}, \quad \forall \theta_2 \in [\theta_a, \theta_b].$$

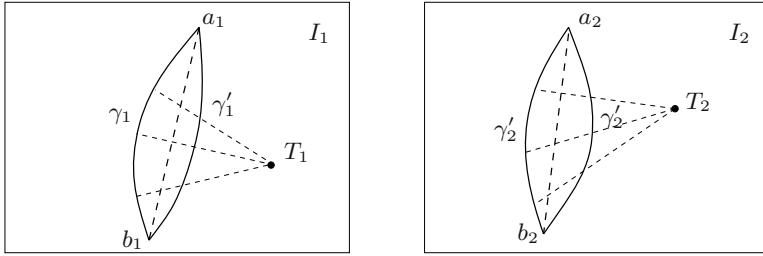


Fig. 4. Two images of a pair of symmetric curves. Assume that (R_{21}, T_{21}) is the relative motion from the first view to the second. T_1 and T_2 are the vanishing points of the two images, respectively.

For simplicity, we typically choose $d_1 = d_2 = 1/2$ at this moment. The correct relative scale between d_1 and d_2 can be determined at a later stage. Given any two vanishing points T_1 and T_2 , from Lemma 1, the above triangulation equations in general have solutions for $[\lambda_1(\theta_1), \lambda_1(\theta_1)']$ and $[\lambda_2(\theta_2), \lambda_2'(\theta_2)]$. We thus obtain two pairs of symmetric 3-D curve $[\Gamma_1(\theta_1), \Gamma'_1(\theta_1)]$ and $[\Gamma_2(\theta_2), \Gamma'_2(\theta_2)]$ via triangulation. $[\Gamma_1(\theta_1), \Gamma'_1(\theta_1)]$ and $[\Gamma_2(\theta_2), \Gamma'_2(\theta_2)]$ are in the two camera coordinates of the two views. If T_1 and T_2 are the correct vanishing points, the reconstructed curves are all related to the same pair of 3-D symmetric curves Γ and $\Gamma' = g_r \Gamma$ in space via the rigid transformations $g_{01} = (R_{01}, T_{01})$ and $g_{02} = (R_{02}, T_{02})$ (the two camera poses), respectively,

$$\begin{aligned}\Gamma_1(\theta_1) &= g_{01}\Gamma(\theta_1), & \Gamma'_1(\theta_1) &= g_{01}g_r\Gamma(\theta_1), \\ \Gamma_2(\theta_2) &= g_{02}\Gamma(\theta_2), & \Gamma'_2(\theta_2) &= g_{01}g_r\Gamma(\theta_2).\end{aligned}\quad (18)$$

The distance between the end points of the curves should be preserved under rigid transformations. So the correct ratio between d_1 and d_2 can be determined as

$$\frac{d_1}{d_2} = \frac{\|\Gamma_1(\theta_{a1}) - \Gamma_1(\theta_{b1})\|}{\|\Gamma_2(\theta_{a2}) - \Gamma_2(\theta_{b2})\|} = \frac{\|l_{c1}\|}{\|l_{c2}\|}. \quad (19)$$

With respect to each view, the canonical pose g_0 of the curves can be recovered from symmetry as follows. Because the vanishing point T is orthogonal to the central plane, T can be chosen as the x -axis of the canonical frame; the central line $l_c = \Gamma(\theta_a) - \Gamma(\theta_b)$ is chosen to be the y -axis since it is in the central plane; an endpoint, say for example $\Gamma(\theta_a)$, can be selected to be the origin of the canonical frame. Then the R_0 can be retrieved as

$$R_0 = \left[T, \quad \frac{l_c}{\|l_c\|}, \quad \widehat{T} \frac{l_c}{\|l_c\|} \right] \in \mathbb{R}^{3 \times 3}, \quad T_0 = \Gamma(\theta_a) \in \mathbb{R}^3. \quad (20)$$

From either Γ_1 or Γ_2 , the 3-D curves in the canonical frame, Γ_{01} or Γ_{02} respectively, can be recovered. According to the Lemma 1, all possible T_1 and T_2 in the two image planes can generate two sets of curves $\Gamma_{01}(T_1)$ and $\Gamma_{02}(T_2)$. If the vanishing points T_1 and T_2 are correct, Γ_{01} and Γ_{02} should be identical. Therefore, the true curve Γ is the intersection of the two sets. This gives the necessary condition for the correct vanishing points,

$[T_1, T_2] = \arg(\Gamma_{01}(T_1) = \Gamma_{02}(T_2)).$

(21)

Table 1. Ambiguity in reconstruction of symmetric curves: A single view is not enough for reconstruction of general symmetric curves, except for case (a) of Figure 2, but sufficient for planar symmetric curves. Two or more views are needed for reconstruction of generally shaped symmetric curves.

# of solutions	General curves			Planar curves
	case (a)	case (b)	case (c)	
One view	unique	1-family	2-family	unique
Two views	unique	unique	unique	unique

For real images with noise, the equality may not be achieved. The following optimization criterion can be used to find T_1, T_2 by minimizing

$$[T_1, T_2] = \arg \min (\text{distance}(\Gamma_{01}(T_1), \Gamma_{02}(T_2))). \quad (22)$$

There are many different choices in the distance between two curves. We use the simplest Euclidean distance, also known in functional analysis as the L^2 -distance:⁶

$$\text{distance}(\Gamma_1, \Gamma_2) \doteq \int_{t_a}^{t_b} \|\Gamma_1(t) - \Gamma_2(t)\|^2 dt, \quad (23)$$

where Γ_1 and Γ_2 are both parameterized by their arc length t . Notice that the above criterion is rather different from most curve reconstruction algorithms that are based on minimizing the reprojection error in the images via the notion of “bundle-adjustment” (e.g., see [11]). The above method minimizes the discrepancy in the shapes of the reconstructed 3-D curves in space. Furthermore, the method can be easily generalized if *multiple images* of the same curves are given.

The optimal T_1 and T_2 can be found via any nonlinear optimization scheme chosen at the user’s discretion.⁷ For case (b) of Figure 2, the vanishing points will lie on the line generated by the two separate end points. So the search for T_1 and T_2 is two-dimensional. For case (c) of Figure 2, each T_1 and T_2 is two dimensional, and therefore the search is in a four-dimensional space. For curves with general shapes, the solution is always unique.

To conclude Section 3, we summarize all cases of symmetric curves studied so far in Table 1, in terms of ambiguities in reconstruction from one or two views.

4 Experimental Results

4.1 Simulations

One view of planar curves. To test the performance of the proposed methods, we have conducted extensive simulations. In the first simulation, a pair of planar 3-D symmetric

⁶ We have also tried other distances such as L^1 -distance and C^1 -distance. They all give similar reconstruction results for the simulations and experiments conducted in this paper.

⁷ For the simulations and experiments given in this paper, we used the simple MATLAB function “fminsearch.” We observed standard convergence rate from such an off-the-shelf nonlinear optimization toolbox. The convergence rate may be improved with a custom-designed algorithm and initialization.

Table 2. The error of the shape and camera pose as a function of the view angle α between the camera axis and the central plane. It indicates that the shape error and the camera pose error in general increase when the angle α is increasing.

view angle α (degree)	10	20	30	40	50	60	70	80
shape error (L^2 -distance)	0.0170	0.0168	0.0223	0.0264	0.0271	0.0320	0.0361	0.0375
camera pose error (degree)	0.2529	0.5571	0.5090	0.3193	0.3275	0.3916	0.5066	1.2401

Table 3. The error of the shape and camera pose as a function of the relative view angle α' between the two camera axes. It indicates that the shape error and the camera pose error in general decrease when the angle α' increases.

relative view angle α' (degree)	10	20	30	40	50	60
shape error (L^2 -distance)	0.0253	0.0228	0.0264	0.0221	0.0245	0.0203
average camera pose error (degree)	3.8600	4.4582	3.7642	3.2266	3.8470	3.4047

curves in case (c) of Figure 2 are generated. A perspective image of their curves is obtained from a pin-hole camera model. In order to test the robustness of our algorithm, 5% asymmetry is added onto the 3-D symmetric curves⁸ and white Gaussian noise is added to the projected image curves with standard deviation σ . The added noise corresponds to approximately one pixel in standard deviation for a 400x320 pixel image. A large variety of view points are tested. From the simulations, we found that the view angle α between the camera optical axis and the central plane P_r is the most important factor for the accuracy in reconstruction. The Table 2 shows the error as a function of the angle α . Only the angles from $+10^\circ$ to $+70^\circ$ are tested because the negative side will give similar results due to symmetry. The shape error is the L^2 -distance between the curves reconstructed and the ground truth.⁹ The camera pose error is the angle (in degrees) between the original camera rotation matrix and the rotation matrix reconstructed. The results indicate that the shape error and the camera pose error increase with an increasing angle α . However, even in the worst case, the errors remain very small, which indicates that our method is quite effective.

Two views of general curves. In the second simulation, a pair of non-planar 3-D curves in case (c) of Figure 2 is generated, and two images are obtained from two view points. A large variety of view points are tested. It is discovered from the simulation results that the relative view angle $\alpha' = |\alpha_1 - \alpha_2|$ (difference in the angles between the two camera axes and the central plane) is the most important factor. Only the angles from 10° to 60° are tested because other angles will result in similar results due to symmetry. Table 3 shows the error as a function of the angle α' . The shape error is the distance between the curves reconstructed from the noisy images and the ground truth. The results indicate that the shape error and the camera pose error in general decrease with the increasing of the relative view angle α' . However, all of these errors remain small.

⁸ We make the curves slightly asymmetric by adding to the curves deformation of a magnitude up to 5% of the maximum distance between the two curves.

⁹ The length of the curves is always normalized to be one for comparison.

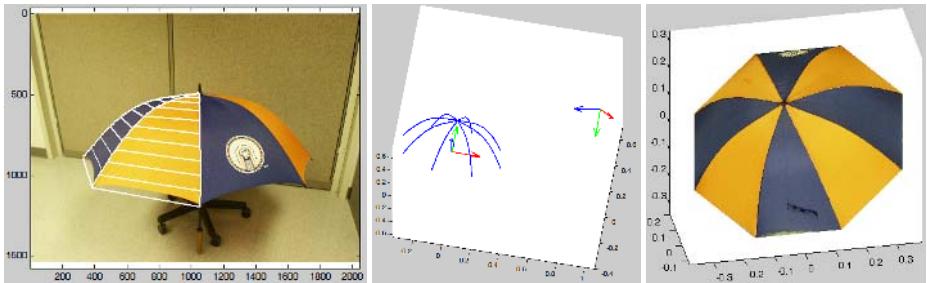


Fig. 5. Left: A single image that is used to recover the whole umbrella. The vanishing lines obtained from the optimization are shown in the image. Middle: The frame of the umbrella recovered from two stripes. Right: A synthetically rendered view of the completely reconstructed umbrella from the top.

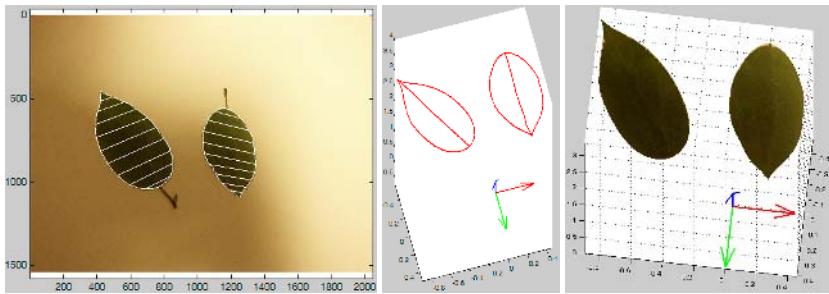


Fig. 6. Left: A single image that is used to recover the leaves. The vanishing lines obtained from the optimization are shown in the image. Middle: The shape of the leaf boundaries and the camera pose recovered from the image. Right: A synthetically rendered image of the reconstructed leaves.

4.2 Experiments on Real Images

Figure 5 shows an example of a reconstructed umbrella. This example belongs to the case (b) of Figure 2. The whole umbrella can be reconstructed from a single image using the two-view method because the stripes on the umbrella are all identical and two stripes in one view can be treated as two views of the same stripe.

Figure 6 shows an example of leaves whose contours can be considered as planar curves, which is in the category of Figure 2 (c). The recovered structures as well as a synthetically rendered image of the reconstructed leaves are shown. This experiment verifies that from only one single view, the structure of symmetric planar curves can be recovered accurately.

Figure 7 shows a reconstruction of a 3-D leaf from two views. It is an example of general curves in the category of Figure 2 (c). The recovered structure as well as a synthetically rendered image of the reconstructed leaf are shown. We can see that the shape of the leaf has been convincingly recovered.

On a Pentium III 866MHz computer with MATLAB 6.0, the algorithm completes in 5 minutes for the one-view examples and 10 minutes for the two-view examples.

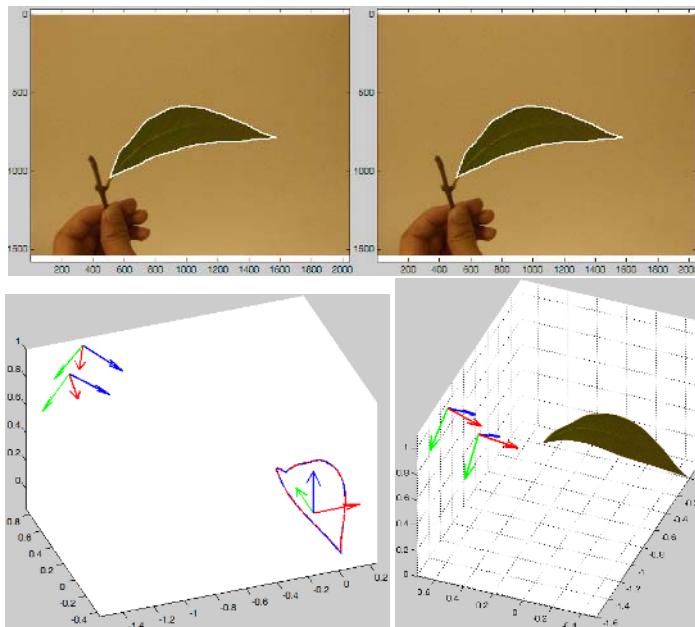


Fig. 7. Top: Two images that are used to recover a nonplanar leaf. Bottom Left: The shape of the leaf boundary and camera poses recovered from the two views (note that the difference between the two recovered boundaries is almost indistinguishable). Bottom Right: A synthetically rendered image of the reconstructed leaf.

5 Conclusions and Future Work

In this paper, we have provided simple and effective algorithms for the simultaneous reconstruction of both the shape of smooth symmetric curves and the camera poses from as few as one or two images without feature correspondences to start with. Both simulations and experiments show that the results are remarkably accurate. In the future, we plan to combine our methods with surface techniques for symmetric shape reconstruction. We will also study the effects of various deformations on the reconstruction of such shapes.

References

1. Faugeras, O.: Three-Dimensional Computer Vision, A geometric approach. MIT Press (1993)
2. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2003)
3. Schmid, C., Zisserman, A.: The geometry and matching of curves in multiple views. In: European Conference on Computer Vision. (1998)
4. Papadopoulou, T., Faugeras, O.: Computing structure and motion of general 3D curves from monocular sequences of perspective images. In: European Conference on Computer Vision. (1996)

5. Papadopoulo, T., Faugeras, O.: Computing structure and motion of general 3D curves from monocular sequences of perspective images. Technical Report 2765, INRIA (1995)
6. Wong, K.Y.K., Cipolla, R.: Structure and motion from silhouettes. In: 8th IEEE International Conference on Computer Vision. Volume II., Vancouver, Canada (2001) 217–222
7. Mendonça, P.R.S., Wong, K.Y.K., Cipolla, R.: Camera pose estimation and reconstruction from image profiles under circular motion. In: European Conference on Computer Vision. Volume II. (2000) 864–877
8. Cipolla, R., Blake, A.: Surface shape from the deformation of apparent contours. In: IEEE International Conference on Computer Vision. (1992) 83–112
9. G. Cross, A.W.F., Zisserman, A.: Parallax geometry of smooth surfaces in multiple views. In: IEEE International Conference on Computer Vision. (1999) 323–329
10. François, A., Medioni, G.G.: A human-assisted system to build 3-D models from a single image. In: IEEE International Conference on Multimedia Computing and Systems. (1999) 282–288
11. Berthilsson, R., Astrom, K., Heyden, A.: Reconstruction of curves in \mathbb{R}^3 , using factorization and bundle adjustment. In: IEEE International Conference on Computer Vision. (1999)
12. Berthilsson, R., Astrom, K.: Reconstruction of 3D-curves from 2D-images using affine shape methods for curves. In: International Conference on Computer Vision and Pattern Recognition. (1997)
13. Kaminski, J., M.Fryers, Shashua, A., Teicher, M.: Multiple view geometry of non-planar algebraic curves. In: IEEE International Conference on Computer Vision. (2001)
14. Kaminski, J., Shashua, A.: On calibration and reconstruction from planar curves. In: European Conference on Computer Vision. (2000)
15. Hong, W.: Geometry and reconstruction from spatial symmetry. Master Thesis, UIUC (2003)
16. Hong, W., Yang, A.Y., Ma, Y.: On symmetry: Structure, pose and calibration from a single image. International Journal on Computer Vision (Submitted 2002)
17. Mitsumoto, H., Tamura, S., Okazaki, K., Fukui, Y.: 3-D reconstruction using mirror images based on a plane symmetry recovering method. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 941–946
18. Zabrodsky, H., Weinshall, D.: Using bilateral symmetry to improve 3D reconstruction from image sequences. Computer Vision and Image Understanding **67** (1997) 48–57
19. Zabrodsky, H., Peleg, S., Avnir, D.: Symmetry as a continuous feature. IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995) 1154–1166

A Topology Preserving Non-rigid Registration Method Using a Symmetric Similarity Function-Application to 3-D Brain Images

Vincent Noblet^{1,2}, Christian Heinrich¹,
Fabrice Heitz¹, and Jean-Paul Armspach²

¹ Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection,
LSIIT, UMR CNRS-ULP 7005,

Bd Sébastien Brant, BP 10413, F-67412 Illkirch Cedex, France.

{noblet,heinrich,heitz}@lsit.u-strasbg.fr

² Institut de Physique Biologique, Faculté de médecine, UMR CNRS-ULP 7004,
4 Rue Kirschleger, F-67085 Strasbourg Cedex, France.

armspach@ipb.u-strasbg.fr

Abstract. 3-D non-rigid brain image registration aims at estimating consistently long-distance and highly nonlinear deformations corresponding to anatomical variability between individuals. A consistent mapping is expected to preserve the integrity of warped structures and not to be dependent on the arbitrary choice of a reference image: the estimated transformation from A to B should be equal to the inverse transformation from B to A. This paper addresses these two issues in the context of a hierarchical parametric modeling of the mapping, based on B-spline functions. The parameters of the model are estimated by minimizing a symmetric form of the standard sum of squared differences criterion. Topology preservation is ensured by constraining the Jacobian of the transformation to remain positive on the whole continuous domain of the image as a non trivial 3-D extension of a previous work [1] dealing with the 2-D case. Results on synthetic and real-world data are shown to illustrate the contribution of preserving topology and using a symmetric similarity function.

1 Introduction

Deformable – inter-subject – registration of 3-D medical images has received considerable attention during the last decade, as a key step for the construction and use of individualized or probabilistic anatomical atlases [2]. In this context, the goal of non-rigid image registration is to estimate the long-distance and highly nonlinear deformations corresponding to anatomical variability between individuals. To ensure the consistency of registration, the mapping should be continuous one-to-one in order to preserve the integrity of warped structures. This property, which also named topology preservation, is enforced by the positivity of the Jacobian of the transformation: it ensures that connected structures remain connected and that the neighborhood relationships between structures

is maintained. It also prevents the disappearance or appearance of existing or new structures. As often noticed, the topology preservation assumption may not be valid in pathological cases, for example when registering brain images with tumor appearance or when registering images before and after surgery. Although in these cases no homeomorphic transformation exists between the two images, we show here that a topology preserving registration algorithm may be used with profit to detect topology violation. Another desirable property of the registration is that the estimated transformation from image A to B should equal the inverse of the transformation from B to A. When using a standard asymmetric cost function such as the sum of squared differences (SSD), this property is seldom fulfilled, and the resulting estimated transformation depends on the arbitrary choice of the reference image.

In [3], Cachier explains why most non-rigid registration methods are asymmetric and proposes some inversion-invariant energy to symmetrize the registration problem. Ashburner also raised the problem of symmetrization in [4], but the symmetrization concerns only the prior of the Bayesian model and not the similarity criterion. Thirion [5] forces the symmetry of the registration by computing the direct deformation T_{12} (from image 1 to image 2), the reverse deformation T_{21} (from image 2 to image 1) and the residual deformation $R = T_{21} \circ T_{12}$, and then by redistributing equally the residual deformation between T_{12} and T_{21} to obtain $R \simeq Id$. By this way, the bijectivity of the deformation is enforced, but preservation of the topology on the continuous image domain is not ensured. Christensen [6] presents a consistent registration scheme by jointly estimating the forward and reverse transformations, constraining them to be inverse of each other, and restricting them to preserve topology by constraining them to obey the laws of continuum mechanics. In this paper, we present an alternative to Christensen's non-parametric continuum model, enabling fast symmetric and topology preserving registration of 3-D images on the continuous image domain. A typical warp on 128^3 images, with "good" accuracy, is computed in less than 30 minutes on a standard 2.4 GHz PC workstation. Our approach is based on a 3-D hierarchical parametric model of the deformation field using B-spline representations. The parameters of the model are estimated by minimizing a symmetric form of the SSD criterion, under the constraint of positivity of the Jacobian. This paper is a non-trivial extension of [1], which addresses the case of 2-D topology preserving mappings using asymmetric cost functions.

This paper is organized as follows. In section 2, we present the hierarchical parametric deformation model, and we detail the optimization method used to minimize a symmetric similarity criterion, subject to the positivity constraint on the Jacobian. In section 3, we illustrate the contribution of topology preservation and of a symmetric objective function on synthetic and real-world data.

2 Registration Method

In this section, we first present the parametric hierarchical deformation model. Then we deal with the mathematical issue of topology preservation and we introduce the symmetric objective function. Finally the optimization strategy for

minimizing the symmetric objective function under the constraint of topology preservation is described.

2.1 The Parametric Hierarchical Deformation Model

Let us consider two images $I_{source}(\mathbf{s})$ and $I_{target}(\mathbf{s})$ ($\mathbf{s} = [x, y, z]^t$), both defined on $\Omega \subset \mathbb{R}^3$. The goal is to find the mapping $\mathbf{h}(\mathbf{s}) = \mathbf{s} + \mathbf{u}(\mathbf{s}) \in \mathcal{H}$, where \mathcal{H} is the Hilbert space of finite energy mappings and \mathbf{u} the associated displacement vector field, so that the warped image $I_{source}(\mathbf{h}(\mathbf{s}))$ is as close as possible to the target image $I_{target}(\mathbf{s})$ in the sense of some similarity criterion $E(\mathbf{h})$. The vector field \mathbf{u} is parameterized at different scales using a decomposition over a sequence of nested subspaces $V_0 \subset V_1 \subset \dots \subset V_l \subset V_{l+1} \subset \dots \subset \mathcal{H}$, defining a multiresolution approximation of \mathbf{u} [7,8]. Let Φ be a scaling function. At scale l (*i.e.*, in space V_l), the displacement field \mathbf{u} is parameterized by the vector of coordinates $\mathbf{a}^l = \{a_{x;i,j,k}^l; a_{y;i,j,k}^l; a_{z;i,j,k}^l\}$ as:

$$\mathbf{u}^l(x, y, z) = \begin{bmatrix} u_x^l(x, y, z) \\ u_y^l(x, y, z) \\ u_z^l(x, y, z) \end{bmatrix} = \begin{bmatrix} \sum_{i,j,k} a_{x;i,j,k}^l \Phi_{i,j,k}^l(x, y, z) \\ \sum_{i,j,k} a_{y;i,j,k}^l \Phi_{i,j,k}^l(x, y, z) \\ \sum_{i,j,k} a_{z;i,j,k}^l \Phi_{i,j,k}^l(x, y, z) \end{bmatrix}, \quad (1)$$

where

$$\Phi_{i,j,k}^l(x, y, z) = 2^{3l/2} \Phi(2^l x - i) \Phi(2^l y - j) \Phi(2^l z - k). \quad (2)$$

Only first degree polynomial spline scaling functions Φ will be addressed in this paper, but the method may easily be extended to higher degree B-spline functions [9]. This multiresolution model allows to estimate the deformation field \mathbf{u} in a coarse-to-fine approach. The cost function $E(\mathbf{h})$ is minimized over each space V_l with respect to parameters \mathbf{a}^l , using as an initialization the parameters \mathbf{a}^{l-1} estimated at the previous scale.

2.2 Topology Preservation Enforcement

As already stated, a desirable property of inter-subject medical image warping is the preservation of the topology of anatomical structures. By enforcing this constraint, the space of possible solutions is restricted to deformations satisfying the real-world property of matter.

Topology preservation is related to the continuity and invertibility of the transformation, which should be a homeomorphism, *i.e.* a one-to-one mapping. This property is enforced by the positivity of the Jacobian of the transformation. In a more general way, we will enforce the Jacobian to be bracketed between two user-defined bounds J_m and J_M on the whole continuous domain of the image, *i.e.*:

$$\forall [x, y, z]^t \in \Omega \quad J(x, y, z) \in [J_m, J_M]. \quad (3)$$

In this framework, topology preservation becomes a particular case by setting $J_m = 0$ and $J_M = +\infty$.

Let $\Omega_{i,j,k}^l$ denote the 3-D support of the first degree spline scaling function $\Phi_{i,j,k}^l(x, y, z)$. $\Omega_{i,j,k}^l$ is partitioned into eight identical subboxes denoted $S_{p,q,r}^l$, where $p \in \{i-1; i\}; q \in \{j-1; j\}; r \in \{k-1; k\}$. The following expression of the Jacobian may be obtained on each $S_{p,q,r}^l$ (the reader is referred to [9] for details):

$$\begin{aligned} J(x, y, z, \boldsymbol{\alpha}) = & \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z + \alpha_5 x^2 + \alpha_6 x y + \alpha_7 x z \\ & + \alpha_8 y^2 + \alpha_9 y z + \alpha_{10} z^2 + \alpha_{11} x^2 y + \alpha_{12} x^2 z + \alpha_{13} x y^2 \\ & + \alpha_{14} x y z + \alpha_{15} x z^2 + \alpha_{16} y^2 z + \alpha_{17} y z^2 + \alpha_{18} x^2 y z \\ & + \alpha_{19} x y^2 z + \alpha_{20} x y z^2, \end{aligned} \quad (4)$$

where $\boldsymbol{\alpha}$ is a $S_{p,q,r}^l$ -dependent function of the \mathbf{a}^l 's.

2.3 Definition of the Objective Function

A standard approach for estimating the parameters of the transformation addressing single modal non-rigid registration is to minimize the sum of squared differences (SSD) criterion:

$$E(\mathbf{h}) = \int_{\Omega} |I_{target}(\mathbf{s}) - I_{source}(\mathbf{h}(\mathbf{s}))|^2 d\mathbf{s}.$$

Using this similarity function requires that a reference (target) image be chosen since both images do not play symmetric roles. The choice of the reference is arbitrary. As a consequence, the estimated transformation from I_{target} to I_{source} will not be equal to the inverse of the transformation from I_{source} to I_{target} , which should be the case ideally. We introduce a modified form of the SSD criterion where both images play a symmetric role (see also [3]):

$$\begin{aligned} E_{sym}(\mathbf{h}) = & \frac{1}{2} \int_{\Omega} |I_{target}(\mathbf{s}) - I_{source}(\mathbf{h}(\mathbf{s}))|^2 d\mathbf{s} \\ & + \frac{1}{2} \int_{\Omega} |I_{target}(\mathbf{h}^{-1}(\mathbf{s})) - I_{source}(\mathbf{s})|^2 d\mathbf{s}. \end{aligned}$$

Thanks to the change of variable $\mathbf{v} = \mathbf{h}^{-1}(\mathbf{s})$ in the second integral term, and reminding that $d\mathbf{s} = |J(\mathbf{v})| d\mathbf{v}$ where J is the Jacobian of \mathbf{h} , we derive the following expression:

$$E_{sym}(\mathbf{h}) = \frac{1}{2} \int_{\Omega} (1 + |J(\mathbf{s})|) |I_{target}(\mathbf{s}) - I_{source}(\mathbf{h}(\mathbf{s}))|^2 d\mathbf{s}. \quad (5)$$

2.4 Optimization Strategy

The parameters \mathbf{a}^l of \mathbf{u} are updated blockwise in a Gauss-Seidel scheme: $[a_{x;i,j,k}^l; a_{y;i,j,k}^l; a_{z;i,j,k}^l]$, which affects only $\Omega_{i,j,k}^l$, are updated jointly. This strategy enables constraint (3) on the Jacobian to be handled at an acceptable computational cost. Thanks to our spline model, the partial derivatives of E with respect to the updated parameters may be expressed as:

$$\begin{aligned} \frac{\partial E_{sym}}{\partial a_{w;i,j,k}^l} &= \int_{\Omega} (1 + |J(\mathbf{s})|) (I_{source}(\mathbf{h}(\mathbf{s})) - I_{target}(\mathbf{s})) \left. \frac{\partial I_{source}(\mathbf{v})}{\partial w} \right|_{\mathbf{v}=\mathbf{h}(\mathbf{s})} \\ &\quad \frac{\partial h_w(\mathbf{s})}{\partial a_{w;i,j,k}^l} + \frac{1}{2} \frac{\partial |J(s)|}{\partial a_{w;i,j,k}^l} |I_{target}(\mathbf{s}) - I_{source}(\mathbf{h}(\mathbf{s}))|^2 d\mathbf{s} \end{aligned}$$

where w stands for x, y or z. A more intricate expression is also obtained for the Hessian components (not presented here for the sake of conciseness). Therefore we implemented two methods of optimization: gradient descent and the Levenberg-Marquardt algorithm [10].

In the case of a blockwise descent scheme, condition (3) needs only be checked on the box $\Omega_{i,j,k}^l$ since the coordinates to be modified do not affect $\Omega \setminus \Omega_{i,j,k}^l$. Considering that the blockwise descent takes place along direction \mathbf{d} (\mathbf{d} is a coordinate vector defined on the space $[a_{x;n_1,n_2,n_3}^l a_{y;n_1,n_2,n_3}^l a_{z;n_1,n_2,n_3}^l]$) with a step δ , the expression of the Jacobian (4) on each $S_{p,q,r}^l$ may be expressed as:

$$J(x, y, z, \delta) = A(x, y, z)\delta + B(x, y, z),$$

where $A(x, y, z)$ and $B(x, y, z)$ are polynomial forms similar to (4). As a consequence, for each point $[x, y, z] \in S_{p,q,r}^l$, the Jacobian is an affine function of δ . Notice that this remains true for higher order spline functions, up to the difference that $A(x, y, z)$ and $B(x, y, z)$ are higher order polynomials [9].

Let us define:

$$\begin{cases} J_m(\delta) = \inf_{[x,y,z]^t \in S_{p,q,r}^l} J(x, y, z, \delta), \\ J_M(\delta) = \sup_{[x,y,z]^t \in S_{p,q,r}^l} J(x, y, z, \delta). \end{cases}$$

It can easily be shown that $J_m(\delta)$ is concave and $J_M(\delta)$ is convex as the infimum and supremum of a set of affine functions. Moreover the Jacobian values $J_m(0)$ and $J_M(0)$ also match condition (3), since they are the result of a previous optimization step enforcing this very condition. Hence, the set of admissible values of δ on box $S_{p,q,r}^l$ will be bracketed between zero and an upper bound as shown on Fig. 1. For obtaining the maximum admissible step δ^+ along \mathbf{d} , we have to take the minimum of all bounds computed on each $S_{p,q,r}^l$. But computing the bound on $S_{p,q,r}^l$ at an acceptable computational burden is tricky as the Jacobian $J(x, y, z)$ has no nice property of convexity and may have several local minima and maxima. Interval analysis techniques provide a good way of quickly finding a bracketing of the global minimum and maximum of the Jacobian for a given step

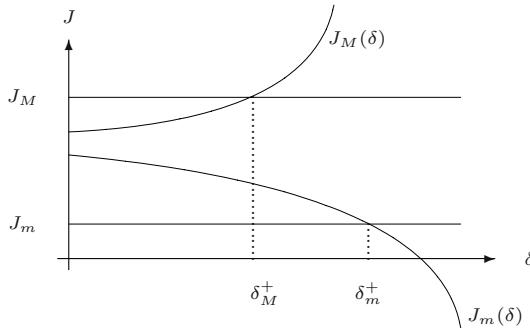


Fig. 1. The upper bound of δ on $S_{p,q,r}^l$ is computed as $\inf\{\delta_M^+, \delta_m^+\}$.

δ . So it becomes possible to quickly find a bracketing of the maximum admissible step δ^+ . We refer the reader to [11] for a general flavor of interval analysis and to [9] for more details on the algorithms leading to the determination of the upper bound δ^+ .

3 Results

In this section we present results on synthetic and real-world data aiming at illustrating the contribution of topology preservation and of using a symmetric similarity criterion with the prospect of detecting tumors. An assessment of registration accuracy is obtained on simulated deformation fields applied on real images.

3.1 Contribution of Topology Preservation

To highlight the contribution of topology preservation, we consider the warping of an anatomical atlas, which is one of the most robust methods for performing automatic segmentation of anatomical structures. Atlas-based segmentation consists in registering a patient MRI towards a reference MRI, which is itself associated to a 3-D reference segmentation map (the atlas). The atlas is then warped on the patient data, using the estimated deformable mapping. On Fig. 2, we present the matching between two 3-D 128³ MR brain images focusing our attention on segmented ventricles. Without topology preservation, a tearing¹ of the ventricle can be observed when the estimated deformation field is applied to the reference atlas segmentation map (Fig. 2(e)). Enforcing the Jacobian to be positive is a way of preserving the integrity of anatomical structures during the registration process (Fig. 2(f)).

¹ In fact, no tearing may happen with the transformation at hand since this deformation is necessarily continuous. Only folding is involved when topology is violated. Folding has the same visual effect as tearing.

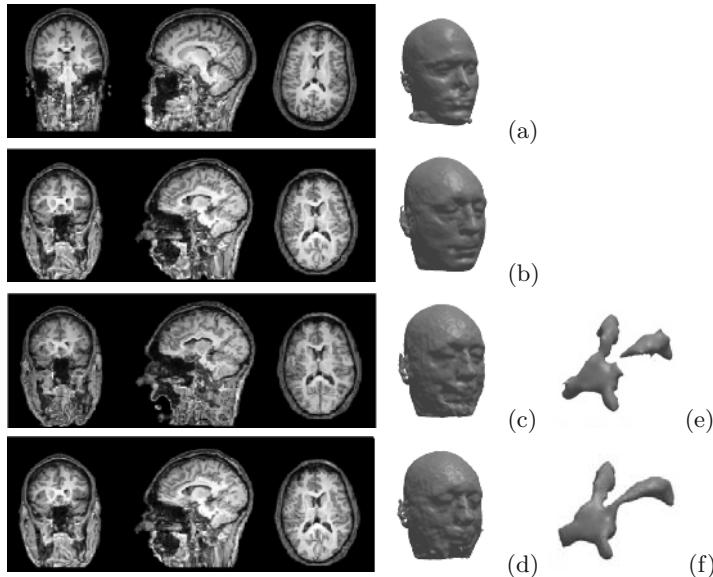


Fig. 2. 3-D non-rigid matching between two different patients and application to atlas-based ventricle segmentation: (a) source image (atlas); (b) target image (patient); (c) result of matching without any constraint; (d) result of matching with the positivity constraint $J > 0$; (e) warping of the ventricle from the atlas onto the target image (no positivity constraint); (f) warping of the ventricle from the atlas onto the target image (with positivity constraint).

3.2 Contribution of a Symmetric Similarity Criterion

The contribution of using a symmetric similarity criterion is illustrated here on a 3-D toy example involving a plain cube (Fig. 3(b)) and a second cube with a spherical hole inside (Fig. 3(a)). The goal is to show the behavior of topology preserving matching using either asymmetric or symmetric similarity criteria, when registering two volumes which are not topologically equivalent.

Let us first consider the case of registering Fig. 3(a) on Fig. 3(b). Whatever the cost function chosen (symmetric or asymmetric), the topology preserving matching algorithm tends to shrink the hole to one point, as illustrated on Fig. 3(c), which shows the effect of the deformation field applied to a synthetic grid. This shrinking of the hole yields high values of the Jacobian of the estimated transformation. This seems to be in contradiction with what we expect since a Jacobian lower (resp. higher) than 1 represents a contraction (resp. dilatation). But in the formalism presented in this paper, the warped image is obtained as $I_{warped}(\mathbf{s}) = I_{source}(\mathbf{h}(\mathbf{s}))$. This means that the small area (ideally a point) resulting from the shrinking of the hole is sent by \mathbf{h} on the whole hole, corresponding indeed to a dilatation.

Now let us consider the reverse case of registering Fig. 3(b) on Fig. 3(a). Using an asymmetric SSD cost function will lead to the estimation of an identity transformation Fig. 3(d) as there is no way to decrease the value of the objec-

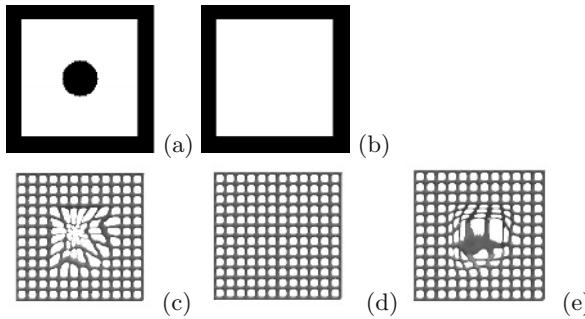


Fig. 3. 2-D slices of a 3-D toy example illustrating the contribution of using a symmetric similarity criterion: (a) plain cube with a spherical hole inside; (b) plain cube; (c) resulting transformation for a registration of (a) on (b) applied to a synthetic grid (the same result is observed when using either symmetric or asymmetric similarity criteria); (d) resulting transformation for a registration of (b) on (a) using an asymmetric similarity criterion; (e) resulting transformation for a registration of (b) on (a) using a symmetric similarity criterion.

tive function. This is not satisfactory as it means that the topology preserving registration method would be able to cope with disappearance of structures but not with appearance of new ones. Moreover this means that registering an image A on an image B and then registering B on A will lead to two transformations which are not the inverse of each other. This problem does not appear when using a symmetric cost function as E_{sym} (5). When registering Fig. 3(b) on Fig. 3(a) using E_{sym} , thanks to the term $(1 + |J|)$, the resulting deformation field tends to send as few points as possible towards the counterpart black hole, in order to assign less weight to this area in the similarity criterion . The resulting transformation (Fig. 3(e)) corresponds indeed to the inverse of the transformation represented on Fig.3(c). Thus, although there is no difference in the warped image, the use of a symmetric cost function allows to deal with the appearance of new structures, with significant consequences on the deformation field. In particular the Jacobian becomes very small at the location corresponding to the new appearing structure.

3.3 Application to Tumor Detection

The benefit of symmetric and topology preserving warping is finally illustrated on a simple simulation of tumor (or lesion) detection. To this end, we introduce a dark inclusion representing a tumor in a real 3-D MRI (Fig. 4(b)) and we register it with a healthy brain image from another patient (Fig. 4(a)). When registering Fig. 4(b) on Fig. 4(a), the topology preserving transformation matches the corresponding brain structures by shrinking the tumor to a small area (ideally a point) (Fig. 4(c)), yielding high values of the Jacobian of \mathbf{h} . By this way, detecting a tumor amounts to finding particularly high values of the Jacobian.

Let us now consider the reverse case of registering Fig. 4(a) on Fig. 4(b). Without the positivity constraint on the Jacobian and using the SSD criterion,

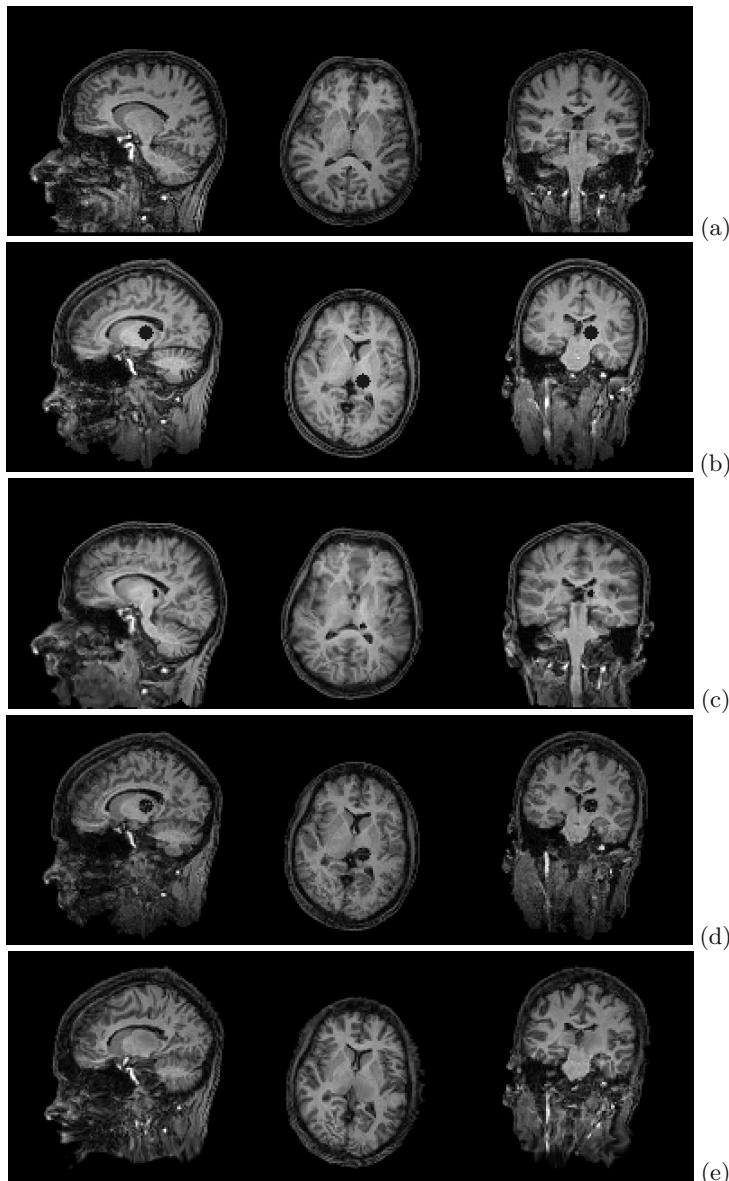


Fig. 4. Registration of two topologically different brain images: (a) normal - patient 1 - brain; (b) brain of patient 2 with a simulated tumor; (c) result of matching (b) on (a) with the symmetric similarity criterion and under the constraint of positivity of the Jacobian; (d) result of matching (a) on (b) with the symmetric similarity criterion and without the positivity constraint on the Jacobian; (e) result of matching (a) on (b) under the constraint of positivity of the Jacobian (same result with both symmetric and asymmetric similarity criteria).

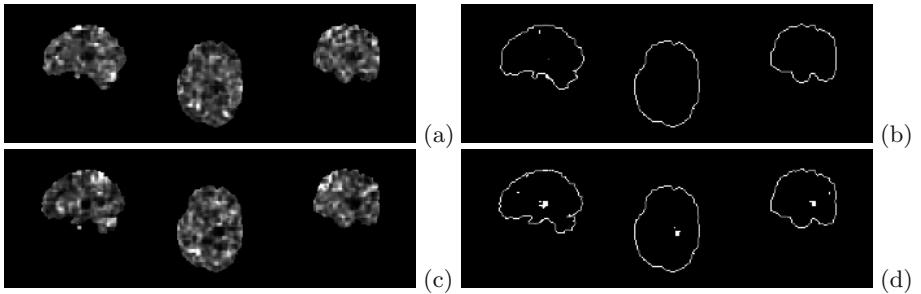


Fig. 5. Jacobian maps computed from the mapping of topologically different brain images: (a) Jacobian map (source: Fig. 4(a), target: Fig. 4(b), positive Jacobian constraint, asymmetric similarity criterion); (b) thresholded Jacobian map (no tumor is detected); (c) Jacobian map (source: Fig. 4(a), target: Fig. 4(b), positive Jacobian constraint, symmetric similarity criterion); (d) thresholded Jacobian map (tumor is detected).

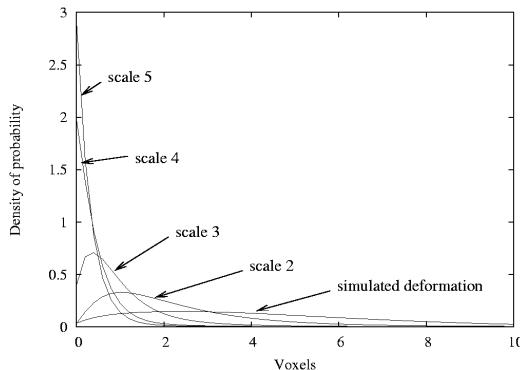


Fig. 6. Histograms of residual vector field errors computed for nine 128^3 estimated deformation fields after deformable matching at scales $L = 2, 3, 4$ and 5.

a new structure is created by the warping to match the tumor. This new structure stems from a tearing of the ventricle (Fig. 4(d)). If the positivity constraint on the Jacobian is enforced, almost nothing can be seen visually on the warped image (Fig. 4(e)). With an asymmetric similarity criterion, the Jacobian of the estimated transformation has no abnormal small values, and consequently nothing can be detected when thresholding the Jacobian map (Fig. 5(b)). On the contrary, when registering Fig. 4(a) on Fig. 4(b) with the symmetric criterion E_{sym} , very small values of the Jacobian are obtained in the critical area. In this case, the tumor can be detected by simply thresholding the Jacobian map (Fig. 5(d)). Detection of tumor may thus be achieved by registering both images (whatever the choice of the reference image) with a topology preserving mapping using a symmetric similarity criterion, and then by finding abnormal (high or low) values of the Jacobian of the estimated transformation.

3.4 Validation of 3-D Deformable Matching with Simulated Fields

Validation of non-rigid image registration is an intricate problem due to the lack of ground truth. We propose to assess the relevance of the registration algorithm by generating a ground truth using simulated transformations applied to real MRIs. To this end, we generate a transformation $\mathbf{h}_{\text{simulated}}$ with large nonlinear deformations while preserving topology, as described in the appendix. For a given real MR Image I , we register $I(\mathbf{h}_{\text{simulated}}(\mathbf{s}))$ on $I(\mathbf{s})$ with the Levenberg-Marquardt algorithm using the symmetric SSD criterion and under the positivity constraint of the Jacobian. The resulting transformation $\mathbf{h}_{\text{estimated}}$ should be compared with $\mathbf{h}_{\text{simulated}}^{-1}$. The errors between the “true” transformation and the estimated one have been evaluated using the standard \mathcal{L}_2 norm: $\mathcal{L}_2(\mathbf{s}) = \|\mathbf{h}_{\text{estimated}}(\mathbf{s}) - \mathbf{h}_{\text{simulated}}^{-1}(\mathbf{s})\|$. Fig. 6 shows the mean histograms of the error vector field computed for three random deformation fields applied on three different MR images. The histograms are computed at different scales ($L = 2, 3, 4, 5$). As can be seen, at scale 5, more than 96% of the voxels are affected by a quadratic error lower than one voxel.

4 Conclusion

In this paper, we have described a novel parametric modeling approach, enabling topology preservation in the registration of 3-D brain images. Topology preservation is enforced on the continuous image domain by the way of a positivity constraint on the Jacobian of the transformation. The registrations of 3-D 128^3 MR brain images are performed² in less than 30 minutes of CPU time on a 2.4 GHz PC workstation. We have also shown that the introduction of a symmetric form of the similarity criterion leads to a more consistent registration and paves the way to tumor detection.

References

1. Musse, O., Heitz, F., Armspach, J.P.: Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Transactions on Image Processing* **10** (2001) 1081–1093
2. Toga, A., Thompson, P.: Image registration and the construction of multidimensional brain atlases. In Bankman, I., ed.: *Handbook of medical imaging. Processing and analysis*. Academic Press (2000) 635–653
3. Cachier, P., Rey, D.: Symmetrization of the non-rigid registration problem using inversion-invariant energies: application to multiple sclerosis. In: Proc. of MICCAI'00 - LNCS 1935, Pittsburgh, USA (2000) 472–481
4. Ashburner, J., Andersson, J., Friston, K.: Image registration using a symmetric prior - in three dimensions. *Human Brain Mapping* **9** (2000) 212–225
5. Thirion, J.P.: Image matching as a diffusion process: an analogy with Maxwell’s demons. *Medical Image Analysis* **2** (1998) 243–260

² Registration is achieved up to scale $L = 5$, corresponding to about 90.000 parameters.

6. Christensen, G., Johnson, H.: Consistent image registration. *IEEE Transactions on Medical Imaging* **20** (2001) 568–582
7. Mallat, S.: A wavelet tour of signal processing. Academic Press (1998)
8. Musse, O., Heitz, F., Armspach, J.P.: Fast deformable matching of 3D images over multiscale nested subspaces. Application to atlas-based MRI segmentation. *Pattern Recognition* **36** (2003) 1881–1899
9. Heinrich, C., Noblet, V., Heitz, F., Armspach, J.P.: 3-D deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization. Technical report, LSIIT, UMR CNRS-ULP 7005 (2003)
10. Rao, S.: Engineering optimization: theory and practice. third edn. Wiley-Interscience (1996)
11. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied interval analysis. Springer (2001)

Appendix: Generation of a Topology Preserving Transformation

First we randomly generate a set of subboxes Ω_i , $i \in [1..N]$, defining a partition of the image Ω . To generate the partition, we randomly draw a point in the image thus defining 8 subboxes. Then for each subbox, we reiterate the operation until the size of each subbox is lower than a user-defined value. On each subbox $\Omega_i = [x_m^i, x_M^i] \times [y_m^i, y_M^i] \times [z_m^i, z_M^i]$, we define a transformation \mathbf{h}_i as follows:

$$\mathbf{h}_i(x, y, z) = \begin{bmatrix} x + a_x^i \sin \pi \left(\frac{x - x_m^i}{x_M^i - x_m^i} \right) \sin \pi \left(\frac{y - y_m^i}{y_M^i - y_m^i} \right) \sin \pi \left(\frac{z - z_m^i}{z_M^i - z_m^i} \right) \\ y + a_y^i \sin \pi \left(\frac{x - x_m^i}{x_M^i - x_m^i} \right) \sin \pi \left(\frac{y - y_m^i}{y_M^i - y_m^i} \right) \sin \pi \left(\frac{z - z_m^i}{z_M^i - z_m^i} \right) \\ z + a_z^i \sin \pi \left(\frac{x - x_m^i}{x_M^i - x_m^i} \right) \sin \pi \left(\frac{y - y_m^i}{y_M^i - y_m^i} \right) \sin \pi \left(\frac{z - z_m^i}{z_M^i - z_m^i} \right) \end{bmatrix}.$$

The resulting transformation leaves the borders of Ω_i invariant, so that preserving topology on Ω is equivalent to ensuring bijectivity of every \mathbf{h}_i on each Ω_i . The set of parameters $[a_x^i; a_y^i; a_z^i]$ are randomly generated while observing the following condition:

$$(\forall \mathbf{s} \in \Omega_i \quad J_{\mathbf{h}_i}(\mathbf{s}) > 0) \Leftrightarrow \left(\frac{|a_x^i|}{x_M^i - x_m^i} + \frac{|a_y^i|}{y_M^i - y_m^i} + \frac{|a_z^i|}{z_M^i - z_m^i} < \frac{1}{\pi} \right).$$

Hence \mathbf{h} is defined as: $\forall \mathbf{s} \in \Omega_i \quad \mathbf{h}(\mathbf{s}) = \mathbf{h}_i(\mathbf{s})$. In this scheme, many points in the image are left invariant by \mathbf{h} since the borders of each Ω_i are invariant. Following the same approach, we thus generate a second transformation \mathbf{h}' defined on another random partition of the image, and then we compose \mathbf{h} and \mathbf{h}' to obtain the final transformation $\mathbf{h}_{simulated}$. Finally, we compute numerically the inverse transformation $\mathbf{h}_{simulated}^{-1}$ of $\mathbf{h}_{simulated}$ while we register the warped image on the original one. The inversion is conducted by estimating for each point an interval containing its antecedent and by iteratively reducing the size of this interval until reaching the desired accuracy.

A Correlation-Based Approach to Robust Point Set Registration

Yanghai Tsin¹ and Takeo Kanade²

¹ Siemens Corporate Research,
755 College Road East, Princeton, NJ 08540, USA

² Carnegie Mellon University,
5000 Forbes Ave., Pittsburgh, PA 15213, USA

Abstract. Correlation is a very effective way to align intensity images. We extend the correlation technique to point set registration using a method we call kernel correlation. Kernel correlation is an affinity measure, and it is also a function of the point set entropy. We define the point set registration problem as finding the maximum kernel correlation configuration of the two point sets to be registered. The new registration method has intuitive interpretations, simple to implement algorithm and easy to prove convergence property. Our method shows favorable performance when compared with the iterative closest point (ICP) and EM-ICP methods.

1 Introduction

Point set registration is among the most fundamental problems in vision research. It is widely used in areas such as range data fusion, medical image alignment, object localization, tracking, object recognition, just to name a few.

One of the most effective methods in registration is correlation. In vision problems, correlation between two image patches has long been used for measuring the similarities between them. When studying discrete point sets, such as those returned by range sensors or feature detectors, however, we are given just the coordinates of a set of points. The definition of correlation is no longer directly applicable since we are given a set of geometric entities without any appearance information to compare.

Nevertheless, the presence or absence of feature points themselves tell a lot more than the coordinates of individual points. They also present the structure implied by the point sets. The simplest way of capturing such structure is to treat the feature points as binary intensity images which have only values 0 (absence) and 1 (presence). However, when noise presents, or when we have different sampling strategies in obtaining the two point sets, the binary images usually do not match.

In the following we present a technique we call kernel correlation that extends the concept of correlation to point sets. We begin by introducing kernel correlation.

2 Kernel Correlation

2.1 Definitions

Kernel correlation (KC) is defined on three levels. First, it is defined on two points. Given two points x_i and x_j , their kernel correlation (KC) is defined as

$$KC(x_i, x_j) = \int K(x, x_i) \cdot K(x, x_j) dx. \quad (1)$$

Here $K(x, x_i)$ is a kernel function centered at the data point x_i . We limit ourselves to the symmetric, non-negative kernel functions that are usually used in the Parzen window density estimation [11], instead of the broader “kernel” definition in the machine learning community. Such kernels include the Gaussian kernel, Epanechnikov kernel, tri-cubic kernel, *et. al.* [7]. In the following we use the Gaussian kernel as an example for its simplicity. The Gaussian kernel has the form:

$$K_G(x, x_i) = (\pi\sigma^2)^{-D/2} \exp(-\|x - x_i\|^2/\sigma^2). \quad (2)$$

Here $\|x - y\|$ is the Euclidean distance between two vectors x and y , and D is the dimension of the vectors.

Because the kernel functions we adopt are symmetric, it’s not surprising to see that the KC defined in (1) is a function of distance between the two points. For example, the KC corresponding to the Gaussian kernel is,

$$KC_G(x_i, x_j) = (2\pi\sigma^2)^{-D/2} \exp\left\{-\|x_i - x_j\|^2/2\sigma^2\right\}. \quad (3)$$

KC’s for other kernels can be shown to be functions of distance $\|x_i - x_j\|$ as well. For clarity of the presentation we will not list them here. But we will discuss their shared properties with the Gaussian kernel whenever necessary. Right side of (3) is known in the vision community as “affinity” or “proximity”[17]: a closeness measure. In this paper we show its utility in registration problems.

Next we define the KC between a point and the whole set of points \mathcal{X} , the *Leave-one-out Kernel Correlation* (LOO-KC),

$$KC(x_i, \mathcal{X}) = \sum_{x_j \neq x_i} KC(x_i, x_j). \quad (4)$$

According to (3) and (4), for the Gaussian case we have

$$KC_G(x_i, \mathcal{X}) = (2\pi)^{-D/2} \sum_{x_j \neq x_i} \exp\left\{-\|x_j - x_i\|^2/2\sigma^2\right\} \quad (5)$$

Leave-one-out KC defines the total affinity from a point to a point set.

Finally, we extend the KC definition to a point set: the total sum of the LOO-KC of all the points x_k in the set,

$$KC(\mathcal{X}) = \sum_i KC(x_i, \mathcal{X}) = 2 \sum_{i \neq j} KC(x_i, x_j). \quad (6)$$

If the points in the set are close to each other, the KC value is large. In this sense KC of a point set is a compactness measure of the point set.

2.2 Entropy Equivalence

If we define the density of the point set \mathcal{X} as the kernel density estimate:

$$P(x) = \sum_{i=1}^N K(x, x_i)/N, \quad (7)$$

and adopt the Renyi's Quadratic Entropy (RQE) [15] as,

$$H_{rqe} = -\log \int_x P(x)^2 dx, \quad (8)$$

KC of the point set has a simple relationship with the entropy measure,

$$KC(\mathcal{X}) \propto C + \exp\{-H_{rqe}\}. \quad (9)$$

The above observation follows directly by expanding the $\int_x P(x)^2 dx$ term in the entropy definition. In fact,

$$\begin{aligned} N^2 \cdot \int P(x)^2 dx &= \left(\sum_i \int_x K(x, x_i)^2 dx + 2 \sum_{i \neq j} \int_x K(x, x_i) K(x, x_j) dx \right) \\ &= C' + KC(\mathcal{X}). \end{aligned}$$

Here we use the fact that $\int_x K(x, x_i)^2 dx$ is a constant and the definition of $KC(\mathcal{X})$ (6). Note that the relationship does not assume any specific form of kernel function, as long as the integrals are defined.

Thus the compactness measure of KC is linked to the compactness measure of entropy. A minimum entropy system is the one with the maximum affinity (minimum distance) between all pairs of points. The information theoretic compactness measure indeed has a geometric interpretation.

We were brought to the attention of the independent work by Principe and Xu [13]. They expanded the RQE definition in the Gaussian case and defined the integral of the cross product terms as “information potential”. Their purpose for such decomposition is efficient evaluation of entropy and entropy gradients in the context of information theoretic learning. In contrast, our goal is instead to configure a dynamic point set.

2.3 KC as an M-Estimator

If there are just two points involved, maximum KC corresponds to minimum distance between them. However, when we are dealing with multiple points, it's not immediately obvious what is being optimized. For instance, in the Gaussian case we have (5). What does it mean to maximize KC? It turns out that in this case we are still minimizing the distance, but in the sense of M-estimators.

In an M-estimator, instead of minimizing the usual sum of quadratic distances, $E_q = \sum_j (x_i - x_j)^2$, we are minimizing a robust version of the distance

function $E_r = \sum_j g((x_i - x_j)^2)$, where g is a robust function [8]. The advantage of changing from the quadratic distance function to the robust function is that local configuration of x_i is insensitive to remote points. To see this we compare the gradients of the above two functions.

$$\partial E_q / \partial x_i \propto \sum_j (x_i - x_j) \quad (10)$$

$$\partial KC_G(x_i, \mathcal{X}) / \partial x_i \propto \sum_j \exp(-\|x_i - x_j\|^2 / 2\sigma^2)(x_j - x_i). \quad (11)$$

The gradient term (10) is very sensitive to outliers in that any outlier point x_j can have arbitrarily large contribution to the gradient. Remember that the gradient is the direction (and magnitude in the quadratic function case) to update x_i . To minimize E_q , estimation of x_i will be severely biased toward the outlier points. In the KC_G case, however, there is a second term $\exp(-\|x_i - x_j\|^2 / 2\sigma^2)$ that decays exponentially as a function of distance. Consequently, remote outliers will have no influence to local E_r minimization.

When we use kernels other than the Gaussian kernel, we can still have the M-estimator equivalence when maximizing KC. For example, by using the Epanechnikov kernel, we implicitly embedded a line process [5] into the correlation process: Points beyond a certain distance don't contribute.

Chen and Meer [3] also observed the equivalence of mode finding in a kernel density estimate and the M-estimators. The difference is that they are fitting parametric models to a set of static data (the projection pursuit example), or clustering the static point set. The introduction of KC is to robustly configure dynamic point sets.

3 Kernel Correlation for Registration

Given two finite size point sets, the *model* set \mathcal{M} and the *scene* set \mathcal{S} , our registration method is defined as finding the parameter θ of a transformation T to minimize the following cost function,

$$\text{COST}(\mathcal{S}, \mathcal{M}, \theta) = - \sum_{s \in \mathcal{S}} \sum_{m \in \mathcal{M}} KC(s, T(m, \theta)). \quad (12)$$

Notice that in the above equation each transformed model point m is interacting with all the scene points. We call (12) a *multiply-linked* registration cost function. This is in contrast to the ICP algorithm, where each model point is connected to its nearest scene point only. It can be shown that,

$$KC(\mathcal{S} \cup T(\mathcal{M}, \theta)) = KC(\mathcal{S}) + KC(T(\mathcal{M}, \theta)) - 2\text{COST}(\mathcal{S}, \mathcal{M}, \theta). \quad (13)$$

$KC(\mathcal{S})$ is independent of θ . Under rigid transformation, $KC(T(\mathcal{M}, \theta))$ is also constant. This is the case because KC is a function of Euclidean distances between pairs of points (e.g. (3)). Rigid transformation reconfigures the point set as a whole and preserves the Euclidean distances between all pairs of points. Thus $KC(T(\mathcal{M}, \theta))$ is invariant. As a result $KC(\mathcal{S} \cup T(\mathcal{M}, \theta)) = C - 2\text{COST}(\mathcal{S}, \mathcal{M}, \theta)$.

Due to the equivalence of KC and entropy (Section 2.2), our registration method implies finding the minimum entropy configuration of the joint point set $\mathcal{S} \cup T(\mathcal{M}, \theta)$ in the RQE sense.

By denoting the kernel density estimates (KDE) as

$$P_{\mathcal{M}}(x, \theta) = \sum_{m \in \mathcal{M}} K(x, T(m, \theta))/N, \quad P_{\mathcal{S}}(x) = \sum_{s \in \mathcal{S}} K(x, s)/N,$$

we can show that the cost function is also proportional to the correlation of the two KDE's,

$$\text{COST}(\mathcal{S}, \mathcal{M}, \theta) = -N^2 \int_x P_{\mathcal{M}} \cdot P_{\mathcal{S}} dx. \quad (14)$$

3.1 Convergence of a KC Registration Algorithm

It's easy to show that the cost function (12) is bounded from below. If we use gradient descent based method to minimize the cost function such that the cost function is decreasing at each step, the convergence of the cost function to a fixed point is guaranteed. Convergence properties for other registration methods, such as ICP or EM-ICP are usually difficult to study because their cost functions, defined on nearest neighbors, change from iteration to iteration as the point configuration evolves. In contrast, the KC registration function is defined globally and each step of minimization decreases the *same* cost function.

3.2 Accuracy of KC Registration

We will empirically study the accuracy of our registration algorithm in Section 5. Here we will discuss one of the simplest cases to theoretically characterize the KC registration algorithm.

Given a point set \mathcal{M} and it's transformed version $\mathcal{S} = T(\mathcal{M}, \theta^*)$, a registration method should satisfy what we call the *minimum requirement for a registration algorithm*. That is, θ^* should correspond to one of the global minima of the cost function. Although this requirement seems to be trivial, we will show in our experiments that it is not met by other multiply-linked registration algorithms. Here we first give a proof that our registration algorithm meets the minimum requirement under rigid transformation. The extension to no-rigid motion is followed. We observe that

$$N^2 \int_x (P_{\mathcal{M}} - P_{\mathcal{S}})^2 dx = N^2 \left(\int_x P_{\mathcal{M}}^2 dx + \int_x P_{\mathcal{S}}^2 dx - 2 \int_x P_{\mathcal{M}} \cdot P_{\mathcal{S}} dx \right) \quad (15)$$

$$= C + KC(T(\mathcal{M}, \theta)) + KC(\mathcal{S}) + 2 \cdot \text{COST}(\mathcal{S}, \mathcal{M}, \theta). \quad (16)$$

Here C is a constant due to KC values of a point with itself. $KC(\mathcal{S})$ is independent of θ . As we discussed in the beginning of this section, $KC(T(\mathcal{M}, \theta))$ is also a constant under rigid transformation. Thus minimizing the left side of (15) is equivalent to minimizing our registration cost function. When $\theta = \theta^*$, $P_{\mathcal{M}}$ and

P_S are exactly the same and the left side of (15) is zero, the global minimum. That is, θ^* corresponds to one of the global minima of the KC registration cost function. Note that this statement is independent of the kernel functions being chosen and the kernel scale, as long as the integrals in the proof are all defined.

The KC registration framework can be extended to non-rigid transformations if we minimize a normalized KC cost function. By denoting the normalization term as $I_{\mathcal{M}} = (\int_x P_{\mathcal{M}}^2 dx)^{1/2}$ the normalized cost function is

$$\text{COST}_n = - \sum_{s \in \mathcal{S}, m \in \mathcal{M}} KC(s, T(m, \theta)) / I_{\mathcal{M}}. \quad (17)$$

Similar to (16), we can show that

$$N^2 \int_x (P_{\mathcal{M}} / I_{\mathcal{M}} - P_S / I_S)^2 dx = 2N^2 + 2 \cdot \text{COST}_n / I_S, \quad (18)$$

where $I_S = (\int_x P_S^2 dx)^{1/2}$ is independent of θ . Given that $\mathcal{S} = T(\mathcal{M}, \theta^*)$, θ^* will again be one of the global minima of the registration cost function (17), even under non-rigid transformations.

3.3 Discrete Approximation of the Registration Cost Function

In practice we don't need to enumerate each pair of model and scene points in order to evaluate the cost function (12) or (17). We can use the discrete version of (14) to approximate the registration cost. That is, we compute two discrete KDE's, $P_{\mathcal{M}}(x, \theta)$ and $P_S(x)$ at grid points x , and use $-\sum_x P_{\mathcal{M}}(x, \theta) \cdot P_S(x)$ to approximate the scaled cost function. Compared to the ICP or EM-ICP methods, there is no nearest neighbor finding step involved in the KC registration, which can result in significant simplification in algorithm implementation. $P_S(x)$ plays the role of an affinity map in our algorithm. The affinity of a model point m to the scene points can be computed by correlating $K(x, m)$ with $P_S(x)$.

4 Related Work

We store the affinity information in a density estimate. This technique bears much resemblance to the registration methods based on distance transform (DT) [2]. However, there are some important differences. First, DT is known to be extremely sensitive to outliers and noise because a single point can have influence to a large area. The influence of each point in the KC case is local. Thus KC based registration can be robust to outliers. Second, our affinity map is usually sparse for usual point sets such as an edge map or a laser range scan, with most grid points having zero values. The affinity map can be efficiently stored in data structures such as an octree. In contrast, high resolution DT in 3D is very costly. This prompted Lavallée and Szeliski to approximate 3D DT using octree spline [10].

One elegant registration method based on DT is the partial Hausdorff distance registration [9]. By minimizing partial Hausdorff distance, a registration algorithm can have up to 50 percent breakdown point. The underlying robustness mechanism is the same as the *least median of squares* (LMedS) algorithm [16] in robust regression. However, the registration depends on a single critical point in the data set and most information provided by other points are ignored. Compared to other registration methods such as ICP and our proposed method, it is very sensitive to noise.

Scott and Longuet-Higgins [17] explored the possibility of finding correspondence by singular value decomposition (SVD) analysis of an affinity matrix, whose elements are proportional to the Gaussian KC values. Their algorithm is known to be vulnerable to perturbations such as large rotations, outliers and noise. In addition, forming a large affinity matrix for a large point set is costly.

One of the most successful point set registration algorithms is the iterative closest point (ICP) algorithm [1,19]. A naive implementation of ICP is not robust because the cost function is a quadratic function of distance. To be robust, line-process like outlier detection or M-estimator like cost functions have been suggested [19,4]. KC registration can be considered as multiply-linked and robust ICP. The benefits of establishing multiple-links will become clear when we compare the two algorithms in Section 5.

KC registration is mathematically most related to the EM-ICP algorithm [6] and the SoftAssignment algorithm [14], which are also multiply-linked ICP. For example, at each step, EM-ICP minimizes the following function:

$$\sum_{m \in \mathcal{M}} \sum_{s \in \mathcal{S}} \exp(-\|T(m, \theta) - s\|^2 / \sigma^2) \|T(m, \theta) - s\|^2 / N(m, \theta). \quad (19)$$

where $N(m, \theta) = \sum_s \exp(-\|T(m, \theta) - s\|^2 / \sigma^2)$ is a normalization term. In fact, the KC_G cost function has the same gradient as EM-ICP, except the normalization term. Due to these mathematical similarity, KC registration and EM-ICP performs very similarly, except that EM-ICP does not meet the minimum requirement of a registration algorithm: The exactly aligned point sets does not correspond to the global minimum of the EM-ICP cost function. Depending on the point sets being registered and the kernel scale, the EM-ICP (as well as SoftAssignment) algorithms can give biased registration even for clean data. This point will be demonstrated in our experiments. For in-depth discussion on this topic, the reader is referred to our technical report [18] (pp. 56-59). In addition, the KC provides a framework for using different kernel functions for registration and its convergence proof does not rely on statistical methods such as EM.

5 Performance Evaluation

We compare the KC registration algorithm with the ICP and EM-ICP algorithm in this section. We implemented two versions of the KC algorithm. The first one is a simple 2D Matlab version that uses the Matlab “fminsearch” function (Nelder-Mead simplex method) for optimization. In this case the gradients are not explicitly computed. The second implementation is a C++ 3D version that computes

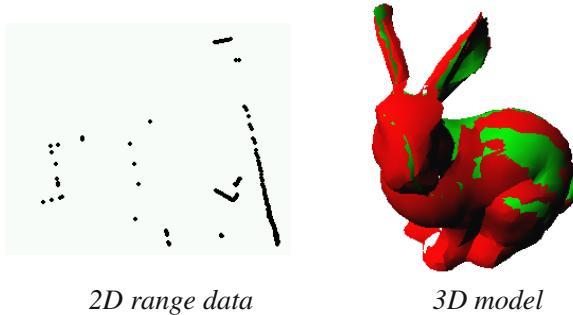


Fig. 1. The data set for performance evaluation.

gradients and uses the variable metric method [12] for optimization. The 2D Matlab code is available at webpage: <http://www.cs.cmu.edu/~ytsin/KCReg/>. The 3D ICP registration algorithm used for comparison is developed independently by the CMU 3D vision group (<http://www.cs.cmu.edu/~3dvision/>). Both the 3D ICP and our 2D ICP code in Matlab implemented the outlier thresholding mechanism [19].

For evaluation purpose, we use a 2D range data (the *road-data*, 277 points) acquired by a SICK LMS 221 laser scanner and a 3D bunny model (the "bunny1"-data, 699 points) acquired by a Minolta Vivid 700 scanner. The models are shown in Figure 1. Extensive registration experiments on thousands of 2D scans and some other 3D models can be found at our website.

5.1 Convergence Region

We first test the convergence properties of ICP, EM-ICP and KC in 2D. Two copies of the *road data* are generated by adding different random noise. One of the copies is then rotated on its center of mass for a certain angle. We study the convergence performance of the three registration methods by registering rotated point sets at different angles. The results are shown in Figure 2. The leftmost plot shows the registration costs as functions of the rotation angle. Note that we allow full 2D Euclidean motions and the cost is a 3D function. We plot a 1D slice of the cost function for clarity. With a kernel scale of $\sigma = 15$, both EM-ICP and KC have very smooth cost functions in the whole test range. In the ICP cost function we see a lot of local minima, which correspond to the much smaller convergence region in the center plot of Figure 2. The plot shows average registration error between corresponding points after registration. The EM-ICP has a little wider convergence region than the KC registration in this data set. However, we observed constantly larger registration error in the EM-ICP case. Here we experimentally demonstrate that EM-ICP does not meet the minimum requirement for registration. The right plot shows the average registration error as a function of the kernel scale in the noiseless case. KC registration has zero error regardless of the kernel scale.

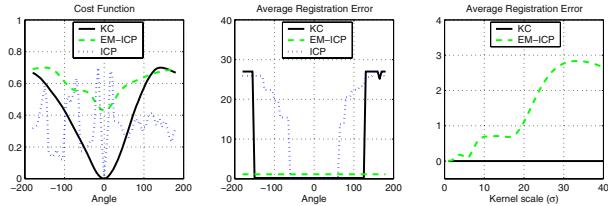


Fig. 2. 2D convergence study. The widest ICP convergence region (with varying outlier detection threshold) is shown here.

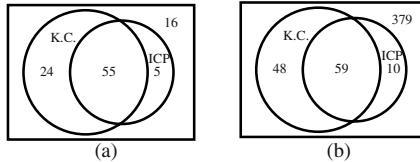


Fig. 3. Venn diagram of the sets of successfully registered model-pairs. Numbers are sizes of the regions. (a) Random transformation. (b) Pairwise.

We conduct two different convergence test on the 3D data. We draw 100 random θ (6D parameter space) samples from a uniform distribution. We transform the *bunny1* data set using the random parameters and form 100 pairs for registration. The Venn diagram of the successful registrations is shown in Figure 3(a). The KC method has larger convergence region(79 versus 60, or 24 versus 5 when excluding the “easy” cases for both).

Next, we study pairwise registration of 32 scans of the bunny model acquired from different views by the laser scanner. There are in total 496 pairs of point sets to be registered. We visually examine each of the registration results. The Venn diagram for this experiment is shown in Figure 3(b). Again, the KC method has larger success rate than ICP (107 versus 69, or 48 versus 10 excluding the “easy” cases).

Our experiments show that the KC registration method has larger convergence region. This is due to a smoothed cost function which enables an optimization algorithm to find a good registration more easily. The smoothness is provided by weighting the contributions of the multiple-links between a point and its neighbors.

5.2 Sensitivity to Noise

For both the 2D and and 3D data, we use the same method to test how sensitive the registration methods are in the presence of noise perturbation. We generate slightly rotated versions of the same point set, and add zero mean random noise

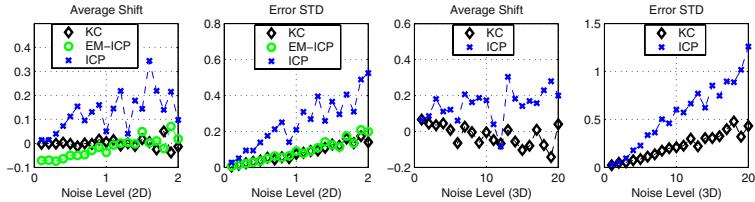


Fig. 4. Sensitivity to noise tests.

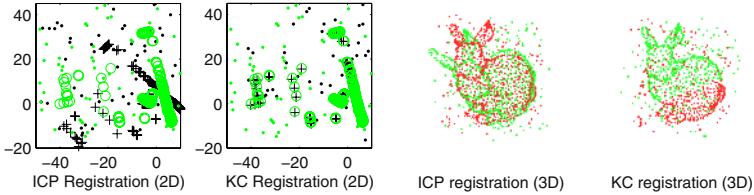


Fig. 5. Robustness tests. Note the added outlier points in all cases.

to both the reference model and the rotated model. At every noise level, we register 30 pairs of noise corrupted point sets.

After registration we compute the average shift between corresponding points in the two registered point sets. If the point sets are registered well, the average shift should be close to zero because the added noise has zero mean. We can thus use the standard deviation of the average shifts over the 30 pairs as a measure of sensitivity to noise. We plot the standard deviation and average shift as a function of the noise level in Figure 4. The kernel scale is 5 for the 2D tests and 20 for the 3D tests. In both (2D and 3D) cases we observe that KC registration has smaller variance than ICP. At the same time, the registration error is small.

The superior capability of the KC technique can be explained by its extended interaction with neighbors. KC considered weighted effects of points in a large neighborhood, instead of just its immediate nearest neighbor, thus achieving better ability to resist noise.

5.3 Robustness

To test robustness, we register outlier corrupted point sets. We generate the reference model and the rotated model the same way as the previous section. Instead of corrupting the models with noise, we add 20% of outliers. The outlier points are randomly drawn from a uniform distribution. The corrupted point sets is illustrated in Figure 5. In both the 2D and 3D cases, we use the ICP and KC methods to register 100 pairs of outlier corrupted data.

Examples of 2D registration final results are presented in the left two plots of Figure 5. For ICP we tried three outlier-detection thresholds, 20, 5 and a concatenation of 20 and 5. The best of the three, by concatenating two ICP

registrations with thresholds 20 and 5, correctly registered 43 out of 100 pairs. In contrast, KC registration robustly registered all 100 pairs.

Exemplar 3D registration final results are demonstrated in the right two plots of Figure 5. The performance of the ICP algorithm is beyond our expectation. It failed only in 8 pairs of the outlier corrupted data sets. Still, KC registration can achieve better robustness. Again, KC registered all 100 pairs without mistake, by using a large range of different kernel scales.

In our experiments with KC registration in 3D, we do observe failed cases when the scale is either too small (easily fell victim to outlier distractions) or too large (containing too many outliers in the neighborhood). Thus in the presence of outliers, proper scale selection is an important and open issue in our technique.

In our experiments there are two kinds of outliers. First, the points that have a large distance to all the model points. These points are taken care of by both the M-estimator mechanism of KC, and the distance thresholding of ICP. Second, the points that fall in the neighborhood of a model point. These points can be very distracting to singly-linked methods such as ICP. For KC, each point is connected to multiple points. As long as the percentage of outliers in the local neighborhood is small, their influence can be averaged out by contributions from other inlier points. Consequently, KC is capable of registering despite of these local distractions.

6 Conclusions

In this paper we introduced a registration method by dynamically configuring point sets, whose fitness is measured by KC. KC is shown to be an M-estimator. KC is also equivalent to an entropy measure.

KC based registration can be considered as a robust, multiply-linked ICP. It has a built-in smoothing mechanism that makes it very important in dealing with noise and outlier corrupted data sets. We experimentally demonstrated that it outperforms ICP in terms of convergence region, robustness and resistance to noise, and it outperforms EM-ICP in terms of registration accuracy.

Kernel function selection is an interesting direction. The choice of kernels determines the underlying robust function to be used. We leave it to our future research.

Acknowledgment. We thank Daniel Huber and Bob Chieh-Chih Wang from Carnegie Mellon for providing the 3D and 2D test data used in this work.

References

1. P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE TPAMI*, 14(2):239–256, February 1992.
2. G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE TPAMI*, 10(6):849–865, November 1988.

3. H. Chen and P. Meer. Robust computer vision through kernel density estimation. In *(ECCV'02)*, pages Part I, 236–250. Springer-Verlag, May 2002.
4. A. Fitzgibbon. Robust registration of 2D and 3D point sets. In *BMVC'01*, 2001.
5. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE TPAMI*, 6:1721–741, 1984.
6. S. Granger and X. Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *(ECCV'02)*, June 2002.
7. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Data Mining, Inference and Prediction*. Springer, 2001.
8. P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.
9. D. P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the hausdorff distance. *IEEE TPAMI*, 15(9):850–863, September 1993.
10. S. Lavallée and R. Szeliski. Recovering the position and orientation of free-form objects from image contours using 3-D distance maps. *IEEE TPAMI*, 17(4):378–390, April 1995.
11. E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
12. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, England, second edition, 1992.
13. J. Principe and D. Xu. Information-theoretic learning using Renyi’s quadratic entropy. In *First International Workshop on Independent Component Analysis (ICA ’99)*, pages 407–412, 1999.
14. A. Rangarajan, H. Chui, and F.L. Bookstein. The softassign procrustes matching algorithm. *Information Processing in Medical Imaging*, pages 29–42, 1997.
15. A. Rényi. On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561. University of California Press, 1961.
16. P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley-Interscience, New York, New York, 1987.
17. G.L. Scott and H.C. Longuet-Higgins. An algorithm for associating the features of two images. *Proceedings: Biological Sciences*, 244(1309):21–26, April 1991.
18. Y. Tsin. Kernel correlation as an affinity measure in point-sampled vision problems. *Techical Report, CMU-RI-03-36*, 2003.
19. Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *IJCV*, 13(2), 1994.

Hierarchical Organization of Shapes for Efficient Retrieval

Shantanu Joshi¹, Anuj Srivastava², Washington Mio³, and Xiuwen Liu⁴

¹ Department of Electrical Engineering, joshi@eng.fsu.edu,

² Department of Statistics, anuj@stat.fsu.edu,

³ Department of Mathematics, mio@math.fsu.edu

⁴ Department of Computer Science, liux@cs.fsu.edu

Florida State University, Tallahassee, FL 32310

Abstract. This paper presents a geometric approach to perform: (i) hierarchical clustering of imaged objects according to the shapes of their boundaries, and (ii) testing of observed shapes for classification. An intrinsic metric on nonlinear, infinite-dimensional shape space, obtained using geodesic lengths, is used for clustering. This analysis is landmark free, does not require embedding shapes in \mathbb{R}^2 , and uses ordinary differential equations for flows (as opposed to partial differential equations). Intrinsic analysis also leads to well defined shape statistics such as means and covariances, and is computationally efficient. Clustering is performed in a hierarchical fashion. At any level of hierarchy clusters are generated using a minimum dispersion criterion and an MCMC-type search algorithm. Cluster means become elements to be clustered at the next level. Gaussian models on tangent spaces are used to pose binary or multiple hypothesis tests for classifying observed shapes. Hierarchical clustering and shape testing combine to form an efficient tool for shape retrieval from a large database of shapes. For databases with n shapes, the searches are performed using $\log(n)$ tests on average. Examples are presented for demonstrating these tools using shapes from Kimia shape database and the Surrey fish database.

1 Introduction

An important goal in image analysis is to classify and recognize objects of interest present in the observed images. Imaged objects can be characterized in many ways: according to their colors, textures, shapes, movements, and locations. The past decade has seen large efforts in modeling and analysis of pixel values or textures in images to attain these goals albeit with limited success. An emerging opinion in the scientific community is that global features such as shapes be taken into account. *Characterization of complex objects using their global shapes is fast becoming a major tool in computer vision and image understanding.* Analysis of shapes, especially those of complex objects, is a challenging task and requires sophisticated mathematical tools. Applications of shape analysis include biomedical image analysis, fisheries, surveillance, biometrics, military target recognition and general computer vision.

Shape is a characteristic that is invariant to rigid motion and uniform scaling of objects, and it becomes natural to analyze shape in a quotient space where these shape preserving transformations have been removed. Shapes have been an important topic of research over the past decade. A majority of this research has been restricted to “landmark-based” analysis where shapes are represented by a coarse, discrete sampling of the object contours [3]. One establishes equivalences of samples (or landmarks) with respect to shape preserving transformations, i.e. rigid rotation and translation, and non-rigid uniform scaling, and then compares shapes in the resulting quotient spaces. This approach is limited in that automatic detection of landmarks is not straightforward and the ensuing shape analysis depends heavily on the choice of landmarks. Another approach is to study the whole object: boundary + interior, in modeling shapes [7]. One limitation of this approach is the need to find computationally expensive diffeomorphisms of \mathbb{R}^n that match shapes. In case the interest lies only in the shapes of boundaries, more efficient techniques can be derived. Another active area of research in image analysis has been the use of level sets and active contours in characterizing object boundaries. However, the focus here is mainly on solving partial differential equations (PDEs) for shape extraction, driven by image features under smoothness constraints, and statistical analysis is more recent [6, 2].

In a recent paper [10], Klassen et al. consider the shapes of continuous, closed curves in \mathbb{R}^2 , without the need for defining landmarks, diffeomorphisms of \mathbb{R}^2 , or nonlinear PDEs. The basic idea is to identify a space of allowable shapes, impose a Riemannian structure on it and utilize its geometry to solve optimization and inference problems. They consider the space of simple closed curves (extended somewhat to simplify analysis), and remove invariances such as rotation, translation, scaling, and re-parametrization to form a shape space. For planar curves in \mathbb{R}^2 , of length 2π and parameterized by the arc length, the coordinate function $\alpha(s)$ relates to the direction function $\theta(s)$ according to $\dot{\alpha}(s) = e^{j\theta(s)}$, $j = \sqrt{-1}$. Direction functions are used to represent curves. Considering closed curves, and making them invariant under rigid motions (rotations, translations), and uniform scaling, one obtains:

$$\mathcal{C} = \left\{ \theta \in \mathbb{L}^2 \mid \frac{1}{2\pi} \int_0^{2\pi} \theta(s) ds = \pi, \int_0^{2\pi} e^{i\theta(s)} ds = 0 \right\}. \quad (1)$$

Here \mathbb{L}^2 is the set of square integrable functions on $[0, 2\pi]$. After removing the re-parametrization group (relating to different placements of $s = 0$ on the same curve), the quotient space $\mathcal{S} \equiv \mathcal{C}/S^1$ is considered as the shape space. Analysis on shapes as elements of \mathcal{S} requires computation of geodesic paths on \mathcal{S} . Let $\Psi(\theta, f, t)$ denote a geodesic path starting from $\theta \in \mathcal{S}$, in the direction $f \in T_\theta(\mathcal{S})$, as a function of time t . In practice, f is represented using an orthogonal expansion according to $f(s) = \sum_{i=1}^{\infty} x_i e_i(s)$, where $\{e_i, i = 1, \dots\}$ form an orthonormal basis of $T_\theta(\mathcal{S})$, and the search for f (to go from one shape to another) can be performed via a search for corresponding $\mathbf{x} = \{x_1, x_2, \dots\}$.

In this paper we advance this idea by studying the following problems.

1. **Problem 1: Hierarchical Clustering:** We will consider the problem of clustering planar objects according to the shapes of their boundaries. To improve efficiency, we will investigate a hierarchy of clusters in which the shapes are recursively clustered in form of a tree. Such an organization can significantly improve database searches and systems with shape-based queries. While retrieving shapes, one can test against a representative shape of each cluster at each level, instead of the whole shape database, and then search for the closest shape in that cluster.
2. **Problem 2: Testing Shape Hypotheses:** Once a shape model is established, it allows for application of decision theory. For example, the question *given an observed shape and two competing shape classes, which class does this shape belong to?* is essentially a binary hypothesis test. Hypothesis tests for landmark-based shape analysis have already been derived [3]. Hypothesis testing, together with hierarchical organization can provide an efficient tool for shape retrieval from a large database.

Beyond these stated goals, these tools can contribute in robust algorithms for computer vision by incorporating shape-based recognition of objects. In this paper we will assume that the shapes have already been extracted from the images and the data is available in form of contours. Of course, in many applications extraction of contours itself is a difficult problem but our focus here is on analyzing shapes once the contours are extracted.

2 **Problem 1: Shape Clustering**

An important need in shape studies is to classify and cluster previously observed shapes. In this section, we develop an algorithm for clustering of objects according to shapes of their boundaries.

Classical clustering algorithms on Euclidean spaces are well researched and generally fall into two main categories: partitional and hierarchical [5]. Assuming that the desired number k of clusters is known, partitional algorithms typically seek to minimize a cost function Q_k associated with a given partition of the data set into k clusters. The total variance of a clustering is a widely used cost function. Hierarchical algorithms, in turn, take a bottom-up approach. If the data set contains n points, the clustering process is initialized with n clusters, each consisting of a single point. Then, clusters are merged successively according to some criterion until the number of clusters is reduced to k . Commonly used metrics include the distance of the means of the clusters, the minimum distance between elements of clusters, and the average distance between elements of the clusters. If the number k of clusters is not known, the problem is harder. Advances in this direction were made in [4].

Clustering algorithms purely based on distances between data points readily generalize to other metric spaces. However, extensions of algorithms that involve finding means of clusters are only meaningful in metric spaces where means can be defined and computed. For the shape space considered here, the notion of

Karcher means for shapes is studied in [10]. However, in some cases the computational cost of computing means may prove to be high. Therefore, it is desirable to replace quantities involving the calculation of means by approximations that can be derived directly from distances between data points as described next.

2.1 Minimum-Variance Clustering

Consider the problem of clustering n shapes (in \mathcal{S}) into k clusters. To motivate our algorithm, we begin with a discussion of a classical clustering procedure for points in Euclidean spaces, which uses the minimization of the total variance of clusters as a clustering criterion. More precisely, consider a data set with n points $\{y_1, y_2, \dots, y_n\}$ with each $y_i \in \mathbb{R}^d$. If a collection $C = \{C_i, 1 \leq i \leq k\}$ of subsets of \mathbb{R}^d partitions the data into k clusters, the total variance of C is defined by $Q(C) = \sum_{i=1}^k \sum_{y \in C_i} \|y - \mu_i\|^2$, where μ_i is the mean of data points in C_i . The term $\sum_{y \in C_i} \|y - \mu_i\|^2$ can be interpreted as the total variance of the cluster C_i . The total variance is used instead of the (average) variance to avoid placing a bias on large clusters, but when the data is fairly uniformly scattered, the difference is not significant and either term can be used. The widely used *k-Means Clustering Algorithm* is based on a similar clustering criterion (see e.g. [5]). The *soft k-Means Algorithm* is a variant that uses ideas of simulated annealing to improve convergence [1,9].

These ideas can be extended to shape clustering using $d(\theta, \mu_i)^2$ instead of $\|y - \mu_i\|^2$, where $d(\cdot, \cdot)$ is the geodesic length and μ_i is the Karcher mean of a cluster C_i on the shape space. However, the computation of Karcher means of large shape clusters is a computationally demanding operation. Thus, we propose a variation that replaces $d(\theta, \mu_i)^2$ with the average distance-square $V_i(\theta)$ from θ to elements of C_i . If n_i is the size of C_i , then $V_i(\theta) = \frac{1}{n_i} \sum_{\theta' \in C_i} d(\theta, \theta')^2$. The cost Q associated with a partition C can be expressed as

$$Q(C) = \sum_{i=1}^k \frac{2}{n_i} \left(\sum_{\theta_a \in C_i} \sum_{b < a, \theta_b \in C_i} d(\theta_a, \theta_b)^2 \right). \quad (2)$$

If the average distance-square within the clusters is used, the scale factor in each term is modified to $\frac{2}{n_i(n_i-1)}$. In either case, we seek configurations that minimize Q , i.e., $C^* = \operatorname{argmin} Q(C)$. In this paper we have used the latter cost function.

2.2 Clustering Algorithm

We will minimize the clustering cost using a Markov chain Monte Carlo (MCMC) search process on the configuration space. The basic idea is to start with a configuration of k clusters and keep on reducing Q by re-arranging shapes amongst the clusters. The re-arrangement is performed in a stochastic fashion using two kinds of moves. These moves are performed with probability proportional to negative exponential of the Q value of the resulting configuration.

1. **Move a shape:** Here we select a shape randomly and re-assign it to another cluster. Let $Q_j^{(i)}$ be the clustering cost when a shape θ_j is re-assigned to the cluster C_i keeping all other clusters fixed. If θ_j is not a singleton, i.e. not the only element in its cluster, then the transfer of θ_j to cluster C_i is performed with the probability:

$$P_M(j, i; T) = \frac{\exp(-Q_j^{(i)}/T)}{\sum_{i=1}^k \exp(-Q_j^{(i)}/T)}, \quad i = 1, 2, \dots, k.$$

Here T plays the role of temperature as in simulated annealing. Note that moving θ_j to any other cluster is disallowed if it is a singleton in order to fix the number of clusters at k .

2. **Swap two shapes:** Here we select two shapes from two different clusters and swap them. Let $Q^{(1)}$ and $Q^{(2)}$ be the Q -values of the original configuration (before swapping) and the new configuration (after swapping), respectively. Then, swapping is performed with the probability:

$$P_S(T) = \frac{\exp(-Q^{(2)}/T)}{\sum_{i=1}^2 \exp(-Q^{(i)}/T)}.$$

Additional types of moves can also be used to improve the search over the configuration space although their computational cost becomes a factor too. In view of the computational simplicity of moving a shape and swapping two shapes, we have restricted the algorithm to these two simple moves.

In order to seek global optimization, we have adopted a simulated annealing approach. That is, we start with a high value of T and reduce it slowly as the algorithm search for configurations with smaller dispersions. Additionally, the moves are performed according to a Metropolis-Hastings algorithm (see [8] for reference), i.e. candidates are proposed randomly and accepted according to certain probabilities (P_M and P_S above). Although simulated annealing and the random nature of the search help in getting out of local minima, the convergence to a global minimum is difficult to establish. As described in [8], the output of this algorithm is a Markov chain but is neither homogeneous nor convergent to a stationary chain. If the temperature T is decreased slowly, then the chain is guaranteed to converge to a global minimum. However, it is difficult to make an explicit choice of the required rate of decrease in T and instead we rely on empirical studies to justify this algorithm. First, we state the algorithm and then describe some experimental results.

Algorithm 1 *For n shapes and k clusters initialize by randomly distributing n shapes among k clusters. Set a high initial temperature T .*

1. Compute pairwise geodesic distances between all n shapes. This requires $n(n - 1)/2$ geodesic computations.
2. With equal probabilities pick one of two moves:

- a) **Move a shape:**
 - i. Pick a shape θ_j randomly. If it is not a singleton in its cluster then compute $Q_j^{(i)}$ for all $i = 1, 2, \dots, k$.
 - ii. Compute the probability $P_M(j, i; T)$ for all $i = 1, \dots, k$ and re-assign θ_j to a cluster chosen according to the probability P_M .
 - b) **Swap two shapes:**
 - i. Select two clusters randomly, and select a shape from each of them.
 - ii. Compute the probability $P_S(T)$ and swap the two shapes according to that probability.
3. Update temperature using $T = T/\beta$ and return to Step 2. We have used $\beta = 1.0001$ in our experiments.

It is important to note that once the pairwise distances are computed, they are not computed again in the iterations. Secondly, unlike k -mean clustering mean shapes are not used here. These factors make Algorithm 1 efficient and effective in clustering diverse shapes.

Now we present some experimental results generated using Algorithm 1. We have applied Algorithm 1 to organize a collection of $n = 300$ shapes (not shown) from the Kimia shape database [12] into 26 clusters. Shown in Figure 1(a) are a few samples from the 26 clusters. In each run of Algorithm 1, we keep the configuration with minimum Q value. Figure 1(b) shows an evolution of the search process where the Q values are plotted against the iteration index. Figure 1(c) shows a histogram of the best Q values obtained in 190 such runs, each starting from a random initial configuration. It must be noted that 90% of these runs result in configurations that are quite close to the optimal. Once pairwise distances are computed, it takes approximately 10 seconds to perform 25,000 steps of Algorithm 1 in the matlab environment. The success of Algorithm 1 in clustering these diverse shapes is visible in these results as similar shapes have been clustered together.

2.3 Hierarchical Organization of Shapes

An important goal of this paper is to organize large databases of shapes in a fashion that allows for efficient searches. One way of accomplishing this is to organize shapes in a tree structure, such that shapes are refined regularly as we move down the tree. In other words, objects are organized (clustered) according to coarser differences (in their shapes) at top levels and finer differences at lower levels. This is accomplished in a bottom up construction as follows: start with all the shapes at the bottom level and cluster them according to Algorithm 1 for a pre-determined k . Then, compute a mean shape for each cluster and at the next level cluster these mean shapes according to Algorithm 1. Applying this idea repeatedly, one obtains a tree organization of shapes in which shapes change from coarse to fine as we move down the tree. Critical to this organization is the notion of the mean of shapes for which we utilize Karcher means.

We follow the procedure above to generate an example of a tree structure obtained for 300 shapes selected from the Kimia database. The Figure 2 shows

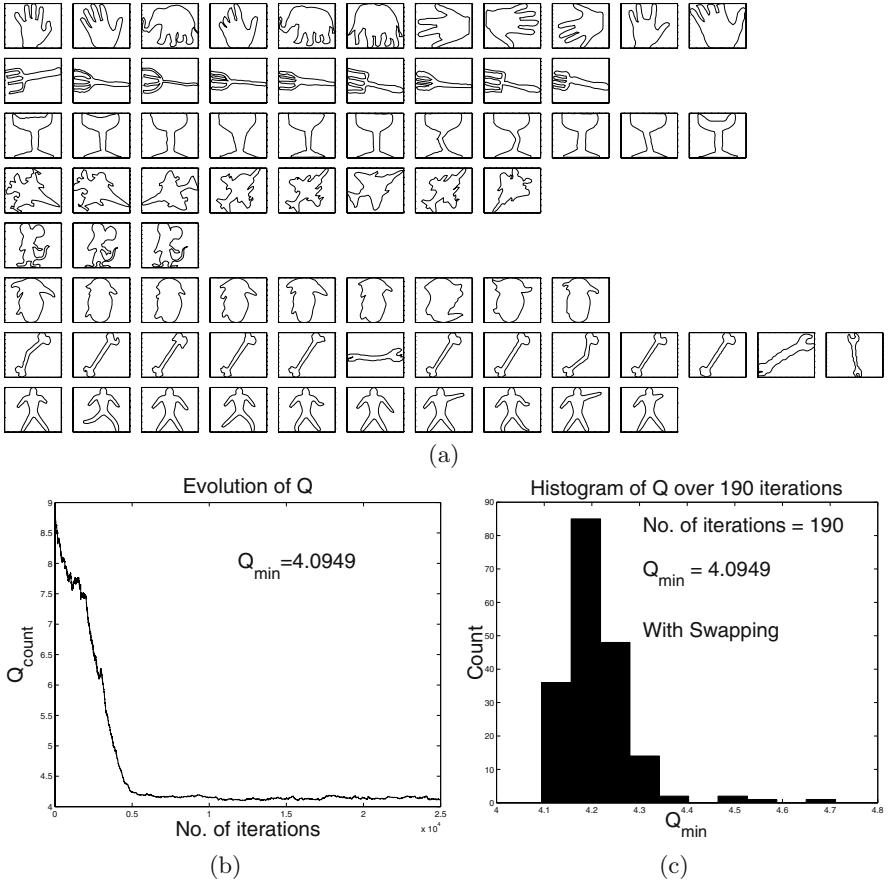


Fig. 1. (a) Samples from 26 clusters of the kimia dataset. Each row is a cluster. (b) Sample evolution of Algorithm 1 for the configuration in (a). (c) Histogram of $Q(C^*)$ for 190 runs.

a hierarchical organization of 300 shapes all the way up to the top. At the bottom level, these 300 shapes are clustered in $k = 26$ clusters, with the clusters denoted by the indices of their element shapes. Computing the means of each these clusters, we obtain shapes to be clustered at the next level. Repeating the clustering for $k = 9$ clusters we obtain the next level and their mean shapes. In this example, we have chosen to organize shapes in five levels with a single shape at the top. The choice of parameters such as the number of levels, and the number of clusters at each level, depends on the required search speed and performance. It is interesting to study the variations in shapes as we follow a path from top to bottom in this tree. Two such paths from the right tree are shown in Figure 3. This multi-resolution representation of shapes has important implications. One is that very different shapes can be efficiently compared at a low resolution while only similar shapes require high-resolution comparison.

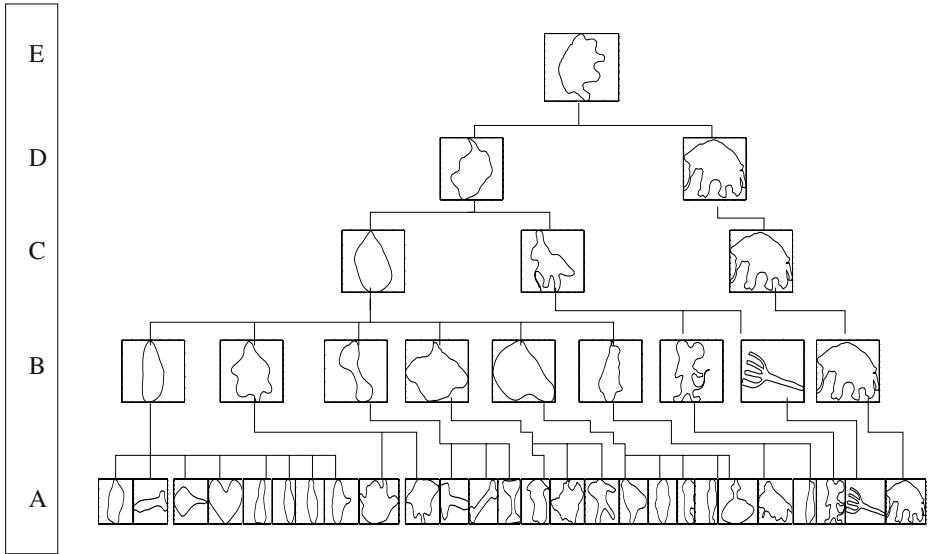


Fig. 2. Hierarchical Organization of 300 shapes from the Kimia database.

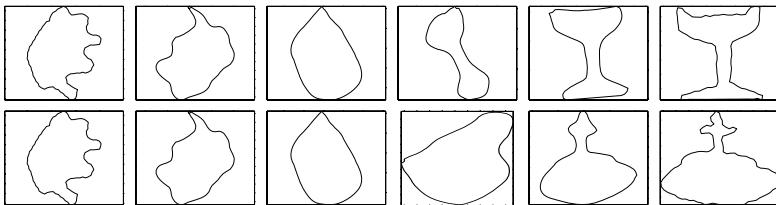


Fig. 3. Increasing resolution of shape as we go from top to bottom in Figure 2.

3 Problem 2: Shape Testing

An important step in statistical analysis of shapes is to impose probability models on shape spaces. Once the shapes are clustered, we are interested in learning models to capture shape variations. Since the shape space \mathcal{S} is curved and infinite-dimensional, we need some approximations to proceed further. We will impose probability models on \mathcal{S} implicitly by imposing models on $T_\mu(\mathcal{S})$, where μ is the mean shape, and inheriting it on \mathcal{S} using the geodesic flows. Secondly, we will approximate the element $f \in T_\mu(\mathcal{S})$ by its coefficients \mathbf{x} truncated to obtain a finite-dimensional representation.

We still have to decide what form does the resulting probability distribution takes. One common approach is to assume a parametric form so that learning is now simply an estimation of the relevant parameters. As an example, a popular idea is to assume a Gaussian distribution on the underlying space. Let \mathbf{x} be multivariate normal with mean $\mathbf{0}$ and variance $K \in \mathbb{R}^{m \times m}$. Estimation of μ and K from the observed shapes follows the usual procedures. Using μ and an

observed shape θ_j , we find the tangent vector $g_j \in T_\mu(\mathcal{S})$ such that the geodesic from μ in the direction g_j passes through θ_j in unit time. This tangent vector is actually computed implicitly through \mathbf{x}_j . From the observed values of \mathbf{x}_j s, one can estimate the covariance using sample covariance. Depending on the number and the nature of the shape observations, the rank of estimated covariance is generally much smaller than its size. Extracting the dominant eigen vectors of the estimated covariance matrix, one can capture the dominant modes of variations. The density function associated with this family of shapes is given by:

$$h(\theta; \mu, K) \equiv \frac{1}{(2\pi)^{m/2} \det(K^\epsilon)^{1/2}} \exp(-\mathbf{x}^T (K^\epsilon)^{-1} \mathbf{x}/2), \quad (3)$$

where $K^\epsilon = K + \epsilon I$, $\Psi(\mu, g, 1) = \theta$ and $g = \sum_{i=1}^m x_i e_i(s)$.

This framework of shape representations, and statistical models on shape spaces, has important applications in decision theory. One is to recognize an imaged object according to the shape of its boundary. Statistical analysis on shape spaces can be used to make a variety of decisions such as: Does this shape belong to a given family of shapes? Does these two families of shapes have similar means or variances? Given a test shape and two competing probability models, which one explains the test shape better?

We restrict to the case of binary hypothesis testing since for multiple hypotheses, one can find the best hypothesis using a sequence of binary hypothesis tests. Consider two shape families specified by their probability models: h_1 and h_2 . For an observed shape $\theta \in \mathcal{S}$, we are interested in selecting one of two following hypotheses: $H_0 : \theta \sim h_1$ or $H_1 : \theta \sim h_2$. We will select a hypothesis

$$H_1$$

according to the likelihood ratio test: $l(\theta) \equiv \log(\frac{h_1(\theta)}{h_2(\theta)}) \gtrless 0$. Substituting for the

$$H_0$$

normal distributions (Eqn. 3) for $h_1 \equiv h(\theta; \mu_1, \Sigma_1)$ and $h_2 \equiv h(\theta; \mu_2, \Sigma_2)$, we can obtain sufficient statistics for this test. Let \mathbf{x}_1 be the vector of Fourier coefficients that encode the tangent direction from μ_1 to θ , and \mathbf{x}_2 be the same for direction from μ_2 to θ . In other words, if we let $g_1 = \sum_{i=1}^m x_{1,i} e_i$ and $g_2 = \sum_{i=1}^m x_{2,i} e_i$, then we have $\theta = \Psi(\mu_1, g_1, 1) = \Psi(\mu_2, g_2, 1)$. It follows that

$$l(\theta) = (\mathbf{x}_1^T (\Sigma_1^\epsilon)^{-1} \mathbf{x}_1 - \mathbf{x}_2^T (\Sigma_2^\epsilon)^{-1} \mathbf{x}_2) - \frac{1}{2} (\log(\det(\Sigma_2^\epsilon)) - \log(\det(\Sigma_1^\epsilon))) \quad (4)$$

In case the two covariances are equal to Σ , the hypothesis test reduces to

$$l(\theta) = (\mathbf{x}_1^T (\Sigma^\epsilon)^{-1} \mathbf{x}_1 - \mathbf{x}_2^T \Sigma^\epsilon \mathbf{x}_2) \gtrless 0, \quad \begin{matrix} H_1 \\ H_0 \end{matrix}$$

and when Σ is identity, and $\epsilon = 0$, the log-likelihood ratio is given by $l(\theta) = \|\mathbf{x}_1\|^2 - \|\mathbf{x}_2\|^2$. The curved nature of the shape space \mathcal{S} makes the analysis of this test difficult. For instance, one may be interested in probability of type one error but that calculation requires a probability model on \mathbf{x}_2 when H_0 is true.

As a first order approximation, one can write $\mathbf{x}_2 \sim N(\bar{\mathbf{x}}, \Sigma_1)$, where $\bar{\mathbf{x}}$ is the coefficient vector of tangent direction in $T_{\mu_2}(\mathcal{S})$ that corresponds to the geodesic from μ_2 to μ_1 . However, the validity of this approximation remains to be tested under experimental conditions.

Shape Retrieval: We want to use the idea of hypothesis testing in retrieving shapes from a database that has been organized hierarchically. In view of its organization, a natural way is to start at the top, compare the query with the shapes at each level, and proceed down the branch that leads to the best match. At any level of the tree, there are some number, say p , of possible shapes, and our goal is to find the shape that matches the query θ best. This is performed using $(p-1)$ binary tests leading to the selection of the best hypothesis. Then, we proceed down the tree following that selected hypothesis and repeat the testing involving shapes at the next level. This continues till we reach the last level and have found the best overall match to the given query. For demonstration of retrieval, we hierarchically organize a collection of 100 shapes from the Surrey fish database [11] as shown in Figure 4.

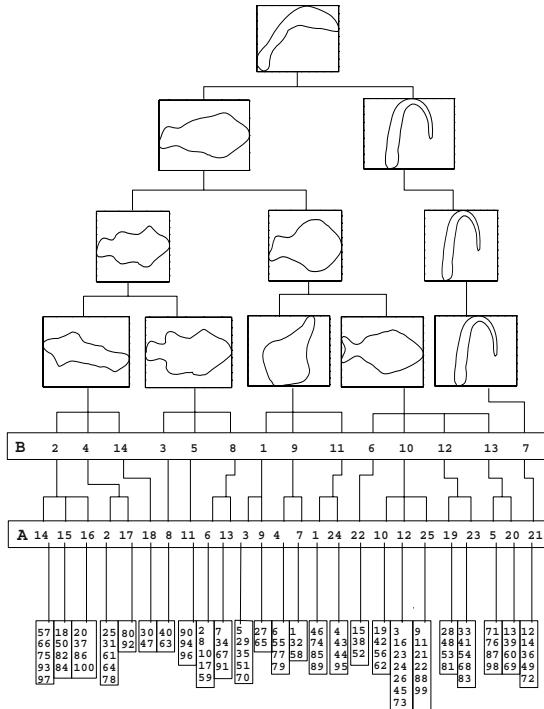


Fig. 4. Hierarchical Organization of 100 shapes from the Surrey Fish Database.

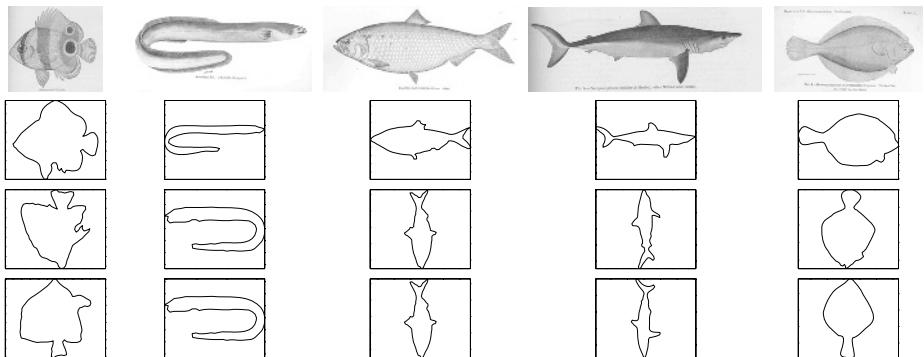


Fig. 5. Image retrieval. Top panels show the test images, the second row shows object boundaries, and the third row shows the closest shape in database obtained using a hierarchical search. Bottom row shows results of an exhaustive search.

Figure 5 shows some examples of the retrieval procedure. Shown in the top panels are images of test objects whose shapes provide queries for shape retrieval. These shapes are not included in the training database. First, we hand extract the contours of test objects (second row), and then use the shapes of these contours to search the database given in Figure 4. Third row shows the best matches from the database obtained using the tree search. For comparison, bottom row shows results from an exhaustive search over the full database. The resulting retrieval times are as follows. For exhaustive search, one needs 100 comparisons while for the organization shown in Figure 4, the required number of comparisons lies between seven and twenty. For example, in above figure, when the query is eel, it takes seven comparisons to reach the answer while for shad it takes seventeen comparisons. In our current implementation, each comparison takes 0.02 seconds approximately, providing a worst case retrieval time of 0.4 seconds for this database.

4 Conclusion

Using Riemannian structure of a space of planar shapes and geodesic length on it as a shape metric, we have presented statistical approaches to clustering and testing shapes. Clustering is performed by minimizing average variance within the clusters, and is used in hierarchically organizing large databases of objects according to their shapes. Using Gaussian distributions model on tangent spaces, we have derived a technique for shape testing, and have applied it to retrieval of shapes from a large database.

Acknowledgements. This material is based upon work supported by NSF DMS-0101429 and NMA 201-01-2010, and by NSF and the Intelligence Technology Innovation Center through the joint “Approaches to Combat Terrorism” Program Solicitation NSF 03-569 (DMS-0345242).

References

1. D. E. Brown and C. L. Huntley. A practical application of simulated annealing to clustering. Technical Report IPC-TR-91-003, Institute for Parallel Computing, University of Virginia, Charlottesville, VA, 1991.
2. D. Cremers and S. Soatto. A pseudo distance for shape priors in level set segmentation. In *2nd IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*, France, 2003.
3. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Son, 1998.
4. M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 2002.
5. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.
6. P. Maurel and G. Sapiro. Dynamic shapes average. In *2nd IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*, France, 2003.
7. M. I. Miller and L. Younes. Group actions, homeomorphisms, and matching: A general framework. *International Journal of Computer Vision*, 41(1/2):61–84, 2002.
8. C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Text in Stat., 1999.
9. K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, 86(11):2210–2239, November 1998.
10. E. Klassen, A. Srivastava, W. Mio and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Pattern Analysis and Machiner Intelligence*, to appear in March 2004.
11. F. Mokhtarian, S. Abbasi and J. Kittler. Efficient and robust shape retrieval by shape content through curvature scale space. *Proceedings of First International Conference on Image Database and MultiSearch*, 1996
12. Sharvit, D. and Chan, J. and Tek, H. and Kimia, B.B. Symmetry-based indexing of image databases. *Content-Based Access of Image and Video Libraries*, 1998

Intrinsic Images by Entropy Minimization

Graham D. Finlayson¹, Mark S. Drew², and Cheng Lu²

¹ School of Information Systems, The University of East Anglia
Norwich, England NR4 7TJ
graham@sys.uea.ac.uk

² School of Computing Science, Simon Fraser University
Vancouver, British Columbia, Canada V5A 1S6
{mark,clu}@cs.sfu.ca

Abstract. A method was recently devised for the recovery of an invariant image from a 3-band colour image. The invariant image, originally 1D greyscale but here derived as a 2D chromaticity, is independent of lighting, and also has shading removed: it forms an *intrinsic image* that may be used as a guide in recovering colour images that are independent of illumination conditions. Invariance to illuminant colour and intensity means that such images are free of shadows, as well, to a good degree. The method devised finds an intrinsic reflectivity image based on assumptions of Lambertian reflectance, approximately Planckian lighting, and fairly narrowband camera sensors. Nevertheless, the method works well when these assumptions do not hold. A crucial piece of information is the angle for an “invariant direction” in a log-chromaticity space. To date, we have gleaned this information via a preliminary calibration routine, using the camera involved to capture images of a colour target under different lights. In this paper, we show that we can in fact dispense with the calibration step, by recognizing a simple but important fact: the correct projection is *that which minimizes entropy* in the resulting invariant image. To show that this must be the case we first consider synthetic images, and then apply the method to real images. We show that not only does a correct shadow-free image emerge, but also that the angle found agrees with that recovered from a calibration. As a result, we can find shadow-free images for images with unknown camera, and the method is applied successfully to remove shadows from unsourced imagery.

1 Introduction

Recently, a new image processing procedure was devised for creating an illumination-invariant, intrinsic, image from an input colour image [1,2,3,4]. Illumination conditions cause problems for many computer vision algorithms. In particular, shadows in an image can cause segmentation, tracking, or recognition algorithms to fail. An illumination-invariant image is of great utility in a wide range of problems in both Computer Vision and Computer Graphics. However, to find the invariant image, calibration is needed and this limits the applicability of the method. In this paper we show a surprising result: an intrinsic image can be found without calibration even when nothing is known about the image.

To date, the method in essence rests on a kind of calibration scheme for a particular colour camera. How one proceeds is by imaging a target composed of colour patches

(or, possibly, just a rather colourful scene). Images are captured under differing lightings — the more illuminants the better. Then knowledge that all these images are registered images of the same scene, under differing lighting, is put to use by plotting the capture RGB values, for each of the pixels used, as the lighting changes. If pixels are first transformed from 3D RGB triples into a 2D chromaticity colour space $\{G/R, B/R\}$, and then logarithms are taken, the values across different lighting tend to fall on straight lines in a 2D scatter plot. And in fact all such lines are parallel, for a given camera.

If change of illumination simply amounts to movement along such a line, then it is straightforward to devise a 1D illumination-invariant image by projecting the 2D chromaticity points into a direction perpendicular to all such lines. The result is hence a greyscale image that is independent of lighting. In a sense, therefore, it is an intrinsic image that portrays only the inherent reflectance properties in the scene. Since shadows are mostly due to removal of some of the lighting, such an image also has shadows removed.

We can also use the greyscale, invariant, image as a guide that allows us to determine which colours in the original, RGB, colour image are intrinsic to the scene or are simply artifacts of the shadows due to lighting. Forming a gradient of the image's colour channels, we can guide a thresholding step via the difference between edges in the original and in the invariant image [3]. Forming a further derivative, and then integrating back, we can produce a result that is a 3-band *colour* image which contains all the original salient information in the image, except that the shadows are removed. Although this method is based on the greyscale invariant image developed in [1], which produces an invariant image which does have shading removed, it is of interest because its output is a colour image, including shading. In another approach [4], a 2D-colour chromaticity invariant image is recovered by projecting orthogonal to the lighting direction and then putting back an appropriate amount of lighting. Here we develop a similar chromaticity illumination-invariant image which is more well-behaved and thus gives better shadow removal.

For Computer Vision purposes, in fact an image that includes shading is not always required, and may confound certain algorithms — the unreal look of a chromaticity image without shading is inappropriate for human understanding but excellent for machine vision (see, e.g., [5] for an object tracking application, resistant to shadows).

The problem we consider, and solve, in this paper is the determination of the invariant image from *unsourced* imagery — images that arise from cameras that are *not calibrated*. The input is a colour image with unknown provenance, one that includes shadows, and the output is the invariant chromaticity version, with shading and shadows removed.

To see how we do this let us remember how we find the intrinsic image for the calibrated case. This is achieved by plotting 2D log-chromaticities as lighting is changed and observing the direction in which the resulting straight lines point — the “invariant direction” — and then projecting in this direction. The key idea in this paper is the observation that, without having to image a scene under more than a single illuminant, projecting in the correct direction *minimizes the entropy* in the resulting greyscale image. The intuition behind this statement is evident if one thinks of a set of colour patches under changing lighting. As lighting changes, for each colour patch, pixels occupy an approximately straight line in a 2D log-chromaticity space. If we project all these pixels onto a line perpendicular to the set of straight lines, we end up with a set of 1D points, as in Fig. 1(a). In a set of real images of colour patches, we would expect a set of peaks, each well separated from the others and corresponding to a single colour patch. On the

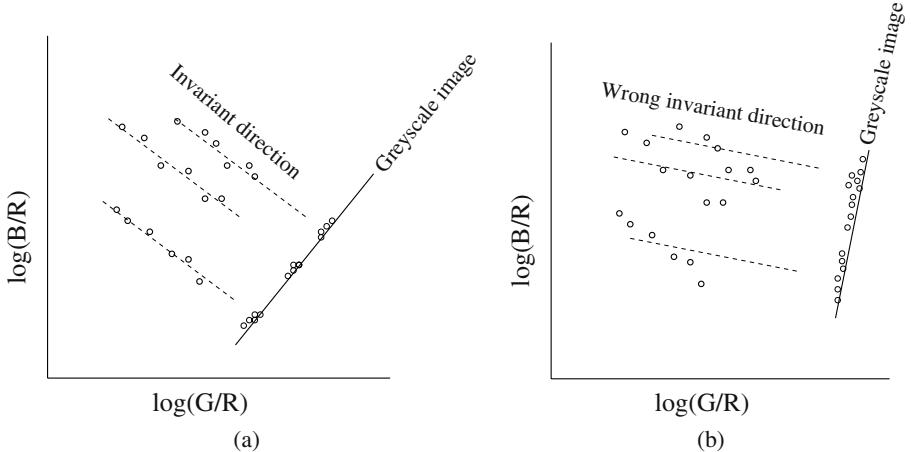


Fig. 1. Intuition for finding best direction via minimizing the entropy.

other hand, if we instead project in some other direction, as in Fig. 1(b), then instead of pixels located in sharp peaks of occurrence we expect the distribution of pixels along our 1D projection line to be spread out. In terms of histograms, in the first instance, in which we guess the correct direction and then project, we see a distribution with a set of sharp peaks, with resulting low entropy. In the second instance we instead see a broader histogram, with resulting higher entropy.

Hence the idea in this paper is to recover the correct direction in which to project by examining the entropy of a greyscale image that results from projection and identifying as the correct “invariant direction” that which minimizes the entropy of the resulting image. Changing lighting is automatically provided by the shadows in the image themselves.

In §2, we first recapitulate the problem of lighting change in imagery, along with the accompanying theory of image formation. The method of deriving an invariant image is given, for known invariant direction, for imagery that was captured using a calibrated camera. Now, without any calibration or foreknowledge of the invariant direction, in §3.1 we create a synthetic “image” that consists of a great many colour patches. Since the image is synthetic, we in fact do know the ground truth invariant direction. Examining the question of how to *recover* this direction from a single image, with no prior information, we show that minimizing the entropy provides a very strong indicator for determining the correct projection. For a synthetic image, results are very good indeed. This result provides a proof in principle for the entropy-minimizing method.

But how do we fare with a real camera? In §3.2 we consider a set of calibration images, taken with a known camera. Since we control the camera, and the target, we can establish the invariant direction. Then comparing to the direction recovered using entropy minimization, we find that not only is the direction of projection recovered correct (within 3 degrees), but also the minimum is global and is a very strong signal — essentially, Nature is telling us that this is indeed the way to go: entropy minimization is a new and salient indicator for the projection that removes shadows.

Real, non-synthesized, images are noisy and might not provide such a clean picture. Nevertheless, by examining real images in § 4, we arrive at a set of steps that will correctly deliver the intrinsic image, without calibration. Finally, we apply the method devised to unsourced images, from unknown cameras under unknown lighting, with unknown processing applied. Results are again strikingly good, leading us to conclude, in § 5, that the method indeed holds great promise for developing a stand-alone approach to removing shadows from (and therefore conceivably re-lighting) any image, e.g. images consumers take to the neighbourhood processing lab.

2 Theory of Invariant Image Formation

2.1 Planckian Lighting, Lambertian Surfaces, Narrowband Camera

Suppose we consider a fairly narrow-band camera, with three sensors, Red, Green, and Blue, as in Fig. 3(a); these are sensor curves for the Sony DXC930 camera. Now if we image a set of coloured Lambertian surfaces under a particular Planckian light, in a controlled light box, say, then for each pixel the log of the band-ratios $\{R/G, B/G\}$ appears as a dot in a 2D plot. Chromaticity removes shading, for Lambertian reflectances under orthography, so every pixel in each patch is approximately collapsed into the same dot (no matter if the surface is curved). Fig. 2(b) shows the log-chromaticities for the 24 surfaces of the Macbeth ColorChecker Chart shown in Fig. 2(a) (the six neutral patches all belong to the same cluster). These images were captured using an experimental HP912 Digital Still Camera, modified to generate linear output.

For narrow-band sensors (or spectrally-sharpened ones [6]), and for Planckian lights (or lights such as Daylights which behave as if they were Planckian), as the illuminant temperature T changes, the log-chromaticity colour 2-vector moves along an approximately straight line which is independent of the magnitude and direction of the lighting. Fig. 2(c) illustrates this for 6 of the patches: the plot is for the same 6 patches imaged under a range of different illuminants. In fact, the camera sensors are not exactly narrow-band and the log-chromaticity line is only approximately straight. Assuming that the change with illumination is indeed linear, projecting colours perpendicular to this “invariant direction” due to lighting change produces a 1D greyscale image that is invariant to illumination. Note that the invariant direction is different for each camera; it

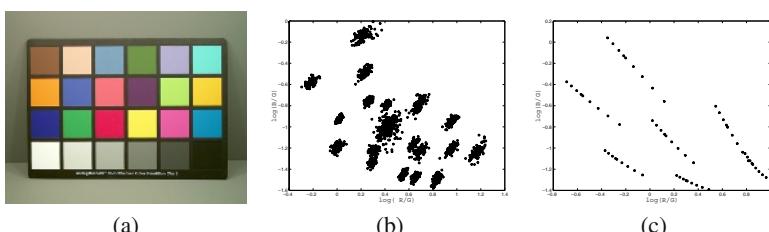


Fig. 2. (a): Macbeth ColorChecker Chart image under a Planckian light. (b): Log-chromaticities of the 24 patches. (c): Median chromaticities for 6 patches, imaged under 14 different Planckian illuminants.

can be recovered from a calibration with plots such as Fig. 2(c). In this paper, we mean to recover the correct direction from a single image, and with no calibration.

Let's recapitulate how this linear behaviour with lighting change results from the assumptions of Planckian lighting, Lambertian surfaces, and a narrowband camera. Consider the RGB colour \mathbf{R} formed at a pixel, for illumination with spectral power distribution $E(\lambda)$ impinging on a surface with surface spectral reflectance function $S(\lambda)$. If the three camera sensor sensitivity functions form a set $\mathbf{Q}(\lambda)$, then we have

$$R_k = \sigma \int E(\lambda)S(\lambda)Q_k(\lambda)d\lambda, \quad k = R, G, B, \quad (1)$$

where σ is Lambertian shading: surface normal dotted into illumination direction.

If the camera sensor $Q_k(\lambda)$ is exactly a Dirac delta function $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, then eq. (1) becomes simply

$$R_k = \sigma E(\lambda_k)S(\lambda_k)q_k. \quad (2)$$

Now suppose lighting can be approximated by Planck's law, in Wien's approximation [7]:

$$E(\lambda, T) \simeq I k_1 \lambda^{-5} e^{-\frac{k_2}{T\lambda}}, \quad (3)$$

with constants k_1 and k_2 . Temperature T characterizes the lighting colour and I gives the overall light intensity.

In this approximation, from (2) the RGB colour R_k , $k = 1 \dots 3$, is simply given by

$$R_k = \sigma I k_1 \lambda_k^{-5} e^{-\frac{k_2}{T\lambda_k}} S(\lambda_k) q_k. \quad (4)$$

Let us now form the band-ratio 2-vector chromaticities \mathbf{c} ,

$$c_k = R_k / R_p, \quad (5)$$

where p is one of the channels and $k = 1, 2$ indexes over the remaining responses. We could use $p = 1$ (i.e., divide by Red) and so calculate $c_1 = G/R$ and $c_2 = B/R$. We see from eq. (4) that forming the chromaticity effectively removes intensity and shading information. If we now form the log of (5), with $s_k \equiv k_1 \lambda_k^{-5} S(\lambda_k) q_k$ and $e_k \equiv -k_2/\lambda_k$ we obtain

$$\rho_k \equiv \log(c_k) = \log(s_k/s_p) + (e_k - e_p)/T. \quad (6)$$

Eq. (6) is a straight line parameterized by T . Notice that the 2-vector direction $(e_k - e_p)$ is *independent of the surface*, although the line for a particular surface has offset that depends on s_k .

An invariant image can be formed by projecting 2D logs of chromaticity, ρ_k , $k = 1, 2$, into the direction \mathbf{e}^\perp orthogonal to the vector $\mathbf{e} \equiv (e_k - e_p)$. The result of this projection is a single scalar which we then code as a greyscale value.

The utility of this invariant image is that since shadows derive in large part from lighting that has a different intensity and colour (temperature T) from lighting that impinges in non-shadowed parts of the scene, shadows are effectively removed by this projection. Before light is added back to such images, they are intrinsic images bearing

reflectivity information only. Below, in § 3.2, we recover an approximate intrinsic RGB reflectivity, as in [8] but with a considerably less complex algorithm.

Clearly, if we calibrate a camera by determining the invariant 2-vector direction e then we know in advance that projecting in direction e^\perp produces the invariant image. To do so, we find the minimum-variance direction of mean-subtracted values ρ for target colour patches [1]. However, if we have a single image, then we do not have the opportunity to calibrate. Nevertheless if we have an image with unknown source we would still like to be able to remove shadows from it. We show in the next Section that the automatic determination of the invariant direction is indeed possible, with entropy minimization being the correct mechanism.

3 Intrinsic Images by Entropy Minimization

3.1 Entropy Minimization

If we wished to find the minimum-variance direction for lines in Fig. 1, we would need to know which points fall on which lines. But what if we did not have that information? Entropy minimization is the key to finding the right invariant direction.

To test the idea that entropy minimization gives an intrinsic image, suppose we start with a theoretical Dirac-delta sensor camera, as in Fig. 3(b). Now let us synthesize an “image” that consists of many measured natural surface reflectance functions interacting with many lights, in turn, and then imaged by our theoretical camera. As a test, we use the reflectance data $S(\lambda)$ for 170 natural objects, measured by Vrhel et al. [9]. For lights, we use the 9 Planckian illuminants $E(\lambda)$ with T from $2,500^\circ$ to $10,500^\circ$ Kelvin with interval of $1,000^\circ$. Thus we have an image composed of 1,530 different illuminant-reflectance colour signal products.

If we form chromaticities (actually we use geometric mean chromaticities defined in eq. (7) below), then taking logarithms and plotting we have 9 points (for our 9 lights) for every colour patch. Subtracting the mean from each 9-point set, all lines go through the origin. Then it is trivial to find the best direction describing all 170 lines via applying the Singular Value Decomposition method to this data. The best direction line is found at angle 68.89° . And in fact we know from theory that this angle is correct, for this camera. This verifies the straight-line equation (6), in this situation where the camera and surfaces exactly obey our assumptions. This exercise amounts, then, to a calibration of our theoretical camera in terms of the invariant direction.

But now suppose we do not know that the best angle at which to project our theoretical data is orthogonal to about 69° — how can we recover this information? Clearly, in this theoretical situation, the intuition displayed in Fig. 1 can be brought into play by simply traversing all possible projection angles that produce a projection direction e^\perp : *the direction that generates an invariant image with minimum entropy is the correct angle.*

To carry out such a comparison, we simply rotate from 0° to 180° and project the log-chromaticity image 2-vector ρ into that direction. A histogram is then formed (we used 64 equally-spaced bins). And finally the entropy is calculated: the histogram is divided by the sum of the bin counts to form probabilities p_i and, for bins that are occupied, the sum of $-p_i \log_2 p_i$ is formed.

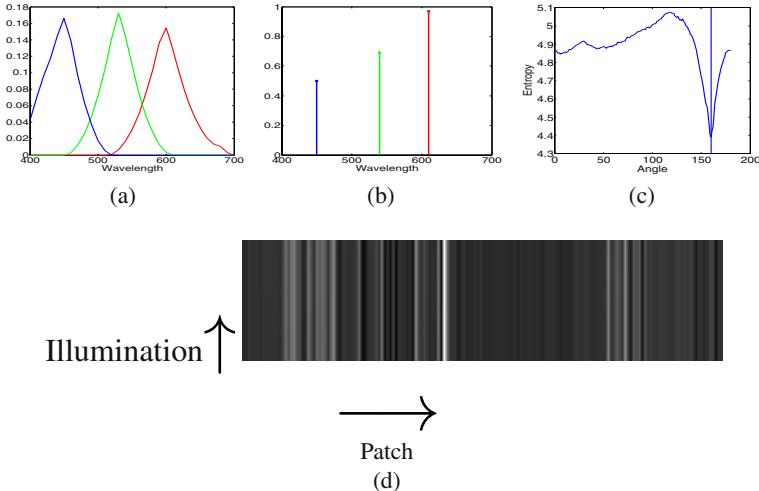


Fig. 3. (a): Typical RGB camera sensors — Sony DXC930 camera. (b): Theoretical narrowband RGB camera sensors. (c): Minimum entropy invariant direction gives same angle as calibration test. (d): Invariant image for theoretical synthetic image — same greylevels across illuminants.

Fig. 3(c) shows a plot of angle versus this entropy measure, for the synthetic image. As can be seen, the correct angle of $159 = 90 + 69^\circ$ is accurately determined (within a degree). Fig. 3(d) shows the actual “image” for these theoretical colour patches, given by exponentiating the projected log-image.

As we go from left to right across Fig. 3(d) we change reflectance. From top to bottom we have pixels calculated with respect to different lights. Because the figure shows the invariant image coded as a grey scale there is very little variation from top to bottom. Yet the greyscale value does change from left to right. So, in summary, Fig. 3(d) tells us that the same surface has the same invariant across lights but different surfaces have different invariants (and so the intrinsic image conveys useful reflectance information).

Next, we consider an “image” formed from measured calibration values of a colour target, as in Fig. 2.

3.2 Calibration Images versus Entropy Minimization

Now let us investigate how this theoretical method can be used for real, non-synthetic images. We already have acquired calibration images, such as Fig. 2(a), over 14 phases of daylight. These images are taken with an experimental HP 912 digital camera with the normal nonlinear processing software disabled.

Geometric Mean Invariant Image. From (4), we can remove σ and I via division by any colour channel: but which channel should we use? If we divide by red, but red happens to be everywhere small, as in a photo of greenery, say, we’re in trouble. A better solution is to divide by the geometric mean [2], $\sqrt[3]{R \times G \times B}$. Then we still retain our straight line in log space, but do not favour one particular channel.

Thus we amend our definitions (5, 6) of chromaticity as follows:

$$c_k = R_k / \sqrt[3]{\prod_{i=1}^3 R_i}, \quad \equiv \quad R_k / R_M, \quad (7)$$

and log version [2]

$$\begin{aligned} \rho_k &= \log(c_k) = \log(s_k/s_M) + (e_k - e_M)/T, \quad k = 1..3, \text{ with} \\ s_k &= k_1 \lambda_k^{-5} S(\lambda_k) q_k, \quad s_M = \sqrt[3]{\prod_{j=1}^3 s_j}, \quad e_k = -k_2/\lambda_k, \quad e_M = -k_2/3 \sum_{j=1}^p \lambda_j, \end{aligned} \quad (8)$$

and for the moment we carry all three (thus nonindependent) components of chromaticity. Broadband camera versions are stated in [2].

Geometric Mean 2-D Chromaticity Space. We should use a 2D chromaticity space that is appropriate for this color space ρ . We note that, in log space, ρ is orthogonal to $u = 1/\sqrt{3}(1, 1, 1)^T$. I.e., ρ lives on a plane orthogonal to u , as in Fig. 4, $\rho \cdot u = 0$. To characterize the 2D space, we can consider the projector P_u^\perp onto the plane.

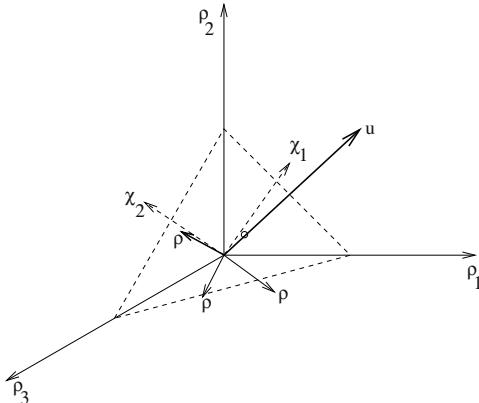


Fig. 4. Geometric mean divisor means every ρ is orthogonal to u . Basis in plane is $\{\chi_1, \chi_2\}$.

P_u^\perp has two non-zero eigenvalues, so its decomposition reads

$$P_u^\perp = I - u u^T = U^T U, \quad (9)$$

where U is a 2×3 orthogonal matrix. U rotates 3-vectors ρ into a coordinate system in the plane:

$$\chi \equiv U \rho, \quad \chi \text{ is } 2 \times 1. \quad (10)$$

Straight lines in ρ are still straight in χ .

In the $\{\chi_1, \chi_2\}$ plane, we are now back to a situation similar to that in Fig. 1: we must find the correct direction θ in which to project, in the plane, such that the entropy for the marginal distribution along a 1D projection line orthogonal to the lighting direction is minimized. The greyscale image \mathcal{I} along this line is formed via

$$\mathcal{I} = \chi_1 \cos \theta + \chi_2 \sin \theta \quad (11)$$

and the entropy is given by

$$\eta = - \sum_i p_i(\mathcal{I}) \log(p_i(\mathcal{I})). \quad (12)$$

Main Idea. Thus the heart of the method is as follows:

1. Form a 2D log-chromaticity representation of the image.
2. for $\theta = 1..180$
 - a) Form greyscale image \mathcal{I} : the projection onto 1D direction.
 - b) Calculate entropy.
 - c) Min-entropy direction is correct projection for shadow removal.

3-Vector Representation. After we find θ , we can go back to a 3-vector representation of points on the projection line via the 2×2 projector P_θ : we form the projected 2-vector χ_θ via $\chi_\theta = P_\theta \chi$ and then back to an estimate (indicated by a tilde) of 3D ρ and c via $\tilde{\rho} = U^T \chi_\theta$, $\tilde{c} = \exp(\tilde{\rho})$. For display, we would like to move from an intrinsic image, governed by reflectivity, to one that includes illumination (cf. [4]). So we add back enough e so that the median of the brightest 1% of the pixels has the 2D chromaticity of the original image: $\chi_\theta \rightarrow \chi_\theta + \chi_{extralight}$.

Entropy Minimization — Strong Indicator. From the calibration technique described in section 3.1 we in fact already know the *correct* characteristic direction in which to project to attenuate illumination effects: for the HP-912 camera, this angle turns out to be 158.5° . We find that entropy minimization gives a close approximation of this result: 161° .

First, transforming to 2D chromaticity coordinates χ , the colour patches of the target do form a scatterplot with approximately parallel lines, in Fig. 5(a). We compose an “image” consisting of a montage of median pixels for all 24 colour patches and 14 lights. The calculation of entropy carried out for this image gives a very strong minimum, shown in Fig. 5(b), and excellent greyscale \mathcal{I} invariant to lighting in Fig. 5(c).

In the next section, we examine the issues involved when we extend this theoretical success to the realm of real non-calibration images.

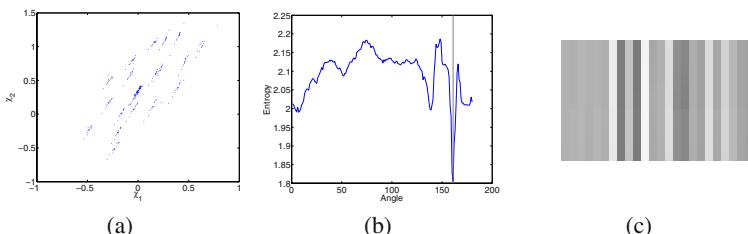


Fig. 5. (a): 2D chromaticity for measured colour patches, HP 912 camera. (b): Minimum entropy invariant direction gives angle close to that of calibration method. (c): Invariant image for measured patch values — projected greylevels same for different illuminants.

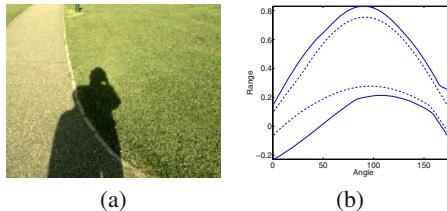


Fig. 6. (a): Input colour image, captured with HP912 Digital Still Camera (with linear output). (b): The range of projected data changes with angle. Range: solid lines; 5th and 95th percentiles: dashed lines.

4 Intrinsic Image Recovery Algorithm

4.1 Algorithm Steps

Consider the colour image in Fig. 6(a): two people are illuminated from behind by strong sunlight. As well, there is a skylight illumination component that creates non-zero RGBs in the shadow region. Here, we have a calibrated camera, so we'll know if entropy minimization produces the correct answer. To find the minimum entropy, we again examine projections \mathcal{I} over angles 0° to 180° , for log-chromaticities χ formed according to eqs. (7), (8), and (10). For each angle, we project the log-chromaticity, and then determine the entropy (12). However, the nature of the data, for real images, presents an inherent problem. Since we are considering ratios, we can expect noise to possibly be enhanced (although this is mitigated by the sum in eq. (14)). To begin with, therefore, we apply Gaussian smoothing to the original image colour channels. But even so, we expect that some ratios may be large. So the question remains as to what we should use as the range, and number of bins, in a histogram of a projected greyscale image \mathcal{I} .

To begin with, then, we can determine the range of invariant image greyscale values, for each candidate projection angle. Fig. 6(b) shows a plot of this range, versus projection angle. The figure also shows the range, dashed, of the 5-percentile and 95-percentile lines. We can see that the full range contains many outliers. Therefore it makes sense to exclude these outliers from consideration.

Hence we use the middle values only, i.e., the middle 90% of the data, to form a histogram. To form an appropriate bin width, we utilize Scott's Rule [10]:

$$\text{bin_width} = 3.5 \text{ std}(\text{projected data}) N^{1/3} \quad (13)$$

where N is the size of the invariant image data, for the current angle. Note that this size is different for each angle, since we exclude outliers differently for each projection.

The entropy calculated is shown in Fig. 7(a). The minimum entropy occurs at angle 156° . For the camera which captures the images, in fact we have calibration images using a Macbeth ColorChecker. From these, we determined that the correct invariant direction is actually 158.5° , so we have done quite well, without any calibration, by minimizing entropy instead. The figure shows that the minimum is a relatively strong dip, although not as strong as for the theoretical synthetic image.

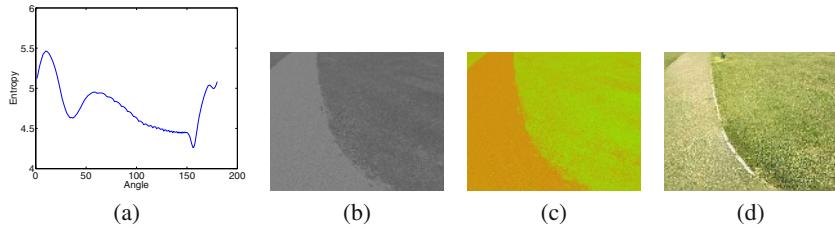


Fig. 7. (a): Entropy of projected image, versus projection angle. (b): Greyscale invariant image \mathcal{I} , at minimum entropy direction. (See <http://www.cs.sfu.ca/~mark/ftp/Eccv04/> for a video of images as the projection angle changes, with shadows disappearing.) (c): Invariant L_1 chromaticity image r . (d): Re-integrated RGB colour image.

Once we have an estimate \tilde{c} of the geometric-mean chromaticity (7), we can also go over to the more familiar L_1 -based chromaticity $\{r, g, b\}$, defined as

$$\mathbf{r} = \{r, g, b\} = \{R, G, B\}/(R + G + B), \quad r + g + b \equiv 1. \quad (14)$$

This is the most familiar representation of colour independent of magnitude. Column 2 of (Fig. 8 shows the L_1 chromaticity for colour images.) To obtain L_1 chromaticity r from c , we simply take

$$\tilde{\mathbf{r}} = \tilde{\mathbf{c}} / \sum_{k=1}^3 \tilde{c}_k. \quad (15)$$

Since r is bounded $\in [0, 1]$, invariant images in r are better-behaved than is \mathcal{I} . The greyscale image \mathcal{I} for this test is shown in Fig. 7(b), and the L_1 chromaticity version r , as per eq. (15), is shown in Fig. 7(c).

Using a re-integration method similar to that in [3], we can go on to recover a full-colour shadow-free image, as in Fig. 7(d). The method [3] uses a shadow-edge map, derived from comparing the original edges to those in the invariant image. Here we use edges from the invariant *chromaticity* image Fig. 7(c), and compare to edges from a Mean-Shift [11] processed original image. As well, rather than simply zeroing edges across the shadow edge, here we use a simple form of in-filling to grow edges into shadow-edge regions. Regaining a full-colour image has two components: finding a shadow-edge mask, and then re-integrating. The first step is carried out by comparing edges in the Mean-Shift processed original image with the corresponding recovered invariant chromaticity image. We look for pixels that have edge values higher than a threshold for any channel in the original, and lower than another threshold in the invariant, shadow-free chromaticity. We identify these as shadow edges, and then thicken them using a morphological operator. For the second stage, for each log colour channel, we first grow simple gradient-based edges across the shadow-edge mask using iterative dilation of the mask and replacement of unknown derivative values by the mean of known ones. Then we form a second derivative, go to Fourier space, divide by the Laplacian operator transform, and go back to x, y space. Neumann boundary conditions leave an additive constant unknown in each recovered log colour, so we regress on the top brightness quartile of pixel values to arrive at the final resulting colour planes.

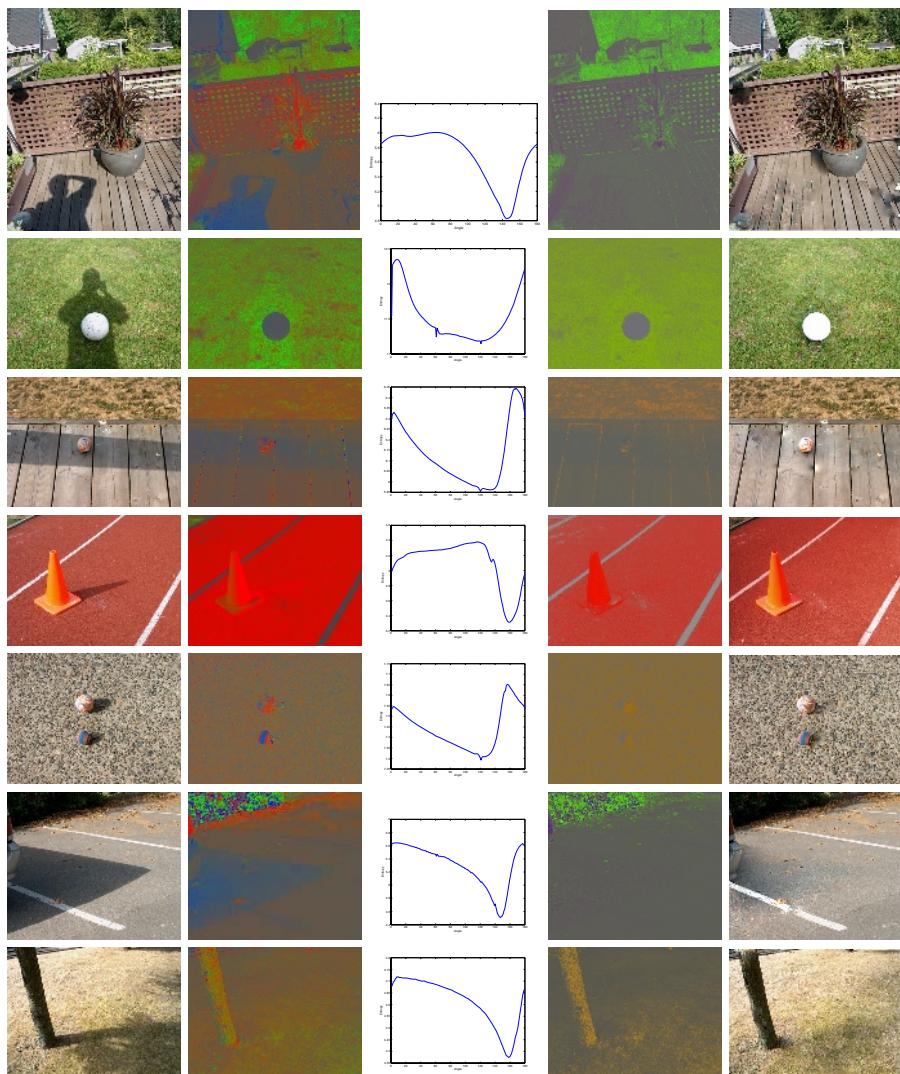


Fig. 8. Additional invariant images, for minimum entropy: columns show original image, L_1 chromaticity image, entropy plot, invariant L_1 chromaticity, and re-integrated colour image. More images are shown at <http://www.cs.sfu.ca/~mark/ftp/Eccv04/>

Other images from the known camera show similar behaviour, usually with strong entropy minima, and shadow-free results very close to those in [3]. Minimum-entropy angles have values from 147° to 161° for the same camera, with 158.5° being correct. Both in terms of recovering the correct invariant direction and in terms of generating a good, shadow-free, invariant image, our intuition that minimization of entropy would lead to correct results is indeed justified.

4.2 Images from an Unknown Camera

Fig. 8 shows results from uncalibrated images, from a consumer HP618 camera. In every case tried, entropy minimization provides a strong guiding principle for removing shadows.

5 Conclusions

We have presented a method for finding the invariant direction, and thus a greyscale and thence an L_1 -chromaticity intrinsic image that is free of shadows, without any need for a calibration step or special knowledge about an image. The method appears to work well, and leads to good re-integrated full-colour images with shadows greatly attenuated.

Future work would involve a careful assessment of how onboard nonlinear processing in cameras affects results. Cameras ordinarily supply images and videos that are compressed, as well as greatly processed away from being linear images. Although the method does indeed work under such processing (see Fig. 8) it would be well to understand how JPEG artifacts impact the method.

For the re-integration step, application of a curl-correction method [12] to ensure integrability would likely be of benefit. Also, if may be the case that consideration of a separate shadow-edge map for x and y could be useful, since in principle these are different. A variational in-filling algorithm would likely work better than our present simple morphological edge-diffusion method for crossing shadow-edges, but would be slower.

The ultimate goal of this work is automated processing of unsourced imagery such that shadows are removed. Results to date have indicated that, at the least, such processing can remove shadows and as well tends to “clean up” portraiture such that faces, for example, look more appealing after processing.

References

1. G.D. Finlayson and S.D. Hordley. Color constancy at a pixel. *J. Opt. Soc. Am. A*, 18(2):253–264, Feb. 2001. Also, UK Patent #2360660, “Colour signal processing which removes illuminant colour temperature dependency”.
2. G.D. Finlayson and M.S. Drew. 4-sensor camera calibration for image representation invariant to shading, shadows, lighting, and specularities. In *ICCV’01: International Conference on Computer Vision*, pages II: 473–480. IEEE, 2001.
3. G.D. Finlayson, S.D. Hordley, and M.S. Drew. Removing shadows from images. In *ECCV 2002: European Conference on Computer Vision*, pages 4:823–836, 2002. Lecture Notes in Computer Science Vol. 2353, <http://www.cs.sfu.ca/~mark/ftp/Eccv02/shadowless.pdf>.
4. M.S. Drew, G.D. Finlayson, and S.D. Hordley. Recovery of chromaticity image free from shadows via illumination invariance. In *IEEE Workshop on Color and Photometric Methods in Computer Vision, ICCV’03*, pages 32–39, 2003.
<http://www.cs.sfu.ca/~mark/ftp/Iccv03ColorWkshp/iccv03wkshp.pdf>.
5. H. Jiang and M.S. Drew. Shadow-resistant tracking in video. In *ICME’03: Intl. Conf. on Multimedia and Expo*, pages III 77–80, 2003. <http://www.cs.sfu.ca/~mark/ftp/Icme03/icme03.pdf>.
6. G.D. Finlayson, M.S. Drew, and B.V. Funt. Spectral sharpening: sensor transformations for improved color constancy. *J. Opt. Soc. Am. A*, 11(5):1553–1563, May 1994.

7. G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. Wiley, New York, 2nd edition, 1982.
8. M.F. Tappen, W.T. Freeman, and E.H. Adelson. Recovering intrinsic images from a single image. In *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.
9. M.J. Vrhel, R. Gershon, and L.S. Iwan. Measurement and analysis of object reflectance spectra. *Color Research and Application*, 19:4–9, 1994.
10. D.W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. Wiley and Kegan Paul, 1992.
11. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24:603–619, 2002.
12. B. V. Funt, M. S. Drew, and M. Brockington. Recovering shading from color images. In G. Sandini, editor, *ECCV-92: Second European Conference on Computer Vision*, pages 124–132. Springer-Verlag, May 1992.

Image Similarity Using Mutual Information of Regions

Daniel B. Russakoff^{1,2}, Carlo Tomasi³,
Torsten Rohlfing², and Calvin R. Maurer, Jr.²

¹ Department of Computer Science, Stanford University, Stanford CA 94305, USA
`daniel.russakoff@cs.stanford.edu`

² Image Guidance Laboratories, Stanford University, Stanford CA 94305, USA

³ Department of Computer Science, Duke University, Durham NC 27708, USA

Abstract. Mutual information (MI) has emerged in recent years as an effective similarity measure for comparing images. One drawback of MI, however, is that it is calculated on a pixel by pixel basis, meaning that it takes into account only the relationships between corresponding individual pixels and not those of each pixel's respective neighborhood. As a result, much of the spatial information inherent in images is not utilized. In this paper, we propose a novel extension to MI called regional mutual information (RMI). This extension efficiently takes neighborhood regions of corresponding pixels into account. We demonstrate the usefulness of RMI by applying it to a real-world problem in the medical domain—intensity-based 2D-3D registration of X-ray projection images (2D) to a CT image (3D). Using a gold-standard spine image data set, we show that RMI is a more robust similarity measure for image registration than MI.

1 Introduction

1.1 Mutual Information

The mutual information (MI) between two variables is a concept with roots in information theory and essentially measures the amount of information that one variable contains about another. Put another way, it is the reduction in uncertainty of one variable given that we know the other [1]. MI was introduced as a similarity measure between images (both 2D and 3D) simultaneously by Viola *et al.* [2] and Maes *et al.* [3]. As a similarity measure, it has a number of advantages. In particular, it assumes no prior functional relationship between the images. Rather, it assumes a statistical relationship that can be captured by analyzing the images' joint entropy. Mutual information is closely related to joint entropy. Specifically, given image A and image B , the joint entropy $H(A, B)$ can be calculated as:

$$H(A, B) = - \sum_{a,b} p_{AB}(a, b) \log p_{AB}(a, b)$$

where p_{AB} is the joint probability distribution of pixels associated with images A and B . The joint entropy is minimized when there is a one-to-one mapping

between the pixels in A and their counterparts in B and it increases as the statistical relationship between A and B weakens. Here, since we are dealing with discrete images, we express all entropies with sums instead of integrals. In general, we must divine the probability distribution associated with each image by binning the values into histograms.

Mutual information considers both the joint entropy $H(A, B)$ and the individual entropies $H(A)$ and $H(B)$ where:

$$H(X) = - \sum_x p_X(x) \log p_X(x)$$

MI is defined as:

$$H(A) + H(B) - H(A, B)$$

Intuitively, as Viola notes, maximizing the mutual information between two images seems to try and find the most complex overlapping regions (by maximizing the individual entropies) such that they explain each other well (by minimizing the joint entropy) [2]. As a similarity measure, mutual information has enjoyed a great deal of success, particularly in the medical imaging domain [4]. It is robust to outliers, efficient to calculate, and generally provides smooth cost functions on which to optimize.

There is, however, one important drawback to mutual information as a way of comparing images: it fails to take geometry into account since it considers only pixel values, and not pixel positions.

1.2 Previous Work

Since its introduction, there have been a number of extensions to mutual information. In [5], an overlap invariant extension called normalized mutual information is introduced. In [6] and [7], different methods for calculating entropy are used. Each of these extensions, however, continues to ignore important spatial information in images.

More recently, researchers have begun to extend mutual information to include spatial information. In particular, in [8], mutual information is multiplied with a term that compares the local gradients of the two images. In [9], “second order” mutual information is defined. This formulation involves calculating the marginal and joint entropies of a pixel and one neighbor instead of a single pixel. Both cases add some amount of spatial information to the existing framework of mutual information and both cases validate their metrics on the problem of medical image registration. Also, both papers report that the final accuracy of their registrations are essentially the same as when using mutual information. The real improvement of incorporating spatial information lies in the robustness of the measure. A robust similarity measure is one with a smooth, convex landscape with respect to misregistration, specifically, one that does not have too many local extrema on the way to a global optimum. Both [8] and [9] report improved robustness of their metrics over standard mutual information. Each,

however, has important limitations. In [9], only one neighbor at a time is considered which leaves out a great deal of spatial information. In [8], they do not actually extend mutual information, rather MI is multiplied by a different term which accounts for the neighborhood information we are after.

In this paper, we present regional mutual information (RMI), an extension to mutual information that incorporates spatial information in a way that leads to smoother, more robust energy functions than have previously been reported. The paper is organized as follows: Section 2 presents our formulation and justification of RMI and details our algorithm for calculating it. Section 3 presents the results from testing our algorithm vs. MI and those in [8] and [9] on a 2D-3D medical image registration problem using a clinical gold-standard for validation. Finally, Sections 4 and 5 discuss our work and present some conclusions and related future research directions.

2 Regional Mutual Information

2.1 Formulation

When using mutual information to compare two images, the actual probability distributions associated with each image are not known. Rather, marginal and joint histograms are usually calculated to approximate the respective distributions. One logical way to extend mutual information is to extend the dimensionality of the histograms. For example, using standard mutual information, one set of pixel co-occurrences in the joint distribution is represented by one entry in a two-dimensional joint histogram. What if we were to consider corresponding pixels and their immediate, 3×3 neighborhoods? This would be an entry into an 18D joint histogram. We could calculate the mutual information exactly as before only this time we would use 9D histograms for the marginal probabilities and an 18D histogram for the joint distribution.

One way to think of a d -dimensional histogram is as a set of points in \mathbb{R}^d . Mutual information treats images as distributions of pixel values in 1D. Our formulation of RMI begins by recasting the problem to treat each image as a distribution of multi-dimensional points where each point represents a pixel and its neighborhood as depicted in Figure 1a.

2.2 The Curse of Dimensionality

At this point, however, we run into the curse of dimensionality. Essentially, the space where these points reside grows exponentially with each new dimension added. Figure 2 offers a look at why this is a problem. Here, what we've done is created some multi-dimensional distributions with known entropies (e.g. normal, exponential) in \mathbb{R}^2 , \mathbb{R}^4 , and \mathbb{R}^6 and estimated their entropies with a varying number of samples. The important thing to notice here is that, as dimensionality increases, more and more samples are needed to populate the space to get a reasonable estimate of the entropy. For distributions in \mathbb{R}^6 , even 2 million samples

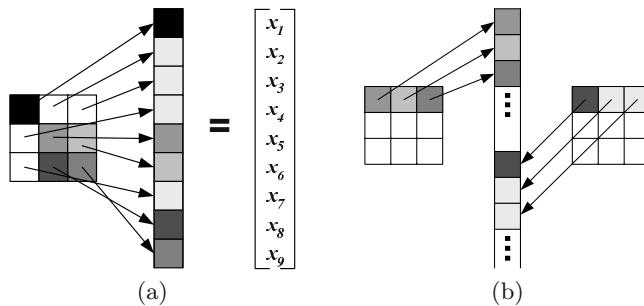


Fig. 1. (a) An illustration of the relationship between an image region and its corresponding multi-dimensional point. (b) Corresponding image neighborhoods and the multi-dimensional point representing them in the joint distribution.

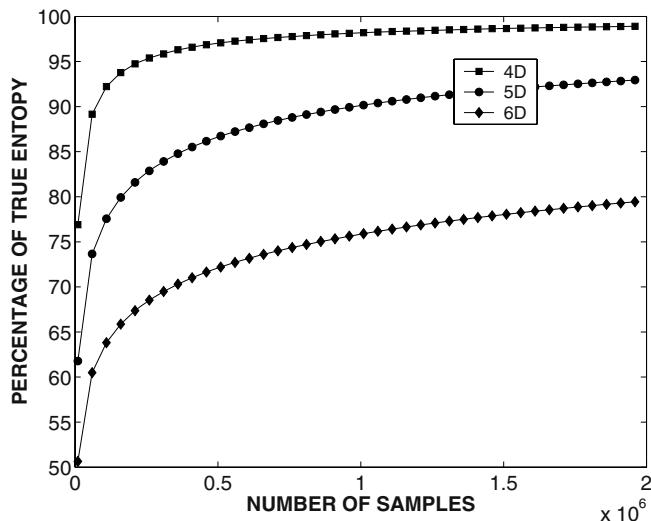


Fig. 2. An illustration of the curse of dimensionality as it pertains to sampling and entropy. On the x -axis, we have the number of samples and on the y -axis, the percentage of the true entropy we get when using those samples to approximate the entropy. As the dimensionality increases linearly, we need an exponentially increasing number of samples to get a reasonable estimate of a distribution's entropy.

are not sufficient to approximate the entropy correctly. Two million samples would correspond roughly to a comparison between images of resolution 1500×1500 , an impractical figure for most real world applications of image similarity.

2.3 A Simplifying Assumption

We can take advantage of the fact that the entropy of a discrete distribution is invariant to rotations and translations [10] in order to make our problem more

tractable. Specifically, we can try and rotate and translate our high-dimensional distribution into a space where each dimension is independent. This is a re-statement of the independent components analysis (ICA) problem as defined by Bell and Sejnowski [11]. Unfortunately, ICA is an extremely underdetermined problem and computationally too expensive for our purposes.

Instead we make the simplifying assumption that our high-dimensional distribution is approximately normally distributed. If this is the case, all we need to do is transform the points into a space where they are uncorrelated which, given the normal assumption, implies independence. Independence in each dimension allows us to decouple the entropy calculation [1] from one involving a d -dimensional distribution to one involving d independent 1-dimensional distributions. Specifically, the entropy of a normally distributed set of points in \mathbb{R}^d with covariance matrix Σ_d is [10]:

$$H_g(\Sigma_d) = \log((2\pi e)^{\frac{d}{2}} \det(\Sigma_d)^{\frac{1}{2}}).$$

This is mathematically equivalent to transforming the points into a new basis (B_u) where each dimension is uncorrelated, projecting the data onto each of the d new axes, and summing the entropies of those d independent 1-dimensional distributions.

In practice, given a high-dimensional distribution represented by a set of data points $P = [p_1, \dots, p_N]$, we can calculate this approximation to its entropy by first centering the data with respect to its mean, then diagonalizing its covariance matrix, and finally summing the entropies along each dimension. This process is the same as that used by principal components analysis (PCA) [12] and, essentially, what we are doing is summing the entropies along each of the orthogonal principal modes of variation.

2.4 Algorithm

Now that we have a method for efficiently calculating the entropy of a high-dimensional histogram, we can use it to calculate the RMI of a pair of images. The algorithm proceeds as follows:

1. Given two images A and B , for each corresponding pair of pixels $[A_{ij}, B_{ij}]$, create a vector \mathbf{v}_{ij} (Figure 1b) representing the co-occurrences of the pixels and their neighbors for some specified square radius r . This vector is now a point p_i in a d -dimensional space where $d = 2(2r + 1)^2$. The image margins can be handled in a number of ways. We chose simply to ignore the pixels along the edges as we assume that they will not have too pronounced of an effect on the final entropy. Given radius r and $m \times n$ images, we now have a distribution of $N = (m - 2r)(n - 2r)$ points represented by a $d \times N$ matrix $P = [p_1, \dots, p_N]$.
2. Subtract the mean from the points so that they are centered at the origin:

$$P_0 = P - \frac{1}{N} \sum_i^N p_i$$
3. Calculate the covariance of the points: $C = \frac{1}{N} P_0 P_0^T$

4. Estimate the joint entropy as: $H_g(C)$
5. Estimate the marginal entropies as $H_g(C_A)$ and $H_g(C_B)$ where C_A is the $\frac{d}{2} \times \frac{d}{2}$ matrix in the top left of C and C_B is the $\frac{d}{2} \times \frac{d}{2}$ matrix in the bottom right.
6. Calculate the RMI = $H_g(C_A) + H_g(C_B) - H_g(C)$

Asymptotically, RMI has the same performance as standard mutual information. Given $n \times n$ images, the performance of standard mutual information is $O(n^2)$. Similarly, RMI's performance is $O(n^2 d^2)$ which represents the work required to calculate the covariance matrix. Since d remains constant for a given choice of the neighborhood radius, asymptotically, RMI's performance converges to $O(n^2)$ as well.

2.5 Justification

In general, 1-dimensional projections of data made up of vectors of independent and identically distributed values tend to be normally distributed [13]. At a high level, this observation is derived by an appeal to the central limit theorem. Unfortunately, in our case, the vectors that make up our data are not independent (pixel values close together are not generally independent of each other) so the central limit theorem does not hold. There are, however, forms of the central limit theorem which allow weak dependence among the data. One such form, proven in [14], holds that the central limit theorem still applies in the case of m -dependent variables. A sequence of random variables (X_1, X_2, \dots, X_n) is m -dependent if, for some positive integer m , the inequality $s - k > m$ implies that the two sets (X_1, X_2, \dots, X_k) and $(X_s, X_{s+1}, \dots, X_n)$ are independent. So, given data made up of m -dependent vectors, the central limit theorem does apply which implies that 1-dimensional projections of this data should also tend to be normally distributed.

Now what we must do is demonstrate that our data, though not independent, is at least m -dependent which would tend to support our assumption that its projections are normal. Intuitively, m -dependence requires that variables far away from each other in a sequence are independent. In our case, each data point represents a sequence of pixels (Figure 1a). Though the pixel values are locally dependent, pixels that are far apart in the sequence are further from each other in the neighborhood they are drawn from and, hence, more likely to be independent. Indeed, as the size of the neighborhood increases, it becomes more and more likely that pixels far enough apart in the sequence representing that neighborhood are independent.

As mentioned earlier, calculating the RMI is equivalent to projecting the data onto each of the axes of the new, uncorrelated basis (B_u) and summing up the entropies. To test the validity of our assumption that these projections are generally normal, we generated 200 random pairs of medical images (amorphous silicon detector X-ray images) of the type we would expect to see in a typical registration problem. We then calculated RMI as usual, projected the data onto each of the axes of B_u , and used a Kolmogorov-Smirnov normality test ($\alpha = 0.01$)

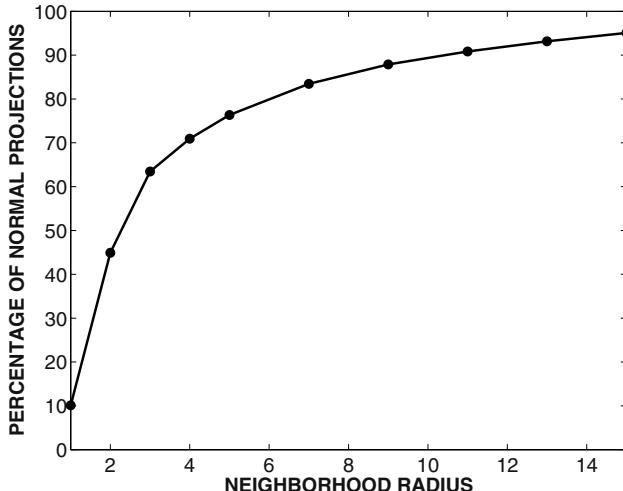


Fig. 3. For 200 random X-ray images, we plot the mean percentage of projections onto B_u which are normal vs. the radius of the neighborhood used. As predicted by the m -dependent central limit theorem, the percentage of projections that can be considered normally distributed increases as the radius increases.

to determine which of the projections could be considered normal. For each pair of images, we performed this test for neighborhood radii ranging from 1 to 15. The results can be seen in Figure 3 and suggest a tradeoff between the time to calculate RMI and the accuracy of our assumption. Experimentally, we have found that even a radius as low as $r = 2$ works quite well in practice.

3 Results

3.1 Validation

As we have mentioned, mutual information as an image similarity measure has enjoyed a large degree of success in medical image registration applications. We chose to validate RMI on just such a problem, specifically 2D-3D medical image registration. The 2D-3D registration problem involves taking one or more X-ray projection (2D) images of a patient's anatomy and using these projections to determine the rigid transformation \mathbf{T} (rotation and translation) that aligns the coordinate system of a CT (3D) image with that of the X-ray projection images and an operating room. 2D-3D registration is an important primitive in applications such as image-guided spine surgery [15,16] and radiosurgery [17,18].

We validate RMI as a means of performing 2D-3D intensity-based registration using the same experimental set-up and clinical gold-standard as in [19]. We use archived clinical data from the CyberKnife Stereotactic Radiosurgery System (Accuray, Inc., Sunnyvale, CA) which includes a preoperative contrast 3D CT

Table 1. 2D-3D Spine Image Target Registration Error

Similarity Measure	TRE (mm)	Unsuccessful Registrations
	Mean	
RMI	1.25	8%
Pluim, <i>et al.</i>	1.31	15%
Mutual Information	1.33	18%
Rueckert, <i>et al.</i>	1.43	30%

scan, 2 orthogonal intraoperative amorphous silicon detector 2D X-ray images, and a built-in gold-standard calculated using bone-implanted fiducial markers. We compared RMI not only to mutual information (implemented using histograms as per [3]), but also to our own implementations of the aforementioned similarity measures with spatial information from Pluim, *et al.* [8] and Rueckert, *et al.* [9]. There have been a number of other similarity measures used for this problem [20] including cross-correlation and gradient-correlation, two measures that also take neighborhood relationships into account. However, following [21], we focus on mutual information as, on real data, it has been shown to be more accurate.

3.2 Experiments

Each experiment involved an initial transformation generated by perturbing the gold-standard reference transformation by adding randomly generated rotations and translations. The initial transformations were characterized by computing the target registration error (TRE) [22] for the transformation and grouped into eight initial TRE intervals: 0–2, 2–4, 4–6, 6–8, 8–10, 10–12, 12–14, and 14–16 mm. For each of 6 patients (3 with cervical lesions and 3 with thoracic lesions) and each similarity measure, 160 registrations were performed, 20 in each of the six misregistration intervals. The TRE value was computed for each registration transformation as the difference between the positions of a target mapped by the evaluated transformation and the gold-standard transformation. The TRE values were computed for each voxel inside a rectangular box bounding the vertebra closest to the lesion and then averaged. The registrations were characterized as either “successful” if the $\text{TRE} < 2.5 \text{ mm}$ or “unsuccessful” if the $\text{TRE} \geq 2.5 \text{ mm}$. The results are listed in Table 1. Here we see that, while slightly more accurate, the real win from using RMI lies in its robustness, or its success rate with respect to misregistration. We take a more detailed look at robustness in Figure 4 where we plot percentage of successful registrations vs. initial TRE for all 4 similarity measures. While most of the measures perform well when started close to the initial solution, as initial TRE increases, RMI performs much better. By the time the initial TRE is in the range of 14–16 mm, RMI performs almost 50% better than the next best measure. In terms of CPU time, the difference between standard mutual information and RMI is quite small. On average registrations with mutual information took 101 sec., while those with RMI took 174 sec.

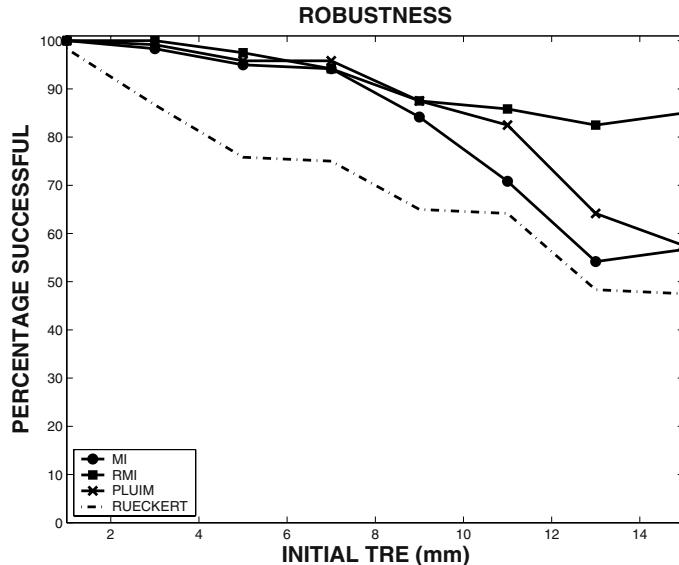


Fig. 4. Percentage of successful registrations for initial transformations with different initial TRE values. Each data point represents a 2 mm range of initial TRE values centered at the x -coordinate.

3.3 Noisy CT Data

Intensity-based 2D-3D registration requires an accurate CT scan. Usually this means that the patients are required to hold their breath during the scan so that breathing artifacts don't introduce noise into the data. Not all patients, however, are able to hold their breath for a sufficient period of time. Those that don't usually leave CTs that are noisy enough to severely affect the performance of an image-based registration algorithm. We performed the same experiments as above on two patients whose CT scans contained severe breathing artifacts. The results, seen in Figure 5, show the same basic trends as those from Figure 4 but are much more dramatic. For initial transformations close to the gold-standard, RMI still succeeds 100% of the time and its performance drops off much more gradually than the other three measures.

4 Discussion

To get a closer look at why RMI is more robust than MI, we analyzed a specific situation from the data-set above where MI often fails. In particular, instead of considering 6 independent parameters, we looked only at misregistration with respect to the x -axis and plotted RMI with varying neighborhood sizes (up to $r = 4$) vs. MI. The results, shown in Figure 6, illustrate one of the advantages of RMI. MI clearly shows a strong local maximum to the right of the global

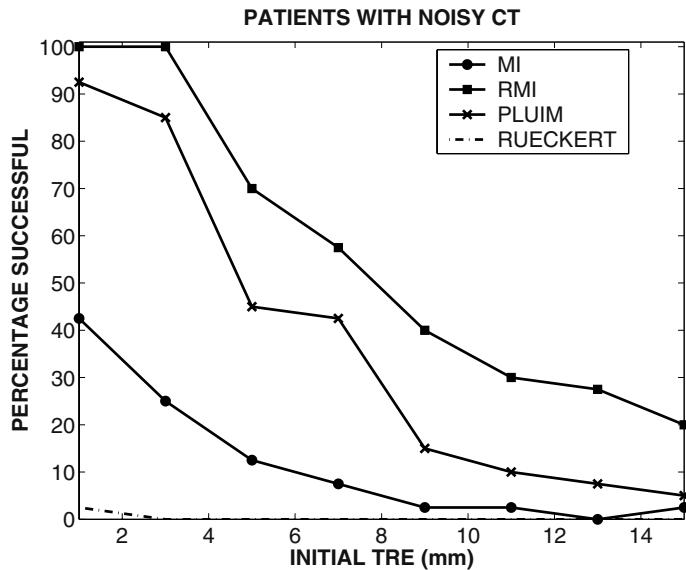


Fig. 5. Percentage of successful registrations for initial transformations with different initial TRE values. Each data point represents a 2 mm range of initial TRE values centered at the x -coordinate.

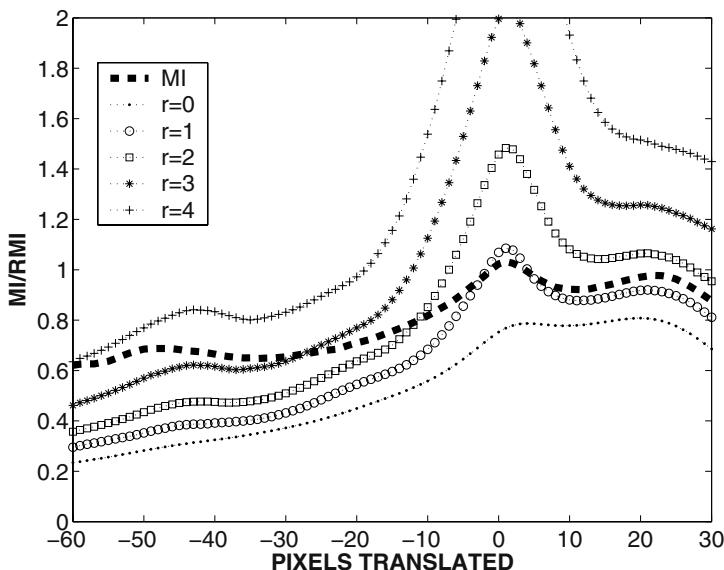


Fig. 6. Plot of mutual information as a function of misregistration along the x -axis. Also includes plots of RMI with neighborhoods of varying sizes ($r = 0$, $r = 1$, $r = 2$, $r = 3$, $r = 4$).

maximum which could have a large affect on the success of the optimization. As we begin to consider RMI with larger and larger neighborhoods, more spatial information is brought into the metric and we get a stronger peak at the global optimum and a smoother, more accurate similarity landscape away from it.

5 Conclusions

Mutual information as an image similarity measure has enjoyed a great deal of success in a variety of fields, medical image registration in particular. We have extended mutual information in a principled way to arrive at RMI which incorporates spatial information inherent in images. We have demonstrated RMI's improved robustness as a similarity measure and validated its use on real, clinical data from the medical imaging domain. In the future, we hope to validate RMI more extensively on larger clinical data sets. In addition, we're interested in applying RMI to non-rigid registration of clinical data. We would also like to apply RMI to more problems from other domains such as the creation of image mosaics, the querying of image databases, and the tracking of human motion.

References

1. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, New York (1991)
2. Viola, P., Wells III, W.M.: Alignment by maximization of mutual information. *International Journal of Computer Vision* **24** (1997) 137–154
3. Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging* **16** (1997) 187–198
4. Pluim, J., Maintz, J., Viergever, M.: Mutual information based registration of medical images: a survey. *IEEE Transactions on Medical Imaging* **22** (2003) 986–1004
5. Studholme, C., Hill, D.L.G., Hawkes, D.J.: An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognit.* **32** (1999) 71–86
6. Rodriguez-Carranza, C., Loew, M.: A weighted and deterministic entropy measure for image registration using mutual information. *Medical Imaging 1998: Image Processing Proc. SPIE* **3338** (1998) 155–166
7. Ma, B., Hero, A.O., Gorman, J., Michel, O.: Image registration with minimal spanning tree algorithm. In: *IEEE Int. Conf. on Image Processing*, Vancouver, BC (2000)
8. Pluim, J.P.W., Maintz, J.B.A., Viergever, M.A.: Image registration by maximization of combined mutual information and gradient information. *IEEE Trans. Med. Imaging* **19** (2000) 809–814
9. Rueckert, D., Clarkson, M.J., Hill, D.L.G., Hawkes, D.J.: Non-rigid registration using higher-order mutual information. *Medical Imaging 2000: Image Processing Proc. SPIE* **3979** (2000) 438–447
10. Shannon, C.: A mathematical theory of communication. *The Bell System Technical Journal* **27** (1948) 379–423

11. Bell, A., Sejnowski, T.: An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* **7** (1995) 1129–1159
12. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons Inc. (2001)
13. Diaconis, P., Freedman, D.: Asymptotics of graphical projection pursuit. *Annals of Statistics* **12** (1984) 793–815
14. Hoeffding, W., Robbins, H.: The central limit theorem for dependent random variables. *Duke Mathematical Journal* **15** (1948) 773–780
15. Lavallée, S., Troccaz, J., Sautot, P., Mazier, B., Cinquin, P., Merloz, P., Chirossel, J.P.: Computer-assisted spinal surgery using anatomy-based registration. In Taylor, R.H., Lavallée, S., Burdea, G., Mösge, R., eds.: Computer-Integrated Surgery: Technology and Clinical Applications. MIT Press, Cambridge, MA (1996) 425–449
16. Weese, J., Penney, G.P., Buzug, T.M., Hill, D.L.G., Hawkes, D.J.: Voxel-based 2-D/3-D registration of fluoroscopy images and CT scans for image-guided surgery. *IEEE Trans. Inform. Technol. Biomedicine* **1** (1997) 284–293
17. Murphy, M.J., Adler, Jr., J.R., Bodduluri, M., Dooley, J., Forster, K., Hai, J., Le, Q., Luxton, G., Martin, D., Poen, J.: Image-guided radiosurgery for the spine and pancreas. *Comput. Aided Surg.* **5** (2000) 278–288
18. Ryu, S.I., Chang, S.D., Kim, D.H., Murphy, M.J., Le, Q.T., Martin, D.P., Adler, Jr., J.R.: Image-guided hypo-fractionated stereotactic radiosurgery to spinal lesions. *Neurosurgery* **49** (2001) 838–846
19. Russakoff, D.B., Rohlfing, T., Maurer, Jr., C.R.: Fast intensity-based 2D-3D image registration of clinical data using light fields. *Proc. 9th IEEE Int. Conf. Computer Vision (ICCV 2003)* (2003) 416–423
20. Penney, G.P., Weese, J., Little, J.A., Desmedt, P., Hill, D.L.G., Hawkes, D.J.: A comparison of similarity measures for use in 2D-3D medical image registration. *IEEE Trans. Med. Imaging* **17** (1998) 586–595
21. Russakoff, D.B., Rohlfing, T., Ho, A., Kim, D.H., Shahidi, R., Adler, Jr., J.R., Maurer, Jr., C.R.: Evaluation of intensity-based 2D-3D spine image registration using clinical gold-standard data. In Gee, J.C., Maintz, J.B.A., Vannier, M.W., eds.: *Proc. Second Int. Workshop on Biomedical Image Registration (WBIR 2003). Lecture Notes in Computer Science 2717*. Springer-Verlag, Berlin (2003) 151–160
22. Maurer, Jr., C.R., Maciunas, R.J., Fitzpatrick, J.M.: Registration of head CT images to physical space using a weighted combination of points and surfaces. *IEEE Trans. Med. Imaging* **17** (1998) 753–761

Author Index

- Abraham, Isabelle IV-37
Agarwal, Ankur III-54
Agarwal, Sameer II-483
Ahmadyfard, Ali R. IV-342
Ahonen, Timo I-469
Ahuja, Narendra I-508, IV-602
Aloimonos, Yiannis IV-229
Antone, Matthew II-262
Argyros, Antonis A. III-368
Armspach, Jean-Paul III-546
Arnaud, Elise III-302
Arora, Himanshu I-508
Åström, Kalle III-252
Attias, Hagai IV-546
Aubert, Gilles IV-1
Auer, P. II-71
Avidan, Shai IV-428
Avraham, Tamar II-58
Ayache, Nichlas III-79
- Bab-Hadiashar, Alireza I-83
Baker, Patrick IV-229
Balch, Tucker IV-279
Bar, Leah II-166
Barnard, Kobus I-350
Bart, Evgeniy II-152
Bartoli, Adrien II-28
Basalamah, Saleh III-417
Basri, Ronen I-574, II-99
Bayerl, Pierre III-158
Bayro-Corrochano, Eduardo I-536
Bebis, George IV-456
Bect, Julien IV-1
Belhumeur, Peter I-146
Belongie, Serge II-483, III-170
Bennamoun, Mohammed II-495
Besserer, Bernard III-264
Bharath, Anil A. I-482, III-417
Bicego, Manuele II-202
Bille, Philip II-313
Bissacco, Alessandro III-456
Blake, Andrew I-428, II-391
Blanc-Féraud, Laure IV-1
Borenstein, Eran III-315
Bouthemy, Patrick III-145
Bowden, Richard I-390
- Brady, Michael I-228, I-390
Brand, Matthew II-262
Bretzner, Lars I-322
Bronstein, Alexander M. II-225
Bronstein, Michael M. II-225
Brostow, Gabriel J. III-66
Broszio, Hellward I-523
Brown, M. I-428
Brox, Thomas II-578, IV-25
Bruckstein, Alfred M. III-119
Bruhn, Andrés IV-25, IV-205
Bülow, Thomas III-224
Burger, Martin I-257
Burgeth, Bernhard IV-155
Byvatov, Evgeny II-152
- Calway, Andrew II-379
Caputo, Barbara IV-253
Carbonetto, Peter I-350, III-1
Carlsson, Stefan II-518, IV-442
Chai, Jin-xiang IV-573
Chambolle, Antonin IV-1
Charbonnier, Pierre II-341
Charnoz, Arnaud IV-267
Chellappa, Rama I-588
Chen, Chu-Song I-108, II-190
Chen, Jiun-Hung I-108
Chen, Min III-468
Chen, Qian III-521
Chen, Yu-Ting II-190
Cheng, Qiansheng I-121
Chiuso, Alessandro III-456
Christoudias, Chris Mario IV-481
Chua, Chin-Seng III-288
Chung, Albert C.S. II-353
Cipolla, Roberto II-391
Claus, David IV-469
Cohen, Isaac II-126
Cohen, Michael II-238
Comaniciu, Dorin I-336, I-549
Cootes, T.F. IV-316
Coquerelle, Mathieu II-28
Cremers, Daniel IV-74
Cristani, Marco II-202
Cristóbal, Gabriel III-158

- Dahmen, Hansjürgen I-614
 Dalal, Navneet I-549
 Daniilidis, Kostas II-542
 Darcourt, Jacques IV-267
 Darrell, Trevor IV-481, IV-507
 Davis, Larry S. I-175, III-482
 Dellaert, Frank III-329, IV-279
 Demirci, M. Fatih I-322
 Demirdjian, David III-183
 Deriche, Rachid II-506, IV-127
 Derpanis, Konstantinos G. I-282
 Devernay, Frédéric I-495
 Dewaele, Guillaume I-495
 Dickinson, Sven I-322
 Doretto, Gianfranco II-591
 Dovgard, Roman II-99
 Drew, Mark S. III-582
 Drummond, Tom II-566
 Duan, Ye III-238
 Duin, Robert P.W. I-562
 Dunagan, B. IV-507
 Duraiswami, Ramani III-482
 Ebner, Marc III-276
 Eklundh, Jan-Olof IV-253, IV-366
 Engbers, Erik A. III-392
 Eong, Kah-Guan Au II-139
 Eriksson, Martin IV-442
 Essa, Irfan III-66
 Fagerström, Daniel IV-494
 Faugeras, Olivier II-506, IV-127, IV-141
 Favaro, Paolo I-257
 Feddern, Christian IV-155
 Fei, Huang III-497
 Fergus, Robert I-242
 Fermüller, Cornelia III-405
 Ferrari, Vittorio I-40
 Finlayson, Graham D. III-582
 Fischer, Sylvain III-158
 Fitzgibbon, Andrew W. IV-469
 Freitas, Nando de I-28, I-350, III-1
 Freixenet, Jordi II-250
 Fritz, Mario IV-253
 Frolova, Darya I-574
 Frome, Andrea III-224
 Fua, Pascal II-405, II-566, III-92
 Fuh, Chiou-Shann I-402
 Furukawa, Yasutaka II-287
 Fussenegger, M. II-71
 Gavrila, Darin M. IV-241
 Gheissari, Niloofar I-83
 Ghodsi, Ali IV-519
 Giblin, Peter II-313, II-530
 Giebel, Jan IV-241
 Ginneken, Bram van I-562
 Goldlücke, Bastian II-366
 Gool, Luc Van I-40
 Grossauer, Harald II-214
 Gumerov, Nail III-482
 Gupta, Rakesh I-215
 Guskov, Igor I-133
 Gyaourova, Aglika IV-456
 Hadid, Abdennour I-469
 Haider, Christoph IV-560
 Hanbury, Allan IV-560
 Hancock, Edwin R. III-13, IV-114
 Hartley, Richard I. I-363
 Hayman, Eric IV-253
 Heinrich, Christian III-546
 Heitz, Fabrice III-546
 Herda, Lorna II-405
 Hershey, John IV-546
 Hertzmann, Aaron II-299, II-457
 Hidović, Džena IV-414
 Ho, Jeffrey I-456
 Ho, Purdy III-430
 Hoey, Jesse III-26
 Hofer, Michael I-297, IV-560
 Hong, Byung-Woo IV-87
 Hong, Wei III-533
 Horaud, Radu I-495
 Hsu, Wynne II-139
 Hu, Yuxiao I-121
 Hu, Zhanyi I-190, I-442
 Huang, Fay II-190
 Huang, Jiayuan IV-519
 Huber, Daniel III-224
 Ieng, Sio-Song II-341
 Ikeda, Sei II-326
 Irani, Michal II-434, IV-328
 Jacobs, David W. I-588, IV-217
 Jawahar, C.V. IV-168
 Je, Changsoo I-95
 Ji, Hui III-405
 Jia, Jiaya III-342
 Jin, Hailin II-114

- Jin, Jesse S. I-270
Johansen, P. IV-180
Jones, Eagle II-591
Joshi, Shantanu III-570
- Kadir, Timor I-228, I-390
Kaess, Michael III-329
Kanade, Takeo III-558, IV-573
Kanatani, Kenichi I-310
Kang, Sing Bing II-274
Kasturi, Rangachar IV-390
Kervrann, Charles III-132
Keselman, Yakov I-322
Khan, Zia IV-279
Kimia, Benjamin II-530
Kimmel, Ron II-225
Kiryati, Nahum II-166, IV-50
Kittler, Josef IV-342
Kohlberger, Timo IV-205
Kokkinos, Iasonas II-506
Kolluri, Ravi III-224
Koulibaly, Pierre Malick IV-267
Koudelka, Melissa I-146
Kriegman, David I-456, II-287, II-483
Krishnan, Arun I-549
Krishnan, Sriram I-336
Kristjansson, Trausti IV-546
Kück, Hendrik III-1
Kuijper, Arjan II-313
Kumar, Pankaj I-376
Kumar, R. III-442
Kuthirummal, Sujit IV-168
Kwatra, Vivek III-66
Kwolek, Bogdan IV-192
- Lagrange, Jean Michel IV-37
Lee, Kuang-chih I-456
Lee, Mong Li II-139
Lee, Mun Wai II-126
Lee, Sang Wook I-95
Lenglet, Christophe IV-127
Leung, Thomas I-203
Levin, Anat I-602, IV-377
Lhuillier, Maxime I-163
Lim, Jongwoo I-456, II-470
Lim, Joo-Hwee I-270
Lin, Stephen II-274
Lin, Yen-Yu I-402
Lindenbaum, Michael II-58, III-392, IV-217
- Lingrand, Diane IV-267
Little, James J. I-28, III-26
Liu, Ce II-603
Liu, Tyng-Luh I-402
Liu, Xiuwen III-570, IV-62
Lladó, Xavier II-250
Loog, Marco I-562, IV-14
López-Franco, Carlos I-536
Lourakis, Manolis I.A. III-368
Lowe, David G. I-28
Loy, Gareth IV-442
Lu, Cheng III-582
- Ma, Yi I-1, III-533
Magnor, Marcus II-366
Maire, Michael I-55
Malik, Jitendra III-224
Mallick, Satya P. II-483
Mallot, Hanspeter A. I-614
Manay, Siddharth IV-87
Manduchi, Roberto IV-402
Maragos, Petros II-506
Marsland, S. IV-316
Martí, Joan II-250
Matei, B. III-442
Matsushita, Yasuyuki II-274
Maurer, Jr., Calvin R. III-596
McKenna, Stephen J. IV-291
McMillan, Leonard II-14
McRobbie, Donald III-417
Medioni, Gérard IV-588
Meltzer, Jason I-215
Mémin, Etienne III-302
Mendonça, Paulo R.S. II-554
Mian, Ajmal S. II-495
Mikolajczyk, Krystian I-69
Miller, James II-554
Mio, Washington III-570, IV-62
Mittal, Anurag I-175
Montagnat, Johan IV-267
Mordohai, Philippus IV-588
Moreels, Pierre I-55
Morency, Louis-Philippe IV-481
Moreno, Pedro III-430
Moses, Yael IV-428
Moses, Yoram IV-428
Muñoz, Xavier II-250
Murino, Vittorio II-202
- Narayanan, P.J. IV-168
Nechyba, Michael C. II-178

- Neumann, Heiko III-158
 Ng, Jeffrey I-482
 Nguyen, Hieu T. II-446
 Nicolau, Stéphane III-79
 Nielsen, Mads II-313, IV-180
 Nillius, Peter IV-366
 Nir, Tal III-119
 Nistér, David II-41
 Noblet, Vincent III-546
- Odehnal, Boris I-297
 Okada, Kazunori I-549
 Okuma, Kenji I-28
 Oliensis, John IV-531
 Olsen, Ole Fogh II-313
 Opelt, A. II-71
 Osadchy, Margarita IV-217
 Owens, Robyn II-495
- Padfield, Dirk II-554
 Pallawala, P.M.D.S. II-139
 Papenberg, Nils IV-25
 Paris, Sylvain I-163
 Park, JinHyeong IV-390
 Park, Rae-Hong I-95
 Pavlidis, Ioannis IV-456
 Peleg, Shmuel IV-377
 Pelillo, Marcello IV-414
 Pennec, Xavier III-79
 Perez, Patrick I-428
 Perona, Pietro I-55, I-242, III-468
 Petrović, Vladimir III-380
 Pietikäinen, Matti I-469
 Pinz, A. II-71
 Piriou, Gwenaëlle III-145
 Pollefeyns, Marc III-509
 Pollitt, Anthony II-530
 Ponce, Jean II-287
 Pottmann, Helmut I-297, IV-560
 Prados, Emmanuel IV-141
- Qin, Hong III-238
 Qiu, Huaijun IV-114
 Quan, Long I-163
- Rahimi, A. IV-507
 Ramalingam, Srikumar II-1
 Ramamoorthi, Ravi I-146
 Ranganath, Surendra I-376
 Redondo, Rafael III-158
 Reid, Ian III-497
- Ricketts, Ian W. IV-291
 Riklin-Raviv, Tammy IV-50
 Roberts, Timothy J. IV-291
 Rohlfing, Torsten III-596
 Rosenhahn, Bodo I-414
 Ross, David II-470
 Rother, Carsten I-428
 Russakoff, Daniel B. III-596
- Saisan, Payam III-456
 Samaras, Dimitris III-238
 Sarel, Bernard IV-328
 Sato, Tomokazu II-326
 Satoh, Shin'ichi III-210
 Savarese, Silvio III-468
 Sawhney, H.S. III-442
 Schaffalitzky, Frederik I-363, II-41, II-85
 Schmid, Cordelia I-69
 Schnörr, Christoph IV-74, IV-205, IV-241
 Schuurmans, Dale IV-519
 Seitz, Steven M. II-457
 Sengupta, Kuntal I-376
 Sethi, Amit II-287
 Shahrokni, Ali II-566
 Shan, Y. III-442
 Shashua, Amnon III-39
 Shokoufandeh, Ali I-322
 Shum, Heung-Yeung II-274, II-603, III-342
 Simakov, Denis I-574
 Singh, Maneesh I-508
 Sinha, Sudipta III-509
 Sivic, Josef II-85
 Smeulders, Arnold W.M. II-446, III-392
 Smith, K. IV-316
 Soatto, Stefano I-215, I-257, II-114, II-591, III-456, IV-87
 Sochen, Nir II-166, IV-50, IV-74
 Soler, Luc III-79
 Sommer, Gerald I-414
 Sorgi, Lorenzo II-542
 Spacek, Libor IV-354
 Spira, Alon II-225
 Srivastava, Anuj III-570, IV-62
 Steedly, Drew III-66
 Steiner, Tibor IV-560
 Stewenius, Henrik III-252
 Stürzl, Wolfgang I-614
 Sturm, Peter II-1, II-28

- Sugaya, Yasuyuki I-310
Sullivan, Josephine IV-442
Sun, Jian III-342
Suter, David III-107
Szepesvári, Csaba I-16
- Taleghani, Ali I-28
Tang, Chi-Keung II-419, III-342
Tarel, Jean-Philippe II-341
Taylor, C.J. IV-316
Teller, Seth II-262
Thiesson, Bo II-238
Thiré, Cedric III-264
Thormählen, Thorsten I-523
Thureson, Johan II-518
Todorovic, Sinisa II-178
Tomasi, Carlo III-596
Torma, Péter I-16
Torr, Philip I-428
Torresani, Lorenzo II-299
Torsello, Andrea III-13, IV-414
Treuille, Adrien II-457
Triggs, Bill III-54, IV-100
Tsin, Yanghai III-558
Tsotsos, John K. I-282
Tu, Zhuowen III-195
Turek, Matt II-554
Tuytelaars, Tinne I-40
Twining, C.J. IV-316
- Ullman, Shimon II-152, III-315
Urtasun, Raquel II-405, III-92
- Vasconcelos, Nuno III-430
Vemuri, Baba C. IV-304
Vidal, René I-1
- Wada, Toshikazu III-521
Wallner, Johannes I-297
Wang, Hanzi III-107
Wang, Jue II-238
Wang, Zhizhou IV-304
Weber, Martin II-391
Weickert, Joachim II-578, IV-25, IV-155, IV-205
Weimin, Huang I-376
Weiss, Yair I-602, IV-377
Weissenfeld, Axel I-523
- Welk, Martin IV-155
Wen, Fang II-603
Wildes, Richard P. I-282
Wills, Josh III-170
Windridge, David I-390
Wolf, Lior III-39
Wong, Wilbur C.K. II-353
Wu, Fuchao I-190
Wu, Haiyuan III-521
Wu, Tai-Pang II-419
Wu, Yihong I-190
- Xiao, Jing IV-573
Xu, Ning IV-602
Xu, Yingqing II-238
Xydeas, Costas III-380
- Yan, Shuicheng I-121
Yang, Liu III-238
Yang, Ming-Hsuan I-215, I-456, II-470
Yao, Annie II-379
Yao, Jian-Feng III-145
Yezzi, Anthony J. II-114, IV-87
Ying, Xianghua I-442
Yokoya, Naokazu II-326
Yu, Hongchuan III-288
Yu, Jingyi II-14
Yu, Simon C.H. II-353
Yu, Tianli IV-602
Yu, Yizhou III-533
Yuan, Lu II-603
Yuille, Alan L. III-195
- Zandifar, Ali III-482
Zboinski, Rafal III-329
Zelnik-Manor, Lihi II-434
Zeng, Gang I-163
Zha, Hongyuan IV-390
Zhang, Benyu I-121
Zhang, Hongjiang I-121
Zhang, Ruofei III-355
Zhang, Zhongfei (Mark) III-355
Zhou, S. Kevin I-588
Zhou, Xiang Sean I-336
Zhu, Haijiang I-190
Zisserman, Andrew I-69, I-228, I-242, I-390, II-85
Zomet, Assaf IV-377